

**TUGAS AKHIR - KS 1501**

**PENYUSUNAN ALTERNATIF SOLUSI FUNGSI-  
FUNGSI YANG OPTIMAL PADA TAHAP  
PERENCANAAN PROYEK PEMBUATAN APLIKASI  
XYZ DENGAN PENDEKATAN *VALUE  
ENGINEERING***

***PREPARATION OF ALTERNATIVE SOLUTIONS OF  
OPTIMAL FUNCTIONS IN ESTABLISHMENT  
PLANNING STAGE OF APPLICAION XYZ PROJECT  
WITH VALUE ENGINEERING***

Oleh:

Febe Sofia Loren Panjaitan NRP 05211440000085

Dosen Pembimbing

Ir. Khakim Ghozali, M.MT

Anisah Herdiyanti, S.Kom, M.Sc., ITIL.

Departemen Sistem Informasi

Fakultas Teknologi Informasi dan Komunikasi

Institut Teknologi Sepuluh Nopember

Surabaya 2018

**TUGAS AKHIR – KS 141501**

**PENYUSUNAN ALTERNATIF SOLUSI  
FUNGSI-FUNGSI YANG OPTIMAL PADA  
TAHAP PERENCANAAN PROYEK  
PEMBUATAN APLIKASI XYZ DENGAN  
PENDEKATAN *VALUE ENGINEERING***

**Febe Sofia Loren Panjaitan**

05211440000085

Dosen Pembimbing

**Ir. Khakim Ghozali, M.MT**

**Anisah Herdiyanti, S.Kom, M.Sc., ITIL.**

DEPARTEMEN SISTEM INFORMASI

Fakultas Teknologi Informasi dan Komunikasi

Institut Teknologi Sepuluh Nopember

Surabaya 2018



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT – KS 141501**

***PREPARATION OF ALTERNATIVE  
SOLUTIONS OF OPTIMAL FUNCTIONS IN  
ESTABLISHMENT PLANNING STAGE OF  
APPLICAION XYZ PROJECT WITH VALUE  
ENGINEERING***

**Febe Sofia Loren Panjaitan**

05211440000085

Supervisor :

**Ir. Khakim Ghozali, M.MT**

**Anisah Herdiyanti, S.Kom, M.Sc., ITIL.**

**INFORMATION SYSTEMS DEPARTMENT**

**Information and Communications Technology Faculty**

**Sepuluh Nopember Institut of Technology**

**Surabaya 2018**

# LEMBAR PENGESAHAN

## PENYUSUNAN ALTERNATIF SOLUSI FUNGSI-FUNGSI YANG OPTIMAL PADA TAHAP PERENCANAAN PROYEK PEMBUATAN APLIKASI XYZ DENGAN PENDEKATAN *VALUE ENGINEERING*

### TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer

pada

Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:

FEBE SOFIA LOREN PANJAITAN

0521144000085

Surabaya, 2018

KEPALA

DEPARTEMEN SISTEM INFORMASI

Dr. Ir. Aris Tjahyanto, M.Kom.

NIP 19650310 199102 1 001

**LEMBAR PERSETUJUAN**

**PENYUSUNAN ALTERNATIF SOLUSI  
FUNGSI-FUNGSI YANG OPTIMAL PADA  
TAHAP PERENCANAAN PROYEK  
PEMBUATAN APLIKASI XYZ DENGAN  
PENDEKATAN *VALUE ENGINEERING***

**TUGAS AKHIR**

Disusun untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Departemen Sistem Informasi  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**FEBE SOFIA LOREN PANJAITAN**

**0521144000085**

Disetujui Tim Penguji: Tanggal Ujian: Juli 2018

Periode Wisuda: September 2018

**Ir. Khakim Ghozali, M.MT**

(Pembimbing 1)

**Anisah Herdiyanti, S.Kom, M.Sc., ITIL.**

(Pembimbing 2)

**Dr. Apol Pribadi, S.T, M.T.**

(Penguji 1)

**Feby Artwodini, S.Kom, M.T.**

(Penguji 2)



# **PENYUSUNAN ALTERNATIF SOLUSI FUNGSI-FUNGSI YANG OPTIMAL PADA TAHAP PERENCANAAN PROYEK PEMBUATAN APLIKASI XYZ DENGAN PENDEKATAN *VALUE ENGINEERING***

**Nama Mahasiswa** : FEBE SOFIA LOREN PANJAITAN  
**NRP** : 0521144000085  
**Jurusan** : Sistem Informasi FTIK – ITS  
**Dosen Pembimbing 1** : Ir. Khakim Ghozali, M.MT  
**Dosen Pembimbing 2** : Anisah Herdiyanti, S.Kom, M.Sc., ITIL.

## **ABSTRAK**

Aplikasi XYZ merupakan aplikasi yang dibangun dengan mengikat kerjasama pihak ketiga untuk tujuan payment gateway pada pelaksanaan Event Olahraga 2018. Pembangunan aplikasi hanya didasarkan pada KAK (Kerangka Acuan Kerja) yang tidak memberikan batasan usulan. Perencanaan pihak ketiga berhasil membuat fungsi dan usulan biaya pembangunan aplikasi, namun penentuan biaya ini didasarkan pada pengalaman kerja pihak ketiga, tanpa mempertimbangkan apakah usulan tersebut sudah optimal atau belum. Hal ini bertentangan dengan prinsip perencanaan proyek TI yang membutuhkan pertimbangan terhadap hubungan biaya dan waktu.

Berdasarkan permasalahan tersebut, penelitian ini menggunakan metode *Value Engineering* yang berfokus pada *re-design* biaya. Terdapat lima fase yakni *Orientation*, *Information*, *Function Analysis*, *Creativity* dan *Evaluation* yang digunakan untuk mencari alternatif optimal pembangunan aplikasi XYZ. Metode ini didukung oleh beberapa teknik seperti *Function point*, *F.A.S.T Diagram*, pencarian indeks, dan

brainstorming, beberapa perhitungan seperti perhitungan waktu lembur dan pembobotan, serta grafik hubungan waktu dan biaya.

Hasil penelitian dengan menerapkan *Value Engineering* terbukti memberikan penghematan biaya pada pembangunan aplikasi XYZ, yakni dengan penerapan alternatif pemanfaatan visual programming. Hasil tersebut akan dituliskan dalam rangkuman *Value Engineering Change Proposals* (VECP).

**Kata kunci:** nilai, fungsi, *Value Engineering*, proyek TI, optimasi, aplikasi XYZ.

***PREPARATION FOR ALTERNATIVE SOLUTIONS  
OF OPTIMAL FUNCTIONS IN ESTABLISHMENT  
PLANNING STAGE OF APPLICATION XYZ  
PROJECT WITH VALUE ENGINEERING  
APPROACH***

**Student Name** : FEBE SOFIA LOREN PANJAITAN  
**Student Number** : 0521144000085  
**Departement** : Sistem Informasi FTIK – ITS  
**Supervisor 1** : Ir. Khakim Ghozali, M.MT  
**Supervisor 2** : Anisah Herdiyanti, S.Kom, M.Sc.,  
ITIL.

**ABSTRACT**

*XYZ is an application that was collaborated with the third as a payment gateway of a sport event in 2018. The established of this application was based on KAK (Kerangka Acuan Kerja) which did not limit the proposal. Plan of the third succeed to make Function and also cost of the application establishment, yet these cost was based on work experience of the third, without considered that the plan has been optimal or not. This was contradicted to the IT project planning principles that required consideration of cost and time relationships.*

*Based on the problem, this research used Value engineering method that focus on costs re-design. There are five Phases namely Orientation, Information, Function Analysis, Creativity and Evaluation which used to find optimal alternative XYZ application development. This method was supported by several techniques such as Function point, F.A.S.T Diagram, index search, and brainstorming, some calculations such as overtime and weight calculation, and time and cost relationship graphs.*

*The results of the research by applying Value Engineering proved to provide cost savings on the development of XYZ*

*applications, namely the application of alternative visual programming utilization. The results would be written in a summary of Value Engineering Change Proposals (VECP).*

***Keywords: Value, Function, Value engineering, IT project, optimize, XYZ application.***

## **KATA PENGANTAR**

Segala puji dan syukur penulis panjatkan pada Tuhan Yang Maha Esa karena atas rahmat dan kasih-Nya, penulis dapat menyelesaikan Tugas Akhir dengan judul :

**“PENYUSUNAN ALTERNATIF SOLUSI FUNGSI-  
FUNGSI YANG OPTIMAL PADA TAHAP  
PERENCANAAN PROYEK PEMBUATAN APLIKASI  
XYZ DENGAN PENDEKATAN *VALUE  
ENGINEERING*”**

sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer di Departemen Sistem Informasi – Institut Teknologi Sepuluh Nopember Surabaya.

Pada kesempatan ini, penulis ingin menyampaikan terima kasih kepada semua pihak yang telah memberikan doa, dukungan, bimbingan, arahan, bantuan, dan semangat dalam menyelesaikan tugas akhir ini, yaitu kepada:

1. Orang tua dan Keluarga penulis yang telah mendoakan dan senantiasa mendukung serta selalu memberikan semangat dalam penyelesaian tugas akhir ini.
2. Bapak Dr. Ir. Aris Tjahyanto, M.Kom, selaku Ketua Departemen Sistem Informasi ITS
3. Bapak Ir. Khakim Ghozali, M.MT. dan Ibu Anisah Herdiyanti, S.Kom, M.Sc., ITIL selaku dosen pembimbing yang telah meluangkan waktu dan pikiran untuk mendukung dan membimbing dalam penyelesaian tugas akhir penulis.
4. Ibu Feby Artwodini, selaku dosen wali yang telah memberikan pengarahan selama penulis menempuh masa perkuliahan dan penelitian tugas akhir.
5. Pak Hermono, selaku admin laboratoriu MSI yang membantu penulis dalam hal administrasi penyelesaian tugas akhir dan mendukung penyelesaian tugas akhir ini.
6. Para dosen Jurusan Sistem Informasi.

7. Tim pihak ketiga sebagai pihak yang turut membantu dalam pemberian informasi dan analisa dalam penyelesaian tugas akhir ini.
8. Terima kasih kepada Theresia Maria Ohoiwutun dan Dameria Maranatha Panjaitan yang selalu bersedia memberikan fasilitas pendukung utama pengerjaan Tugas Akhir ini.
9. Terima kasih kepada Harry Dio Sirait yang selalu mendukung dan memberikan semangat serta menemani penulis untuk menyelesaikan tugas akhir ini.
10. Teman-teman seperjuangan di Lab. MSI dan OSIRIS yang tidak dapat disebutkan namanya satu per-satu, terima kasih telah memberi semangat dan mendukung untuk segera menyelesaikan tugas akhir.
11. Pihak-pihak lain yang telah mendukung dan membantu dalam kelancaran penyelesaian tugas akhir.

Penyusunan laporan ini masih jauh dari sempurna, untuk itu penulis menerima adanya kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi para pembaca dan menjadi sebuah kontribusi bagi ilmu pengetahuan.

Surabaya, Juli 2018

**Penulis**

# DAFTAR ISI

ABSTRAK .....	i
ABSTRACT .....	iii
KATA PENGANTAR .....	v
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	xi
DAFTAR GRAFIK .....	xiii
DAFTAR TABEL .....	xv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	3
1.3 Batasan Masalah .....	4
1.4 Tujuan Tugas Akhir .....	4
1.5 Manfaat Penelitian .....	4
1.6 Relevansi .....	5
1.7 Target Luaran .....	5
BAB II TINJAUAN PUSTAKA .....	7
2.1 Dasar Teori .....	7
2.1.1 Proyek .....	7
2.1.2 Nilai .....	9
2.1.3 Fungsi .....	11
2.1.4 Optimalisasi .....	13
2.1.5 <i>Value engineering</i> .....	13
2.1.6 <i>Value Engineering Change Proposal (VECP)</i> .....	22
2.1.7 Perbandingan Metode .....	29
2.1.8 Teknik-Teknik Terkait .....	33
2.1.9 Hubungan Proyek, Kontrak dan <i>Value engineering</i> .....	52
BAB III METODOLOGI .....	55
3.1 Langkah-Langkah Penelitian .....	55
3.1.1 Flowchart Langkah Penelitian .....	55
3.1.2 <i>Input</i> , Proses, <i>Output</i> Langkah-Langkah Penelitian .....	56
3.2 Timeline Pengerjaan Tugas Akhir .....	64
BAB IV PERANCANGAN .....	67

4.1	Penjelasan Studi Kasus.....	67
4.2	<i>Value engineering</i> .....	69
4.2.1	<i>Pre-Analysis</i> .....	69
4.2.2	<i>Analysis</i> .....	69
BAB V IMPLEMENTASI .....		111
5.1	<i>Information Phase</i> .....	111
5.1.1	Ulasan Dokumen Internal .....	111
5.2	<i>Function Analysis Phase</i> .....	117
5.3	Hambatan dan Rintangan .....	121
BAB VI HASIL DAN PEMBAHASAN.....		123
6.1	<i>Information Phase</i> .....	123
6.1.1	Pengkajian Data Masa Lalu .....	123
6.1.2	Perhitungan Estimasi Biaya dengan <i>Function point</i> .....	131
6.1.3	Perbandingan Estimasi Biaya dengan Penyedia Jasa.....	154
6.2	<i>Function Analysis Phase</i> .....	159
6.2.1	Perhitungan Biaya Kebutuhan Fungsional.....	159
6.2.2	Pemilihan Objek Penelitian.....	182
6.2.3	Pembangunan FAST Diagram .....	187
6.3	<i>Creativity Phase</i> .....	203
6.3.1	Menampilkan Informasi Saldo.....	203
6.3.2	Menampilkan Aktivitas Transaksi .....	211
6.3.3	Menampilkan Aktivitas Pemesanan Layanan dan Jasa .....	218
6.3.4	Menampilkan Informasi Pendamping.....	227
6.3.5	Menampilkan Informasi Hotel di Sekitar.....	234
6.3.6	Menampilkan Informasi Mall <i>Official Partner</i> di Sekitar .....	241
6.3.7	Menampilkan Informasi Lokasi .....	248
6.4	<i>Evaluation Phase</i> .....	258
6.4.1	Pencarian Indeks Nilai .....	263
6.4.2	Pemilihan Rekomendasi.....	264
6.4.3	Perbandingan Estimasi Biaya .....	268
6.4.4	<i>Value Engineering Change Proposal</i> .....	271
BAB VII KESIMPULAN DAN SARAN .....		273
7.1	Kesimpulan.....	273

7.2 Saran.....	274
DAFTAR PUSTAKA .....	277
BIODATA PENULIS .....	281
LAMPIRAN A .....	A-1
LAMPIRAN B .....	B-1

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 2.1 <i>Software Development Life Cycle</i> (SDLC).....	8
Gambar 2.2 Proses dalam <i>Value engineering</i> [18].....	15
Gambar 2.3 Contoh Analisis Komponen Fungsi.....	17
Gambar 2.4 Item-Oriented <i>Analysis</i> .....	18
Gambar 2.5 <i>Function-Oriented Analysis</i> .....	19
Gambar 2.6 Hubungan Biaya dan Waktu Pelaksanaan Proyek .....	20
Gambar 2.7 (a) (b) dan (c) contoh <i>Value Engineering</i> <i>Change Proposal</i> pada Proyek Kontruksi.....	26
Gambar 2.8 Proses Pelaksanaan <i>Value Engineering</i> <i>Change Proposal</i> .....	28
Gambar 2.9 F.A.S.T Diagram .....	42
Gambar 2.10 <i>Technical</i> Oriented FAST .....	46
Gambar 2.11 <i>Customer</i> (Task) Oriented FAST .....	47
Gambar 2.12 Contoh FAST Diagram pada Pintu Teakwood.....	49
Gambar 2.13 Contoh FAST Diagram pada PC Proyektor .....	50
Gambar 2.14 Hubungan Tantangan Proyek dengan <i>Value</i> <i>engineering</i> .....	53
Gambar 3.1 Flowchart Langkah-Langkah Penelitian.....	55
Gambar 4.1 Pengelompokan Tipe Fungsi Aplikasi Mobile ...	96
Gambar 4.2 Pembagian Aktivitas Penggunaan Rasional Rose.....	106
Gambar 5.1 Contoh F.A.S.T Diagram pada Pembangunan Fungsi Aplikasi .....	119
Gambar 6.1 Arsitektur Aplikasi XYZ.....	126
Gambar 6.2 Aliran Informasi Aplikasi.....	127
Gambar 6.3 Proses Pelaksanaan Transaksi.....	128
Gambar 6.4 Proses Pencarian Informasi Umum .....	129
Gambar 6.5 Gambar (a), (b) dan (c) Gambar Tampilan Aplikasi yang Diharapkan.....	130

Gambar 6.6 Estimasi Eksponenesial oleh Jone's .....	137
Gambar 6.7 FAST Diagram Menampilkan Informasi Saldo .....	195
Gambar 6.8 FAST Diagram Menampilkan Aktivitas Transaksi.....	196
Gambar 6.9 FAST Diagram Menampilkan Aktivitas Pemesanan Layanan dan Jasa .....	197
Gambar 6.10 FAST Diagram Menampilkan Informasi Pendamping dan Penjemputan.....	198
Gambar 6.11 FAST Diagram Menampilkan Informasi Hotel di Sekitar.....	199
Gambar 6.12 FAST Diagram Menampilkan Informasi <i>Mall Official Partner</i> .....	200
Gambar 6.13 FAST Diagram Menampilkan Informasi Lokasi .....	201

## **DAFTAR GRAFIK**

Grafik 6.1 Perubahan Waktu Pengerjaan Aplikasi.....	265
Grafik 6.2 Perubahan Biaya Pembangunan Aplikasi .....	266
Grafik 6.3 Hubungan Waktu dan Biaya .....	267

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 2.1 Perbandingan <i>Value engineering</i> dengan <i>Value Management</i> dan <i>Investation Management</i> .....	29
Tabel 2. 2 Tabel Bobot Kompleksitas <i>Crude Function points</i> .....	35
Tabel 2.3 Karakteristik <i>Software</i> .....	37
Tabel 2.4 <i>Jone's First Order Estimate Exponent</i> .....	39
Tabel 2.5 Distribusi Usaha dari masing-masing Aktivitas .....	40
Tabel 3.1 <i>Input</i> , Proses dan <i>Output</i> penelitian .....	56
Tabel 3.2 Tabel Timeline Pengerjaan Tugas Akhir .....	65
Tabel 4.1 Kebutuhan Fungsional Usulan Pihak Ketiga .....	67
Tabel 4.2 Perhitungan Bobot Kompleksitas .....	71
Tabel 4.3 Template Hasil Pengelompokan <i>Inputan</i> .....	72
Tabel 4.4 Template Hasil Perhitungan CFP .....	73
Tabel 4.5 Standar Documentation GSC Data Communication .....	74
Tabel 4.6 Standar Documentation GSC Distributed Processing .....	75
Tabel 4.7 Standar Documentation GSC Performance .....	75
Tabel 4.8 Standar Documentation GSC Heavily Used Configuration .....	76
Tabel 4.9 Standar Documentation GSC Transaction Rate .....	77
Tabel 4.10 Standar Documentation GSC Online Data Entry .....	77
Tabel 4.11 Standar Documentation GSC End-user efisiensi .....	79
Tabel 4.12 Standar Documentation GSC Online Update .....	79
Tabel 4.13 Standar Documentation GSC Complex Processing .....	80
Tabel 4.14 Standar Documentation GSC Reuseability .....	80
Tabel 4. 15 Standar Documentation GSC Installation Ease .....	81
Tabel 4.16 Standar Documentation GSC Operational Ease .....	82
Tabel 4.17 Standar Documentation GSC Multiple Sites .....	83
Tabel 4.18 Standar Documentation GSC Facilitate Change .....	84

Tabel 4.19 Template Hasil Perhitungan RCAF .....	85
Tabel 4.20 Template Hasil Perhitungan Estimasi Biaya Total.....	86
Tabel 4.21 Daftar Tarif/Gaji Tenaga IT .....	88
Tabel 4.22 Template Penambahan Jumlah SDM .....	90
Tabel 4.23 Fungsi Perbandingan .....	91
Tabel 4.24 Template Tabel Perbandingan Aplikasi .....	92
Tabel 4.25 Template Breakdown effort dan Waktu Pengerjaan .....	93
Tabel 4.26 Template Estimasi Biaya Kebutuhan Fungsional.	95
Tabel 4.27 Template Penentuan Objek Penelitian.....	97
Tabel 4.28 Template Hasil Pencarian Alternatif Kebutuhan Fungsi.....	106
Tabel 4.29 Template Hasil Perubahan Biaya berdasarkan Skenario.....	107
Tabel 4.30 Template Hasil Pencarian Indeks Nilai .....	108
Tabel 5.1 Kebutuhan Fungsional Aplikasi .....	111
Tabel 5.2 Pihak Penanggungjawab Pembangunan Aplikasi...	112
Tabel 5.3 Tahapan Pengerjaan Aplikasi .....	113
Tabel 5.4 Detail Waktu Pengerjaan Pembangunan Aplikasi	115
Tabel 6.1 Hasil Pengelompokan <i>Inputan</i> .....	133
Tabel 6.2 Hasil Perhitungan CFP .....	134
Tabel 6.3 Hasil Perhitungan RCAF .....	135
Tabel 6.4 Hasil Perhitungan Estimasi Biaya Total.....	138
Tabel 6.5 Perhitungan Hari Kerja Biasa dalam 5 Bulan.....	140
Tabel 6.6 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 5 Bulan.....	144
Tabel 6.7 Perubahan Hari Kerja .....	145
Tabel 6.8 Perhitungan Jam Lembur .....	146
Tabel 6.9 Perhitungan Estimasi Biaya Lembur selama 5 Bulan .....	148
Tabel 6.10 Penambahan Jumlah SDM .....	150

Tabel 6.11 Perhitungan Estimasi Biaya dengan Penambahan SDM.....	151
Tabel 6.12 Penggunaan Visual Programming (Informasi Saldo) .....	153
Tabel 6.13 Hasil Perbandingan Aplikasi.....	154
Tabel 6.14 Perbandingan Hasil Perhitungan Estimasi Biaya .....	158
Tabel 6.15 Perhitungan CFP Kebutuhan Fungsional .....	163
Tabel 6.16 Breakdown effort dan Waktu Pengerjaan Login	166
Tabel 6.17 Breakdown effort dan Waktu Pengerjaan Informasi Saldo .....	167
Tabel 6.18 Breakdown effort dan Waktu Pengerjaan Aktivitas Transaksi.....	169
Tabel 6.19 Breakdown effort dan Waktu Pengerjaan Aktivitas Pemesanan Layanan dan Jasa .....	171
Tabel 6.20 Breakdown effort dan Waktu Pengerjaan Informasi Pendamping dan Penjemput Atlet.....	172
Tabel 6.21 Breakdown effort dan Waktu Pengerjaan Informasi Hotel.....	174
Tabel 6.22 Breakdown effort dan Waktu Pengerjaan Aktivitas Menampilkan <i>Mall Official</i> Partner.....	175
Tabel 6.23 Breakdown effort dan Waktu Pengerjaan Menampilkan Informasi Jadwal dan Pertandingan .....	177
Tabel 6.24 Breakdown effort dan Waktu Pengerjaan Aktivitas Menyampaikan Keluhan.....	179
Tabel 6.25 Breakdown effort dan Waktu Pengerjaan Aktivitas Memberi Petunjuk Lokasi .....	180
Tabel 6.26 Breakdown effort dan Waktu Pengerjaan Aktivitas Memberi Petunjuk Lokasi .....	182
Tabel 6.27 Estimasi Biaya Kebutuhan Fungsional .....	182
Tabel 6. 28 Penentuan Objek Penelitian .....	184

Tabel 6.29 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Informasi Saldo).....	204
Tabel 6.30 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Informasi Saldo).....	206
Tabel 6.31 Perhitungan Hari Lembur (Informasi Saldo).....	207
Tabel 6.32 Perhitunga Estimasi Biaya Lembur 77 hari (Informasi Saldo).....	208
Tabel 6.33 Penambahan Jumlah SDM (Informasi Saldo) ....	209
Tabel 6.34 Perhitungan Estimasi Biaya dengan Penambahan SDM .....	210
Tabel 6.35 Penggunaan Visual Programming (Informasi Saldo).....	211
Tabel 6.36 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Aktivitas Transaksi) .....	212
Tabel 6.37 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Aktivitas Transaksi) .....	213
Tabel 6.38 Perhitungan Hari Lembur (Aktivitas Transaksi) .....	213
Tabel 6.39 Perhitunga Estimasi Biaya Lembur 77 hari (Aktivitas Transaksi) .....	215
Tabel 6.40 Penambahan Jumlah SDM (Aktivitas Transaksi) .....	216
Tabel 6.41 Perhitungan Estimasi Biaya dengan Penambahan SDM .....	217
Tabel 6.42 Penggunaan Visual Programming (Aktivitas Transaksi) .....	217
Tabel 6.43 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Pemesanan Layanan dan Jasa) .....	219
Tabel 6.44 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Pemesanan Layanan dan Jasa) ..	220
Tabel 6.45 Perhitungan Hari Lembur (Pemesanan Layanan dan Jasa) .....	220

Tabel 6.46 Perhitunga Estimasi Biaya Lembur 77 hari (Pemesanan Layanan dan Jasa) .....	222
Tabel 6.47 Penambahan Jumlah SDM (Pemesanan Layanan dan Jasa) .....	223
Tabel 6.48 Perhitungan Estimasi Biaya dengan Penambahan SDM .....	224
Tabel 6.49 Penambahan Jumlah SDM (Pemesanan Layanan dan Jasa) .....	225
Tabel 6.50 Perhitungan Estimasi Biaya dengan Penambahan SDM .....	225
Tabel 6.51 Penggunaan Visual Programming (Pemesanan Layanan dan Jasa) .....	226
Tabel 6.52 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Informasi Pendamping).....	228
Tabel 6.53 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Informasi Pendamping).....	229
Tabel 6.54 Perhitungan Hari Lembur (Informasi Pendamping) .....	229
Tabel 6.55 Perhitunga Estimasi Biaya Lembur 77 hari (Informasi Hotel di Sekitar) .....	231
Tabel 6.56 Penambahan Jumlah SDM (Informasi Pendamping) .....	232
Tabel 6.57 Perhitungan Estimasi Biaya dengan Penambahan SDM.....	233
Tabel 6.58 Penggunaan Visual Programming (Informasi Pendamping) .....	234
Tabel 6. 59 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Informasi Hotel di Sekitar) .....	235
Tabel 6.60 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Informasi Hotel di Sekitar) .....	236

Tabel 6.61 Perhitungan Hari Lembur (Informasi Hotel di Sekitar) .....	236
Tabel 6.62 Perhitunga Estimasi Biaya Lembur 77 hari (Informasi Hotel di Sekitar).....	238
Tabel 6.63 Penambahan Jumlah SDM (Menampilkan informasi hotel di sekitar).....	239
Tabel 6.64 Perhitungan Estimasi Biaya dengan Penambahan SDM .....	240
Tabel 6.65 Penggunaan Visual Programming (Informasi Hotel di Sekitar) .....	241
Tabel 6.66 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Informasi <i>Mall Official Partner</i> di Sekitar) ..	242
Tabel 6.67 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Informasi <i>Mall Official Partner</i> di Sekitar) .....	243
Tabel 6.68 Perhitungan Hari Lembur (Informasi <i>Mall Official Partner</i> di Sekitar).....	243
Tabel 6.69 Perhitunga Estimasi Biaya Lembur 77 hari (Informasi <i>Mall Official Partner</i> di Sekitar) .....	245
Tabel 6.70 Penambahan Jumlah SDM (Informasi <i>Mall Official Partner</i> di Sekitar).....	246
Tabel 6.71 Perhitungan Estimasi Biaya dengan Penambahan SDM .....	247
Tabel 6.72 Penggunaan Visual Programming (Informasi <i>Mall Official Partner</i> di Sekitar) .....	248
Tabel 6.73 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Informasi Lokasi) .....	249
Tabel 6.74 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Informasi Lokasi).....	250
Tabel 6.75 Perhitungan Hari Lembur (Informasi Lokasi) ....	250
Tabel 6.76 Perhitunga Estimasi Biaya Lembur 77 hari (Informasi Lokasi).....	252

Tabel 6.77 Penambahan Jumlah SDM (Informasi Lokasi) ..	253
Tabel 6.78 Perhitungan Estimasi Biaya dengan Penambahan SDM.....	254
Tabel 6.79 Penambahan Jumlah SDM (Informasi Lokasi) ..	255
Tabel 6.80 Perhitungan Estimasi Biaya dengan Penambahan SDM.....	255
Tabel 6.81 Penggunaan Visual Programming (Informasi Lokasi).....	256
Tabel 6.82 Hasil Tahapan Creative .....	257
Tabel 6.83 Hasil Perubahan Biaya berdasarkan Skenario Pemanfaatan Waktu Lembur .....	258
Tabel 6.84 Hasil Perubahan Biaya berdasarkan Skenario Penambahan SDM versi 1 .....	259
Tabel 6.85 Hasil Perubahan Biaya berdasarkan Skenario Penambahan SDM versi 2 .....	260
Tabel 6.86 Hasil Perubahan Biaya berdasarkan Skenario Penggunaan Visual Programming.....	262
Tabel 6.87 Hasil Pencarian Indeks Nilai .....	263
Tabel 6.88 Perbandingan Harga Aplikasi.....	269

*Halaman ini sengaja dikosongkan*

# BAB I

## PENDAHULUAN

Pada bab ini akan dijelaskan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat tugas akhir serta relevansi tugas akhir.

### 1.1 Latar Belakang

Seiring dengan perkembangan, kebutuhan akan teknologi informasi juga semakin meningkat pada berbagai aspek kehidupan termasuk dalam dunia kerja. Di dalam dunia kerja, perusahaan dituntut untuk menerapkan teknologi informasi dalam berbagai proses bisnis agar dapat bersaing dengan kompetitor. Hal ini memacu munculnya berbagai proyek pengembangan sistem/teknologi informasi yang mampu memberikan nilai terkait permasalahan pada perusahaan mereka.

Nilai adalah rasio kualitas dan biaya [1]. Nilai tersebut dapat berupa otomatisasi yang berdampak pada pengurangan biaya tenaga kerja, reduksi waktu pengerjaan yang jika dikonversikan dapat melakukan penghematan sekian rupiah, pengambilan keputusan yang lebih cepat, dan lain-lain. Dalam implementasinya, setiap pihak yang akan menjalankan proyek akan membangun perencanaan yang dilengkapi dengan nilai yang ingin dicapai.

Namun pada implementasinya, kegagalan proyek IT justru sering terjadi. Berdasarkan penelitian dari Standish Group, dalam kurun waktu 2010-2014 terdapat sebanyak 31,1% proyek TI yang dibatalkan sebelum proyek tersebut selesai, dan terdapat sebanyak 52,7% proyek yang menghabiskan biaya lebih dari anggaran yang telah ditetapkan. Dilihat dari sisi kesuksesannya, hanya terdapat 16,2% dari *software* TI yang dapat terselesaikan dengan biaya dan waktu yang sesuai [2].

Permasalahan muncul karena terkadang *customer* sendiri tidak mengetahui nilai-nilai apa yang ingin mereka dapatkan dari

pembangunan proyek TI tersebut, selain itu, permintaan *customer* terhadap nilai-nilai (*Values*), yang ingin mereka dapatkan juga sering berubah-ubah bahkan pada saat pengembangan TI sudah dimulai atau sedang dilaksanakan [3]. Hal ini dapat disebabkan oleh penggalan informasi yang tidak *clear* pada saat identifikasi, *customer* tidak memahami bagaimana teknologi informasi itu bekerja, *customer* tidak dapat membedakan mana yang menjadi kebutuhan dan mana yang merupakan keinginannya dan *customer* tidak mengerti apakah semua hal yang mereka inginkan benar-benar optimal bagi penyelesaian permasalahan terkait isu-isu teknis dari sistem/teknologi informasi pada perusahaan mereka atau tidak. Hal ini berdampak pada ketidakseimbangan antara biaya, waktu dan sumber daya yang tersedia dengan nilai-nilai yang perusahaan inginkan. Sehingga berpotensi pada kegagalan.

Aplikasi XYZ merupakan aplikasi yang dibangun dengan tujuan *payment gateway* pada pelaksanaan *Event* Olahraga 2018 dengan target pengguna yakni atlet yang mengikuti olahraga tersebut, selama *event* olahraga dilaksanakan. Aplikasi ini dibangun dengan mengikat kerjasama dengan pihak ketiga, yang artinya tim *Event* Olahraga 2018 tidak bertanggungjawab terhadap identifikasi, klasifikasi dan analisis pada pembangunan aplikasi tersebut. Hasilnya adalah pihak ketiga telah berhasil membuat usulan dalam sebuah dokumen singkat yang berisi tentang kebutuhan fungsional, fitur, skenario, *timeline*, biaya, dll, yang dianggap sesuai dengan tujuan *payment gateway*. Permasalahan yang terjadi adalah usulan yang dimunculkan oleh pihak ketiga hanya didasarkan pada pengalaman yang disesuaikan dengan rentang harga yang diberikan oleh tim *event* olahraga 2018. Sehingga pihak ketiga berusaha untuk memberikan usulan biaya yang dirasa paling hemat dengan kecenderungan melakukan penyederhanaan persyaratan teknis, keterkaitan dan ketergantungan antara banyak fungsi sehingga dapat bersaing dengan kompetitor. Selain itu, pihak ketiga tidak mengetahui apakah usulan yang mereka buat sudah optimal atau belum, artinya skenario tersebut sudah merupakan usulan

dengan biaya paling hemat, sesuai dengan waktu dan SDM yang diestimasikan atau belum [4]. Hal ini beresiko mengakibatkan *overruns* biaya, penundaan jadwal, masalah pengendalian mutu, perselisihan, fungsi yang tidak berjalan sebagai mana mestinya, dan lain sebagainya.

Berdasarkan permasalahan tersebut, Tugas Akhir ini dibangun dengan menggunakan metode *Value engineering* yang merupakan metode yang dapat mencari keseimbangan fungsi terbaik antara biaya, keandalan dan kinerja sebuah proyek dengan berfokus pada *re-design* biaya [5], artinya rekayasa nilai yang dilakukan terkait pada pengurangan biaya pembangunan aplikasi, bukan kepada perubahan proses atau aliran pekerjaan sistem aplikasi. *Re-design* biaya yang dilakukan akan didasarkan pada usulan kebutuhan fungsional dan gaji karyawan bukan kepada fitur yang diberikan pihak ketiga. *Re-design* ini juga akan difokuskan pada pemanfaatan waktu lembur dan rekayasa SDM, artinya pengurangan biaya didapatkan dengan mencari beberapa alternatif pengurangan biaya dengan skenario terkait gaji karyawan. Hasil alternatif yang paling optimal dengan berbagai pertimbangan akan dituliskan dalam usulan perubahan berupa *Value Engineering Change Proposals* (VECP) [6].

## 1.2 Perumusan Masalah

Rumusan masalah pada pembuatan Tugas Akhir ini adalah Bagaimana hasil analisis *Value engineering* pada Aplikasi XYZ. Adapun analisis tersebut akan menjawab beberapa pertanyaan berikut ini.

1. Apa saja fungsi-fungsi aplikasi yang dapat di hemat?
2. Bagaimana alternatif solusi yang paling optimal dalam pembangunan aplikasi XYZ dengan fungsi utama sebagai *payment gateway*?
3. Berapa besar penghematan biaya yang dapat dilakukan setelah penerapan *Value engineering* dilakukan?

### 1.3 Batasan Masalah

Dari permasalahan yang telah disebutkan di atas, batasan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Tugas Akhir ini hanya terbatas pada usulan perubahan proposal, tidak sampai pada pengimplementasian usulan pada keberlangsungan pembangunan aplikasi XYZ.
2. Data *input*-an awal Tugas Akhir ini terbatas pada KAK (Kerangka Acuan Kerja) dari tim *Event Olahraga* 2018 dan pihak ketiga.

### 1.4 Tujuan Tugas Akhir

Tujuan pembuatan Tugas Akhir ini adalah mengetahui bagaimana analisis *Value engineering* dapat memberikan usulan perubahan kinerja yang lebih optimal sesuai dengan biaya, sumber daya dan waktu pengerjaan Aplikasi XYZ. Adapun perubahan tersebut terkait pada hal berikut ini.

1. Perolehan data fungsi-fungsi aplikasi yang dapat dihemat. Penghematan tersebut dapat berupa pengurangan atau penggabungan fungsi-fungsi aplikasi sehingga dapat memperpendek aliran informasi atau hal lain yang dapat menghemat biaya pembangunan aplikasi.
2. Perolehan skenario atau aliran informasi yang paling optimal untuk diimplementasikan dalam pembangunan aplikasi XYZ dengan *payment gateway* sebagai fungsi utama.
3. Perolehan data besar penghematan biaya yang dapat dilakukan setelah analisis *Value engineering* ini dilakukan.

### 1.5 Manfaat Penelitian

Tugas akhir ini diharapkan dapat memberikan manfaat antara lain:

1. Bagi pihak ketiga, membantu dalam menentukan rencana implementasi yang optimal bagi pembangunan Aplikasi XYZ serta dapat meminimalisir biaya pengeluaran, yang berarti dapat memberikan keuntungan yang lebih besar dari sebelumnya.

2. Bagi penulis, mendapatkan wawasan, pengalaman dan kesempatan untuk menerapkan ilmu pengetahuan yang diperoleh selama menempuh kuliah di Departemen Sistem Informasi terlebih pada bidang yang berhubungan dengan proyek TI
3. Bagi Jurusan Sistem Informasi ITS, menambah portofolio terkait penelitian terkait proyek perencanaan pengembangan teknologi informasi.

### **1.6 Relevansi**

Tugas Akhir ini memiliki relevansi terhadap mata kuliah pada Departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi (FTIK), ITS. Usulan Tugas Akhir ini berkaitan dengan matakuliah Statistika, Manajemen Pengadaan dan Investasi Teknologi Informasi (MPITI), Manajemen Proyek TI (MPTI) dan PSSI (Perencanaan Strategi Sistem Informasi). Tugas Akhir ini masuk kedalam lingkup Laboratorium Manajemen Sistem Informasi (MSI) dalam ranah Rencana TI (Departemen Sistem Informasi. 2017).

### **1.7 Target Luaran**

Target luaran dari pengerjaan Tugas Akhir ini adalah sebagai berikut.

1. Dokumenn pengerjaan Tugas Akhir berupa Buku Tugas Akhir, paper dan/atau jurnal ilmiah.

*Halaman ini sengaja dikosongkan*

## **BAB II**

### **TINJAUAN PUSTAKA**

Sebelum melakukan penelitian tugas akhir, dilakukan tinjauan pustaka terhadap tulisan dari beberapa sumber-sumber terpercaya sebagai berikut.

#### **2.1 Dasar Teori**

Bagian ini akan menjelaskan teori dan bahan penelitian lain yang menjadi dasar pengerjaan Tugas Akhir ini.

##### **2.1.1 Proyek**

Proyek adalah kegiatan/usaha sementara yang dilakukan oleh individu, kelompok, organisasi maupun perusahaan untuk menciptakan produk, layanan atau hasil lain yang bersifat unik dengan durasi waktu mulai dari beberapa minggu sampai beberapa tahun. Proyek bersifat sementara menunjukkan bahwa proyek memiliki permulaan dan akhir. Akhir yang dimaksud bukanlah masa berlaku/umur dari hasil akhir proyek, namun masa berakhirnya kontrak pengerjaan proyek. Proyek dikatakan berakhir ketika tujuan proyek telah tercapai atau ketika proyek dihentikan oleh pihak *client* karena tujuan yang tidak dapat dipenuhi, atau sudah tidak ada kebutuhan lagi [8].

Proyek bersifat unik, artinya tidak ada proyek yang sama sekali sama. Sekalipun hasil akhir proyek terbilang sama atau sejenis dan dikerjakan oleh tim yang sama pula, namun selalu ada hal yang membedakannya, misalnya lokasi, desain maupun keadaan atau kondisi lingkungan hasil akhir proyek [8]. Berikut ini adalah contoh cakupan proyek, namun tidak terbatas pada :

- Pengembangan produk atau jasa baru
- Perancangan sarana transportasi baru
- Pengembangan, pembangunan atau perubahan sistem informasi
- Penerapan prosedur atau proses bisnis baru
- Penanganan permohonan sebuah kontrak

Proyek memiliki siklus pengerjaan yang berbeda-beda, tergantung pada jenis proyek. Dalam proyek TI, siklus pengerjaan proyek disebut *Software Development Life Cycle* (SDLC) (**Gambar 2.1**) [9].



Gambar 2.1 *Software Development Life Cycle* (SDLC) [9]

1. *Planning*. Tahapan ini merupakan tahapan pengumpulan dan analisis kebutuhan, untuk memperoleh data masa lalu atau acuan kerja dan standar lain yang berhubungan dengan objek penelitian. Penelitian dengan *Value Engineering* berada pada tahapan ini. Saat pengumpulan data dilakukan, metode VE dijalankan untuk mencari tahu usulan yang tepat kepada pihak ketiga terkait pembangunan proyek.
2. *Analysis*. Tahapan ini merupakan tahapan analisis atau pengolahan data yang diperoleh pada tahap *planning*, termasuk melakukan analisis terhadap usulan hasil *Value engineering*. Tahapan ini akan menghasilkan dokumen Spesifikasi Kebutuhan yang akan digunakan sebagai pedoman untuk pengerjaan tahap selanjutnya.

3. *Design*. Tahapan ini merupakan tahapan persiapan desain *system* dan *software* yang sesuai dengan kebutuhan. Tahap ini juga membantu dalam menspesifikasikan kebutuhan *hardware* dan *software* serta seluruh arsitektur sistem.
4. *Implementation*. Tahap ini merupakan tahapan pengimplementasian perencanaan yang telah dibangun pada tahapan sebelumnya ke dalam *modules/units* dan kodingan.
5. *Testing*. Tahap ini merupakan tahapan pengujian hasil kodingan. Pengujian dapat dilakukan dengan berbagai cara seperti *blaxkbox testing*, *whitebox testing*, dll.
6. *Maintanance*. Tahapan ini merupakan tahapan perbaikan, pengembangan atau pemeriksaan berkala yang dilakukan setelah program diimplemntasikan.

### **2.1.2 Nilai**

Nilai memiliki arti yang berbeda-beda bagi setiap orang. Seorang pelanggan akan mengartikannya sebagai “Pembelian terbaik”, seorang *manufacturer* akan mengartikannya sebagai “harga terendah”, dan desainer akan mengartikannya sebagai “fungsionalitas tertinggi”. Nilai tidak berdiri sendiri, dengan kata lain nilai adalah penggabungan konsep waktu, manusia, subjek, dan keadaan [10].

#### **2.1.2.1 Pengertian Nilai**

Secara Tradisional, nilai adalah rasio kualitas dan biaya. Sehingga nilai dapat ditingkatkan dengan memperbaiki fungsi atau mengurangi biaya [11]. Namun saat ini kebanyakan manajer mengakui bahwa nilai berorientasi kepada pelanggan yang artinya nilai adalah rasio antara kepuasan akan kebutuhan dan sumber daya yang dibutuhkan untuk mencapainya. Nilai dikatakan meningkat jika kepuasan pelanggan meningkat dan/atau pengeluaran sumber daya berkurang [1].

Permasalahan yang sering terjadi adalah kebutuhan yang tidak selalu dapat terpenuhi karena kurangnya sumber daya, sehingga dibentuklah sistem nilai yang bertujuan untuk

menyeimbangkan kedua elemen ini agar proyek menjadi sukses. Secara tradisional, rasio dinyatakan sebagai keseimbangan antara ruang lingkup dan kualitas (kinerja) pada satu sisi dan waktu dan biaya (sumber daya) pada sisi lain [1]. Untuk memudahkan pengingatan nilai yang berorientasi pada pelanggan, dapat melihat rumusan sebagai berikut :

*Customer Value* = **NOT MORE** =

$$\frac{\textit{Need} + \textit{Objectives} + \textit{Targets}}{\textit{Maximum Overall Resources Expended}}$$

Dimana **Function (Need + Objectives + Targets)** dihitung dari performa permintaan pelanggan dan **resources** dihitung dari material, tenaga kerja, biaya, waktu, dan lain-lain yang diperlukan untuk mencapai fungsi itu [13]. Rumusan tersebut menunjukkan bahwa kemampuan antara apa yang tersedia dan apa yang dibutuhkan harus seimbang atau dengan kata lain apa yang tersedia harus sama dengan atau lebih besar dari apa yang dibutuhkan agar kebutuhan dapat terpenuhi dan hasilnya tercapai. Namun sebaliknya jika apa yang dibutuhkan lebih kecil dari apa yang tersedia, maka kebutuhan tidak dapat terpenuhi seluruhnya atau harus diadakan perubahan rencana (penyeimbangan *Function* dengan *resources*) untuk dapat mencapai hasil yang diinginkan.

### 2.1.2.2 Tipe-Tipe Nilai

Nilai atau *Value* terbagi atas lima tipe sebagai berikut.

1. *Use Value*  
Nilai didasarkan pada jumlah sumber daya yang dikeluarkan untuk memproduksi sebuah produk sesuai rancangan.
2. *Esteem Value*  
Nilai didasarkan pada jumlah sumber daya yang dikeluarkan untuk menghasilkan produk yang fungsinya dapat disesuaikan dengan keinginan.
3. *Exchange Value*

Nilai didasarkan pada kemampuan produk untuk diperdagangkan.

4. *Cost Value*

Nilai yang didasarkan pada keuntungan berupa uang yang dapat diperoleh dari hasil akhir produk.

5. *Function Value*

Nilai yang didasarkan pada hubungan kelayakan dengan biaya.

### 2.1.3 Fungsi

Pendekatan fungsional adalah dasar dari management nilai; analisis fungsi adalah langkah yang fundamental diberbagai pembelajaran tentang nilai.

#### 2.1.3.1 Pengertian Fungsi

Fungsi adalah konsep yang mendeskripsikan keinginan atau kebutuhan yang diperlukan untuk mewujudkan hasil atau keluaran (dapat berupa produk, layanan, sistem, bangunan, proses, dan lain-lain) yang diharapkan. Dalam rekayasa nilai, fungsi selalu digambarkan dalam ringkasan dua kata yang terdiri dari kata kerja aktif dan benda terukur (apa yang sedang dilakukan – kata kerja – dan apa yang sedang dilakukan – kata benda). Inilah dasar dari apa yang disebut rekayasa nilai sebagai "analisis fungsi" [11]. Fungsi berorientasi kepada pelanggan sama seperti nilai. Fungsi diperoleh saat pelanggan mampu mendeskripsikan keinginan mereka dan praktisi dapat mendeskripsikan kebutuhan atau sumber daya [13].

$$\text{Customer Function} = \text{Wants} + \text{Needs/Resource}$$

Kebutuhan atau **needs** adalah sesuatu yang harus dimiliki karena tingkat keperluan dan urgensinya yang tinggi. Jika seseorang memiliki kebutuhan terhadap barang atau jasa, biasanya pertimbangan yang paling penting adalah manfaat yang dapat diambil dari barang atau jasa tersebut beserta fungsinya, sedangkan keinginan atau **wants** bersifat subjektif, tidak terlalu berpengaruh terhadap kehidupan seseorang. Pemenuhan “keinginan” biasanya bersifat kepuasan semata dan

cenderung menyesuaikan dengan selera individu. Keinginan bisa bersifat positif jika pemenuhannya memberikan nilai tambah atau memberi dukungan terhadap pemenuhan kebutuhan yang telah tercapai. Misalnya, seorang pengusaha ingin mengembangkan bisnisnya. Hal ini sangat bernilai positif bagi kemajuan bisnis maupun kesejahteraan karyawan, namun keinginan positif ini harus didukung oleh pemenuhan kebutuhan yang harus terpenuhi seperti efektifitas kerja, tambahan karyawan maupun peningkatan mutu produksi.

### **2.1.3.2 Karakteristik Fungsi**

Fungsi memiliki beberapa karakteristik dasar sebagai berikut :

- Fungsi berorientasi pada penggunaan atau performans
- Sebuah produk dapat memiliki beberapa fungsi
- Fungsi merupakan solusi yang independen
- Masing-masing fungsi pada produk yang sama adalah independen.

### **2.1.3.3 Tipe-Tipe Fungsi**

Terdapat beberapa tipe fungsi sebagai berikut :

1. Fungsi dasar. Fungsi ini terdiri dari fungsi kebutuhan dan fungsi keinginan yang dapat memberikan kontribusi secara langsung terhadap strategi objektif dengan membenarkan adanya produk dan menjamin kinerjanya sesuai dengan harapan [13]. Dalam melakukan rekayasa nilai, fungsi dasar dipertahankan dan tidak dikurangi sebagai konsekuensi untuk mengejar perbaikan nilai. [11].
2. Fungsi pendukung. Fungsi ini berguna untuk melengkapi kebutuhan fungsi dasar. Namun fungsi pelengkap tidak mendukung kinerja teknis produk secara langsung, namun tetap dapat memberikan nilai tambah.
3. Fungsi teknis. Fungsi ini dihasilkan dari desain atau pembuatan produk. Fungsi ini sering tidak disadari oleh pelanggan, namun keberadaannya mendukung kinerja produk.

4. Fungsi kendala. Fungsi ini berkaitan dengan seluruh kendala yang mungkin terjadi terkait lingkungan produk seperti kode, peraturan, standar, situs, keterbatasan teknologi, pasar dan lain-lain. Dalam istilah pelanggan ini berarti “jaminan garansi”.
5. Fungsi yang tidak perlu. Fungsi ini merupakan semua fungsi yang bisa dieliminasi tanpa mempengaruhi kinerja produk. Dalam menjalankan fungsi ini pihak praktisi berperan untuk mengenali dan menentang batasan-batasan yang sudah menjadi kebiasaan.

#### **2.1.4 Optimalisasi**

Optimalisasi adalah hasil yang dicapai sesuai dengan keinginan, jadi optimalisasi merupakan pencapaian hasil sesuai harapan secara efektif dan efisien. Menurut Kamus Besar Bahasa Indonesia, optimalisasi berasal dari kata optimal yang berarti terbaik, tertinggi. Optimalisasi banyak juga diartikan sebagai ukuran dimana semua kebutuhan dapat dipenuhi dari kegiatan-kegiatan yang dilaksanakan. Menurut Winardi (1996) optimalisasi adalah ukuran yang menyebabkan tercapainya tujuan. Secara umum optimalisasi adalah pencarian nilai terbaik dari yang tersedia dari beberapa fungsi yang diberikan pada suatu konteks. Dalam hal ini pencapaian terbaik proyek TI adalah keberhasilan proyek TI yang diukur dengan menggunakan hubungan waktu dan biaya. Dengan mengikuti prinsip optimalisasi proyek, waktu pengerjaan tercepat dan biaya paling murah merupakan pencapaian terbaik (optimal) [4].

#### **2.1.5 Value engineering**

##### **2.1.5.1 Pengertian Value engineering**

*Value engineering* merupakan sebuah metode studi yang sistematis terkait permintaan project yang melibatkan tim dengan kemampuan multi-disiplin untuk melakukan identifikasi, klasifikasi dan analisis untuk mencari keseimbangan fungsi terbaik antara biaya, keandalan dan kinerja sebuah proyek [14]. Metode ini berfokus pada Alasan

dibalik dilakukannya rekayasa nilai adalah jika pemasar mengharapkan produk menjadi lebih praktis atau tidak usang secara gaya dalam jangka waktu tertentu, sehingga rancangan yang dibangun hanya untuk jangka waktu tertentu. Dalam banyak kasus, *Value engineering* ini mengidentifikasi dan menghilangkan biaya yang tidak perlu, sehingga meningkatkan nilai bagi pelanggan [11]. Hasil akhir dari pengimplementasian VE ini berupa *Value Engineering Change Proposals* (VECP) atau proposal usulan perubahan nilai [6].

### 2.1.5.2 Keuntungan *Value engineering*

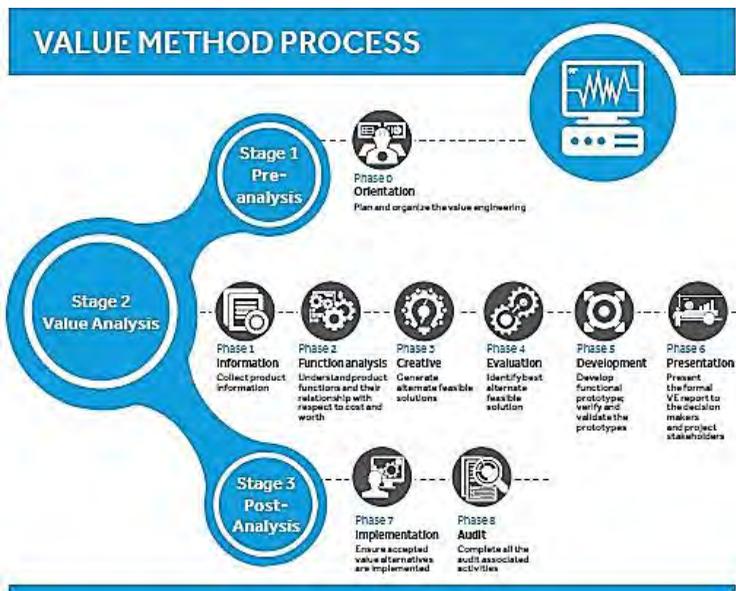
Dengan mempelajari proses, *tools*, teknik dan proses asosiasi pada *Value engineering*, seorang *Value engineer* dapat memperoleh keuntungan sebagai berikut [16] :

- *Value to Sponsor and Stakeholder – Project manger* dan tim dapat mendemokan nilai yang akan diperoleh selama masa pengerjaan proyek.
- *Enhanced Professionalism – Project manger* dan tim dapat mengasah kemampuan, terkhusus pada analisis, penentuan dan pembuktian nilai untuk meningkatkan *image Project manger* dan kemampuan dalam memimpin.
- *Predictabel result* – Proyek menggunakan teknik yang dapat dibuktikan sehingga dapat menjamin hasil yang diperoleh.
- *Repeatabel Success – Project manger* dapat mengulang keberhasilan proyek seperti yang pernah dilakukan sebelumnya, sehingga meningkatkan *image* perusahaan atau organisasi sehingga semakin terpercaya dimata publik.
- *Enhanced business Value – Project manger* dan tim tidak hanya memberikan apa yang diekspektasikan oleh organisasi namun juga memberikan nilai bisnis yang memberikan keuntungan yang terbaik
- *Clear project vision and focus* – dengan level pre-project teknik dan proses, *Project manger* dan *stakeholder* dapat menjalin kerja sama dengan satu tujuan yang sama.

- *Leadership* – dapat membantu memperkuat jiwa kepemimpinan dan kerjasama tim
- *Creativity, innovation and problem solving* – penggunaan *tools* dan teknik untuk memilih solusi terbaik membantu tim untuk kreatif, inovatif dan menemukan hal yang bersifat *problem solving*.
- Penghematan dapat diterapkan ke semua unit produksi
- Pengurangan biaya yang tinggi, biaya produksi berikutnya, dan biaya konsekuensi yang terkait dengan operasi dan dukungan dapat direalisasikan.

### 2.1.5.3 Proses dalam *Value engineering*

Berdasarkan hasil penelitian SAVE Internasional pada tahun 2007, dibutuhkan 3 level penerapan *Value engineering* pada proyek dengan 9 fase proses yang harus dijalankan dalam melakukan perubahan proyek [17].



Gambar 2.2 Proses dalam *Value engineering* [18]

➤ *Pre-Analysis*

Level ini berfungsi untuk melakukan perencanaan analisis *Value* pada level selanjutnya [18]. Terdiri atas 1 fase sebagai berikut :

0. *Orientation Phase*

Level ini berfungsi untuk menambah pemahaman konsep *Value engineering* dan melakukan persiapan dalam pelaksanaan usulan perubahan nilai. Keluaran dari tahap ini adalah kondisi-kondisi terkini terkait dengan perusahaan pemegang proyek.

➤ *Value Analysis*

Level ini berfungsi untuk melakukan eksekusi terhadap perencanaan yang diatur pada level pertama. Level ini terdiri dari 6 fase proses sebagai berikut :

1. *Information Phase*

Fase ini adalah fase pendefinisian masalah. Pada tahap ini semua informasi dari *owner* harus didapatkan untuk mencari tahu *goals* yang ingin dicapai dan sejauh mana batasannya. Selain itu, tahap ini juga menjadi tempat anggaran bagi masing-masing fase dan timeline proyek akan disepakati. Biasanya tahap ini dilakukan dengan wawancara kepada pihak *owner* [16]. Terdapat beberapa data yang dibutuhkan dalam fase ini [14] :

- Desain spesifikasi dan grafik data seperti flow diagram dan schematic yang disepakati untuk menampilkan informasi hasil pengolahan.
- Kode/standard atau *guide* yang berhubungan dan teknik manual
- Anggaran proyek
- Informasi special, data masa lalu, jadwal dan *owner* yang bersangkutan atau permintaan user
- Orang yang akan menerima konsultasi

Fase ini akan menjadi dasar penentuan objek yang akan diteliti, sehingga perlu dipastikan apa yang ingin diselesaikan dengan adanya proyek (menghemat energi, mengurangi biaya, menghemat waktu, meminimalisir perawatan) serta apa yang ingin diberikan kepada pelanggan (kecepatan penerimaan, peningkatan kualitas, pengurangan biaya).

## 2. *Function Analysis Phase*

Pada fase ini dilakukan analisis lebih lanjut terhadap kondisi klien saat ini, untuk mencari tahu fungsi dari permasalahan; program, proyek, sistem, produk, item peralatan, bangunan, fasilitas, layanan, atau pasokan seluruh badan eksekutif yang terkait dengan *goals* project. Fase ini dilakukan dengan 2 teknik analisis, jika pada produk analisis dilakukan dengan analisis pareto dan FAST, jika pada sistem analisis harga masing-masing fungsi dan FAST. Terdapat beberapa pendekatan yang bisa dilakukan untuk melakukan analisis fungsi terkait permasalahan [13].

- Mendefinisikan fungsi permintaan terkait permasalahan, dan kemudian mendata seluruh komponen yang berhubungan dengan masing-masing fungsi.
- Melakukan pendataan fungsi dari objek permasalahan yang dapat membantu mewujudkan sasaran yang ingin dicapai atau mengembangkan fungsi yang telah ada.

Contoh analisis komponen fungsi dapat dilihat pada gambar berikut ini.

Project: Install Equipment

COMPONENT	COST	FUNCTION
Foundation		Support Weight
Electrical Panel		Distribute Power      Protect Personnel      Protect Equipment
Equipment Room		Furnish Space      Shelter Equipment
Air Conditioning Unit		Control Environment      Condition Air

Gambar 2.3 Contoh Analisis Komponen Fungsi

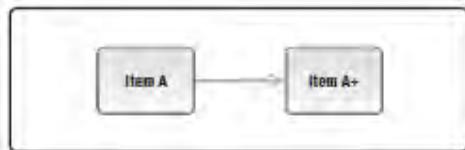
### 3. *Creativity Phase*

Fase ini akan dilakukan untuk pencarian alternatif solusi yang memungkinkan untuk diterapkan dan dikembangkan sebagai solusi permasalahan perusahaan. Pencarian solusi didasarkan pada *goals* (Fase 1) dan fungsi dari suatu program, proyek, sistem, produk, item peralatan, bangunan, fasilitas, layanan, atau pasokan seluruh badan eksekutif terkait (Fase 2). Kebanyakan tahapan ini dilakukan dengan teknik brainstorming dengan didasarkan pada menentukan atau menjawab beberapa pertanyaan berikut dengan tepat [13].

- Dapatkah fungsi dari suatu program, proyek, sistem, produk, item peralatan, bangunan, fasilitas, layanan, atau pasokan seluruh badan eksekutif tersebut dikurangi, dimodifikasi atau dikombinasikan atau tidak?
- Apakah spesifikasi pengurangan biaya bisa dimodifikasi?
- Apakah desain dan konstruksi metode pengurangan biaya dapat disederhanakan?
- Apakah biaya operasi atau maintenance dapat dikurangi seiring dengan pergantian desain?

Fase tersebut dapat diimplementasikan pada dua jenis perubahan orientasi fungsi, yakni :

- Item-Oriented *Analysis*

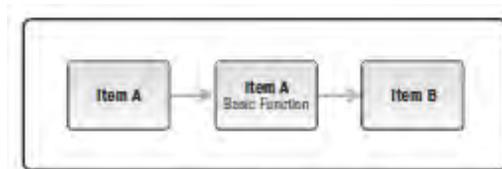


Gambar 2.4 Item-Oriented *Analysis*

Orientasi ini melakukan *Analysis* terhadap item dengan menjawab pertanyaan “Bagaimana item A dapat ditingkatkan?”. Hasilnya adalah modifikasi dari item tersebut. Contohnya pada aplikasi *google maps*, pada tahun 2017 *google maps* memiliki fitur penunjuk jalan bagi

pengendara mobil, pejalan kaki dan transportasi umum. Pada tahun 2018, terdapat fitur baru berupa fitur penghubung *google maps* dengan aplikasi booking kendaraan online. Dimana saat user pencari sebuah lokasi pada *google maps*, user dapat langsung melakukan booking kendaraan online melalui fitur tersebut sesuai dengan aplikasi yang mereka miliki dan harga yang mereka inginkan. Karena pada fitur baru tersebut, ditampilkan pula harga tempuh untuk masing-masing jenis kendaraan online.

- *Function-Oriented Analysis*



Gambar 2.5 *Function-Oriented Analysis*

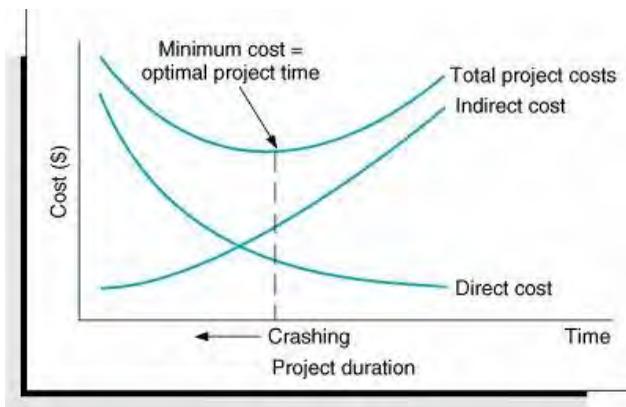
Orientasi ini melakukan *Analysis* terhadap fungsi item dengan menjawab pertanyaan “Bagaimana fungsi dasar item tersebut dapat memberikan nilai yang lebih baik?” atau dengan kata lain mencari tahu fungsi dasar item tersebut dan mengembangkannya menjadi fungsi baru yang dapat menghasilkan nilai yang lebih baik. Contohnya pada aplikasi *google translate*, fungsi dasarnya adalah menterjemahkan kalimat dalam bentuk tulisan word, pdf, maupun *input* manual pada field *google translate* kedalam bahasa yang diinginkan. Namun jika fungsi tersebut diubah menjadi menterjemahkan gambar maupun kalimat dalam gambar kedalam bahasa yang diinginkan maka aplikasi yang baiknya digunakan adalah Snap & Translate.

Keluaran dari fase ini berupa skenario-skenario yang memungkinkan untuk dikembangkan pada tahapan berikutnya. Pembentukan skenario ini tidak terbatas jumlahnya. Tim dapat membuat skenario apa saja

sebanyak-banyaknya terkait objek kajian. Dimana skenario-skenario tersebut akan dimasukkan kedalam VECP secara lengkap

#### 4. *Evaluation Phase*

Pada fase ini dilakukan evaluasi alternatif solusi yang diperoleh pada fase 3. Masing-masing solusi tersebut akan dilakukan perhitungan indeks nilainya. Berdasarkan hasil yang diperoleh, solusi yang tidak berpeluang untuk dijalankan akan dieliminasi dan solusi dengan biaya implementasi yang tinggi akan diolah kembali untuk melakukan reduksi biaya terlebih dahulu sebelum akhirnya dieliminasi jika biaya yang dikeluarkan tetap tergolong tinggi, hingga akhirnya menghasilkan solusi yang memberikan nilai penghematan tertinggi untuk dikembangkan pada tahapan selanjutnya. Berdasarkan bukti kelayakan tersebut, dengan menggunakan prinsip rekomendasi optimal proyek TI, tentang hubungan biaya dan waktu pelaksanaan proyek (**Gambar 2.5**) [19].



Gambar 2.6 Hubungan Biaya dan Waktu Pelaksanaan Proyek

Hasil yang diperoleh dengan spesifikasi waktu pengerjaan paling cepat dan biaya paling rendah akan dijadikan rekomendasi.

## 5. Development *Phase*

Pada fase ini akan dilakukan re-analisis terhadap solusi yang telah dipilih pada fase 4. Re-analisis ini akan berfungsi untuk mengembangkan solusi yang telah dipilih, untuk meningkatkan kualitas fungsi alternatif solusi yang dipilih. Pengembangan solusi ini dilakukan dengan beberapa tahapan sebagai berikut [13] :

- Memastikan bahwa alternatif yang dipilih sudah memenuhi kebutuhan user.
- Menentukan teknis implementasi.
- Mengembangkan kebutuhan biaya dan jadwal untuk implementasi dari alternatif yang terpilih.

Pada tahap ini *Project manger* akan memastikan bahwa ide yang sudah dikembangkan dan diterima akan diimplementasikan. *Project manger* akan mempersiapkan perencanaan implementasinya dan memberikan instruksi bagi orang-orang yang akan memegang tanggungjawab pada proyek tersebut.

Hasil akhir fase ini berupa perencanaan implementasi proyek, list prosedur dan *milestone* yang akan digunakan oleh *Project manger* sebagai *control system* untuk memastikan bahwa solusi yang akan mereka terapkan berjalan dengan baik dan sesuai dengan *timeline*.

## 6. Presentation *Phase*

Fase ini merupakan fase dimana hasil dari ke-lima fase pada tahapan *Value Analysis* di presentasikan kepada pihak yang berkepentingan, secara terperinci, sehingga pihak ketiga dapat memahami aktivitas yang akan mereka lakukan dan berapa sumber daya yang mereka butuhkan dalam mewujudkan hasil analisis tersebut.

Selanjutnya, pada fase ini akan dilakukan penyerahan VECP kepada pihak ketiga untuk disetujui dan diarsipkan sebagai

bukti dan acuan pengimplementasian usulan hasil *Value engineering*

➤ *Post-Analysis*

Level ini berfungsi untuk menjalankan rencana implementasi proyek serta membuat dokumentasi dan presentasi hasil pengimplementasian alternatif solusi yang terpilih dan telah disetujui. Level ini terbagi atas 2 fase sebagai berikut [18]:

7. *Implementation Phase*

Fase ini adalah fase dimana pihak pemegang proyek melaksanakan hasil pada tahap *Value Analysis* sesuai dengan VECP yang diterima.

8. *Audit Phase*

Fase ini adalah fase untuk melakukan audit terhadap aktivitas yang tertulis pada VECP. Lama suatu aktivitas dapat berubah sesuai dengan kemampuan masing-masing SDM yang tersedia dalam mengolah informasi yang diperoleh pada setiap tahapan *Analysis*.

Namun pada pengerjaan Tugas Akhir ini, penulis hanya mengimplementasikan level *Value Analysis* dari tahap *Information Phase* sampai pada *Evaluation Phase*. Hingga menghasilkan Buku Tugas Akhir.

## **2.1.6 *Value Engineering Change Proposal (VECP)***

### **2.1.6.1 *Pengertian Nilai***

Penyusunan VECP harus didasari oleh proposal konseptual atau dasar sebagai pembanding. Tujuan utama adanya *Value engineering* adalah mendorong pemegang proyek dan pihak ketiga untuk mencapai alternatif namun tetap mempertimbangkan jadwal proyek, perkiraan biaya desain, estimasi penghematan, dan manfaat dan dampak dibandingkan dengan konsep awal [20].

### 2.1.6.2 Konten/Isi dalam VECP

Dalam penyusunannya, VECP tidak memerlukan metode khusus, intinya VECP harus berisi hal berikut ini.

1. Informasi-informasi alternatif yang ada yang sudah dikonversi kedalam bentuk harga.
2. Bahan atau peralatan yang digunakan
3. Perkiraan penghematan

Untuk memastikan kelengkapan *Value Engineering Change Proposal* (VECP) *checklist* berikut ini dapat digunakan sebagai acuan [21].

1. Apakah VECP yang anda susun telah menggambarkan perbedaan antara skenario awal dengan modifikasi yang diusulkan, yang keuntungan dan kerugian relatifnya juga telah dipertimbangkan dan menunjukkan adanya perubahan ? Hal ini harus dijelaskan secara rinci
2. Sudahkah VECP tersebut berisi spesifikasi rinci tentang fungsi yang akan diubah?
3. Apakah VECP tersebut telah berisi perkiraan penghematan biaya sesuai rekomendasi anda dibandingkan dengan konsep awal?
4. Apakah biaya pengujian dan evaluasi serta biaya operasi dan dukungan lain telah dituliskan didalam VECP anda?
5. Sudahkah anda menentukan kapan waktu kembalinya keuntungan dari pengimplementasian solusi yang anda tawarkan?
6. Sudahkah anda secara singkat mengidentifikasi semua pengajuan VECP lainnya sebelumnya?
7. Sudahkah anda melakukan verifikasi proposal anda untuk memastikan akurasi dan kelengkapannya?

Jika anda telah melakukan beberapa list diatas, maka dapat dipastikan bahwa VECP anda sudah lengkap dan dapat memberikan informasi yang jelas kepada pihak penerima VECP. Berikut ini adalah contoh VECP pada Proyek Konstruksi [19].

## VALUE ENGINEERING CHANGE PROPOSAL SUMMARY

VECP SUBMISSION DATE : 09/12/14

VECP SUBMISSION (Check One) :  CONCEPTUAL  FORMAL  JOINT CONCEPTUAL & FORMAL

---

### CONTRACT INFORMATION

CONTRACT ID# : D262468	LETTING DATE: 07/13/14
PIN : 6543 21	ORIGINAL CONTRACT BID : \$16,765,852.43
DESCRIPTION: Replacement of Rte 3 Bridges over CSX RR & Yellow Brick Rd	REGION : 6
CONTRACTOR : Highways R Us Corp	COUNTY : Niagara
	FEDERAL AID # : 12-3456789

---

### VECP INFORMATION

**SHORT DESCRIPTION OF VECP :** Reduction of the MSES Wall Limits And Constructing 2H:1V Side Slopes

(A.) TYPE OF VECP? (Check One)  COST SAVINGS  TIME SAVINGS ONLY(B.) IS THERE A DATE BY WHICH THE VECP WORK MUST BE AUTHORIZED?:  YES  NO DATE?: 12/25/14(C.) ANY NEW OR EXISTING PAY ITEMS REQUIRING AGREED PRICES?:  YES  NO HOW MANY?: 1(D.) ANY PAY ITEMS WITH LONG LEAD TIMES THAT REQUIRE APPROVAL?:  YES  NO HOW MANY?: 1

(Note: Describe items A. through D. in further detail in &gt;COMMENTS&lt; as appropriate.)

(E.) CONTRACT COST W/O VECP:	\$16,765,852.43	(Note: This is the latest cost as of the VECP submission date)
(F.) CONTRACT COST W/ VECP*:	\$16,762,101.31	(* Note: Excludes any VECP savings or design cost reimbursements)
(G.) VECP CONSTRUCTION SAVINGS:	\$3,751.12	(Note: Equals item E. minus item F.)
(H.) VECP DESIGN COST:	\$0.00	
(I.) DIRECT COST SAVINGS:	\$3,751.12	(Note: Equals item G. minus item H.)
(J.) NET SAVINGS TO STATE:	\$1,875.56	(Note: Equals 0.5 times item I.)
(K.) TOTAL ADJUSTED CONTRACT COST:	\$16,763,976.87	(Note: Equals item E. minus item J.)
(L.) VECP SAVINGS REIMBURSEMENT TO CONTRACTOR:	\$1,875.56	(Note: Equals 0.5 times item G.)
(M.) VECP DESIGN REIMBURSEMENT TO CONTRACTOR:	\$0.00	(Note: Equals 0.5 times item H.)

---

### COMMENTS

1. Cost savings to the State will come from reduction in total area of MSES Wall by 44% (10,112 SF). See revised shop drawings reflecting the VECP change are already uploaded to Contract Manager for review and approval.
2. The date for VECP authorization is by the end of December because of time for Wall shop drawing approval and the fabrication lead time, otherwise the overall project schedule may be impacted.
3. Existing Item 203.03 would be one of the Items needing an agreed price for the additional amount of 25,250 CY of Embankment over the existing original quantity. The price noted on the attached VECP Itemized Construction Savings Sheet, would be proposed at \$5.00 per CY. This item would encompass the embankment needed in lieu of backfill for the wall not being installed, for the northbound 2:1 slope and for the southbound 2:1 slope. Will use existing unit prices for the increased quantity of Item 610.0203 - Establishing Turf and Item 613.03 - Placing Topsoil Type B, to cover the new embankment area.
4. Item 554.0101 - Mechanically Stabilized Earth System, No Color, Plain Concrete would be the existing item which would require a long lead time.
5. The unit cost for Item 554.0101 - Mechanically Stabilized Earth System would need to be increased by \$4.50/SF to become \$46.00/SF. This unit cost increase is explained in the attached letter from the manufacturer.
6. No design cost would be applicable to this VECP.

Gambar (a)

## VECP - DESIGN COST

CONTRACTOR : Geed Guys Construction Corp

CONTRACT D#: 0262001

VECP DESCRIPTION : On-site Detour Crossovers

DATE : 2/14/2014

	DESCRIPTION	RATE	HOURS	TOTALS
1	<b>Arhos, Porthos &amp; Aramis Consulting LLC</b>			
2	DIRECT SALARY COSTS :			
3	Detour Design	\$75.00 / HR	15	\$1,125.00
4	CADD/Drafting	\$35.00 / HR	10	\$350.00
5		/ HR		\$0.00
6			TOTAL LABOR	\$1,475.00
7	OVERHEAD/PROFIT @ %	10		\$147.50
8			TOTAL DIRECT SALARY COSTS	\$1,622.50
9				
10	DIRECT NON-SALARY COSTS :			
11	Printing/Photocopying			\$250.00
12				
13			TOTAL NON-DIRECT SALARY COSTS	\$250.00
14				
15			CONSULTANT #1 TOTAL	\$1,872.50
16				
17				
18				
19				
20				
21	<b>L.R. Tonfo Hydrological Services</b>			
22	DIRECT SALARY COSTS :			
23	Stormwater Design	\$100.00 / HR	7	\$700.00
24			TOTAL DIRECT SALARY COSTS	
25	DIRECT NON-SALARY COSTS :			
26			TOTAL NON-DIRECT SALARY COSTS	\$0.00
27				
28			CONSULTANT #2 TOTAL	\$700.00
29				
30			COMBINED DESIGN COSTS	\$2,572.50
31			6% PROFESSIONAL SERVICE FEE	\$128.13
32				
33			LUMP SUM VECP DESIGN COST	\$2,701.13
34				
35				
36				
37				
38				
39				
40				
41				
42				
43				
44				
45				

Gambar (b)

**VECP - ITEMIZED CONSTRUCTION SAVINGS**  
**VECP DESCRIPTION: TIE BACK WALL**

SHEET 1 of 1

CONTRACT NO: D081234  
 DATE: 12/13/14

LINE	ITEM SPEC #	ITEM DESCRIPTION	COMMENTS	UNIT PRICE	UNIT OF MEASURE	PRIOR APPROVED QUANTITY	CHANGE IN QUANTITY	CHANGE IN FUNDS (\$)	
								INCREASE	DECREASE
1	x 552.13	TEMPORARY STEEL SHEETING		\$115.000 / SM		15,090.00	-893.00		\$79,000.00
2	x 211.10	GRouted TIE BACKS (TEMPORARY)		\$0.000.000 / EA		102.00	-85.00		\$360,000.00
3	x 552.13	TEMPORARY STEEL SHEETING		\$115.000 / SM		15,090.00	248.00	\$28,520.00	
4	x 211.10	GRouted TIE BACKS (TEMPORARY)		\$0.000.000 / EA		102.00	34.00	\$204,000.00	
5	x 203.02	UNCLASSIFIED EXCAVATION		\$16.000 / CM		399,543.00	2,900.00	\$44,800.00	
6	x 203.21	SELECT STRUCTURAL FILL		\$40.000 / CW		29,821.00	310.00	\$12,400.00	
7	x 203.03	EMBANKMENT IN PLACE		\$6.000 / CM		383,385.00	2,703.00	\$16,200.00	
8				/					
9				/					
10				/					
11				/					
12				/					
13				/					
14				/					
15				/					
16				/					
17				/					
18				/					
19				/					
20				/					
21				/					
22				/					
<b>SUBTOTALS</b>								<b>\$305,820.00</b>	<b>\$405,000.00</b>
<b>VECP CONSTRUCTION SAVINGS</b>									<b>\$159,560.00</b>

**NOTES:**

- Do not include the VECP design cost or VECP construction savings pay items.
- Place a "0" in either the "M" column for new pay items or in the "C" column for existing pay items.
- Under "COMMENTS," list as applicable: "replaced piece," "existing piece," or any other descriptive comment to clarify the entry.
- For pay items with proposed agreed prices, use an "A" prefix for both the "Item Spec#" and the "Item Description".
- For proposed agreed pieces of existing pay items, zero out the existing pay item and put it in under a new pay item (use the existing pay item with a "A" prefix, ex. "A-819.17") at the new unit price.

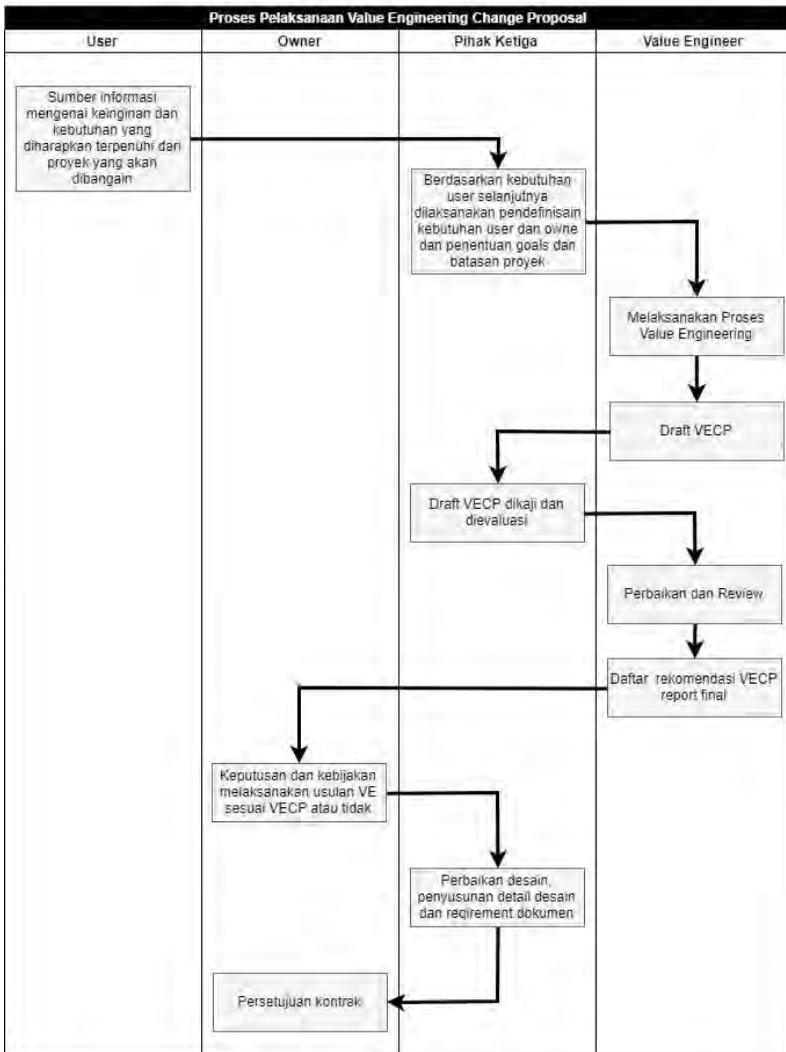
Gambar (c)  
 Gambar 2.7 (a) (b) dan (c) contoh *Value Engineering Change Proposal* pada Proyek Kontruksi

### 2.1.6.3 Para Pihak yang terlibat dalam VECP

Para pelaksana yang terlibat dalam VECP adalah sebagai berikut [22]

1. *User* (Pengguna) dalam hal ini atlet yang akan menggunakan aplikasi yang akan dibangun pada proyek ini. Sebagai sumber atau objek yang diteliti untuk mencari tahu keinginan dan kebutuhan yang diharapkan terpenuhi melalui proyek yang akan dibangun.
2. *Owner* (Pemilik/Pemberi tugas) adalah pihak yang menginvestasikan sejumlah dana untuk mewujudkan proyek pengembangan aplikasi, serta sebagai pihak akan yang memegang wewang untuk mengambil keputusan atau kebijakan untuk melaksanakan hasil *Value engineering* yang diperoleh atau tidak.
3. Pihak Ketiga adalah pihak yang ditugaskan untuk *owner* selaku pemilik untuk mengatur hubungan kerja antara pihak yang terlibat dalam proyek, pihak yang memberikan desain aliran informasi aplikasi awal yang akan menjadi dasar dan pembanding dengan apa yang akan *Value engineer* usulkan, serta menjadi pihak yang akan melakukan brainstorming kepada pihak *Value engineer* dalam mencari usulan yang paling optimal.
4. *Value engineer* adalah pihak yang melakukan studi VE dan melakukan analisis penghematan biaya terhadap objek penelitian untuk kemudian menghasilkan usulan perubahan, dan menyusun *Value Engineering Change Proposals* (VECP).

### 2.1.6.4 Proses Pelaksanaan VECP



Gambar 2.8 Proses Pelaksanaan *Value Engineering Change Proposal*

Pada pengerjaan Tugas Akhir ini, penulis hanya mengikuti proses dari tahapan melaksanakan proses *Value engineering* hingga tahapan pengerjaan Daftar Rekomendasi VECP report final dan menyerahkannya kepada pihak ketiga. Selanjutnya proses akan dilakukan antara pihak ketiga dengan *owner*.

### 2.1.7 Perbandingan Metode

Dalam menjalankan proyek, terdapat beberapa metode yang lain yang dapat digunakan dalam *manage* setiap tahapan proyek seperti *Value management* dan *investiation management*. Masing-masing metode memiliki karakteristik yang berbeda-beda. Berikut ini merupakan perbandingan metode *Value engineering* dengan *Value management* dan *investiation management*.

Tabel 2.1 Perbandingan *Value engineering* dengan *Value Management* dan *Investation Management*

<i>Characteristics</i>	<i>Value engineering</i>	<i>Value Management</i>	<i>Investation Management</i>
<i>Typical Objectives</i>	Mengurangi biaya pengeluaran tanpa mengorbankan kualitas atau performa.	Mengembangkan panduan untuk perencanaan dan desain pada level briefing pada proyek.	Menempatkan uang atau dana pada hal atau tempat tertentu dengan harapan memperoleh keuntungan atau tambahan tertentu atas dana tersebut.
	Dapat menentukan pilihan yang terbaik dari berbagai pilihan yang ada.	Pemilihan konsep desain yang terbaik dari range pilihan yang ada	Dapat memilih, mengontrol dan mengevaluasi proposal investasi terbaik secara berkala dari pilihan proposal yang ada.

<i>Characteristics</i>	<i>Value engineering</i>	<i>Value Management</i>	<i>Investation Management</i>
	Pemilihan tipe komponen	Mengembangkan proposal untuk meningkatkan nilai uang pada konsep atau level detail desain	
		Menyelesaikan isu pada tahap perencanaan dan desain	
<i>Stage of project development</i>	Diimplementasikan pada tahap pembuatan konsep atau desain dan sebagainya.	Diimplementasikan pada tahapan yang paling awal dari pengembangan, jika memungkinkan sebelum briefing project selesai dipersiapkan	Diimplementasikan sebelum, saat dan sesudah investasi dilakukan (Herdianti, H., Nugtoho, A.A., dkk. 2011).
	Dapat digunakan pada saat proyek sudah disepakati dan akan mulai dikerjakan namun berpotensi untuk anggaran dan waktu yang melebihi batas.		Dapat menentukan waktu untuk menginvestasikan dan mendisvestasikan investasi nasabah.

<i>Characteristics</i>	<i>Value engineering</i>	<i>Value Management</i>	<i>Investation Management</i>
<i>Participants</i>	Fokus kepada sumber daya yang berhubungan dengan analisis.	Mebutuhkan partisipasi dari pihak manajemen, perencanaan strategis hingga operasional.	Fokus pada individu atau organisasi yang ingin melakukan investasi dan sumber daya manusia yang memiliki kemampuan akuntansi.
<i>Number of Participants</i>	Berkisar antara 8 sampai 15 orang.	Berkisar antara 15-25 tapi terkadang berkisar antara 40-50.	Tidak ada ketentuan.
<i>Function Analysis</i>	Analisis fungsi konvensional dari masing-masing fungsi yang akan dianalisis.	Tujuan dasar; Hasil keluaran, karakter utama yang harus didapatkan – pada masing-masing tahapan di semua level.	Analisis proses kritis yang dinilai, pengelolaan investasi yang diraih dan dasar pemikiran berbasis bukti yang memiliki peluang terbaik untuk memberikan keuntungan berkelanjutan (Publications Caalog. GAO. 2017).
	Dapat melakukan penghematan biaya pembangunan aplikasi, biaya konsekuensi	Mebutuhkan analisis fungsi permintaan yang detail	

<i>Characteristics</i>	<i>Value engineering</i>	<i>Value Management</i>	<i>Investation Management</i>
	terkait operasi dan dukungan.		
<b><i>Cost comparisons</i></b>	Dapat memberikan kepastian dari penggunaan biaya dan model dari apa yang dijalankan.	Memungkinkan untuk memberikan kepastian, namun umumnya bersifat komperatif (leboh besar dari/ kurang dari) jika harga dapat dipahami – cth. 50% lebih baik dari yang ini.	Dapat memberikan dan membandingkan perhitungan waktu pengembalian modal investasi (Herdianti, H., Nugtoho, A.A., dkk. 2011)..
		Tidak membutuhkan dana pada tahapan awal. Dapat menjadi dasar pembangunan proyek.	

**Sumber :** *Value engineering vs Value Management* [23]

Berdasarkan perbandingan tersebut, Dr. Roy Barton salah satu tim ACVM (Australian Centre for Value management) mengatakan bahwa *value engineering* dan *value management* sering disamakan dan biasanya digunakan secara bersamaan. Namun *value engineering* sangat erat kaitannya dengan capaian penghematan biaya (*Value engineering* secara tradisional didefinisikan untuk menemukan harga terendah dari kebutuhan fungsional yang diinginkan), sedangkan *value management* tidak hanya focus pada penghematan biaya, namun lebih banyak berbicara tentang nilai lain yang lebih luas dari sebuah

proyek, sedangkan investment management lebih banyak berbicara tentang bagaimana melakukan investasi dengan baik dan melakukan peramalan terhadap hasil investasi. Sehingga jika ingin mencapai tujuan penghematan biaya sebaiknya dilakukan dengan memanfaatkan metode *Value Engineering* [23].

## **2.1.8 Teknik-Teknik Terkait**

### **2.1.8.1 *Benchmarking***

*Benchmarking* merupakan suatu strategi mengadaptasi proses-proses serta produk-produk perusahaan terbaik untuk ditiru di lingkungan perusahaan sendiri dengan melakukan beberapa modifikasi sesuai dengan kebutuhan, dengan tujuan untuk memperbaiki performa organisasi [24]. Terdapat beberapa jenis *Benchmarking* sebagai berikut.

#### 1. *Benchmarking* Internal

Merupakan tipe *Benchmarking* paling sederhana karena hanya mencakup kegiatan dengan operasi-operasi internal atau dengan kata lain antar divisi pada satu perusahaan.

#### 2. *Benchmarking* Industri

Merupakan tipe *Benchmarking* yang dilakukan dalam lingkup antar perusahaan dalam satu industri yang sama. *Benchmarking* ini membandingkan antar perusahaan yang mengandung ciri teknologikal dari pasar yang sama.

#### 3. *Benchmarking* Proses

*Benchmarking* ini proses yang paling sulit dilaksanakan karena perusahaan yang terlibat hanya memiliki sedikit persamaan, sebab prinsipnya dilakukan pada pemimpin-pemimpin atau perusahaan-perusahaan terbaik tanpa memperdulikan apakah perusahaan tersebut satu industri atau tidak.

### 2.1.8.2 *Function point*

*Function point* merupakan sebuah metode pengukuran volume *software* tidak berdasarkan pada banyaknya baris kode program namun lebih kepada suatu hal yang dapat diukur lebih awal yakni banyak dan kompleksnya *software* yang dibangun pada SDLC sehingga dapat menyediakan perkiraan sumber daya yang dibutuhkan perusahaan *software* untuk mempersiapkan proposal tender dan project plan. Adapun tahapan dalam menghitung *Function point* adalah sebagai berikut [25].

#### **Langkah 1. Menghitung *Crude Function point* (CFP)**

*Crude Functional Point* salah satu metode analisis dengan melakukan perhitungan bobot dari masing-masing komponen *software* atau aplikasi yang akan dibuat. Terdapat 5 tipe komponen yang termasuk kedalam analisis tersebut, yakni :

- ***External Input (EI)***, berhubungan dengan data dengan aplikasi yakni fungsi yang memindahkan data ke dalam aplikasi tanpa mempresentasikan manipulasi data.
- ***External Output (EO)***, berhubungan dengan data dan pengguna yakni fungsi yang memindahkan data ke pengguna dengan menyajikan beberapa manipulasi data.
- ***Internal Logical File (ILF)***, berhubungan dengan data ke pengguna yakni fungsi yang memindahkan data ke pengguna tanpa mempresentasikan manipulasi data.
- ***External Interface Files (EIF)***, berhubungan dengan logika data yakni logika berupa data tetap yang dikelola oleh aplikasi melalui penggunaan *External Input (EI)*.
- ***External Inquiry (EI)***, berhubungan dengan logika data yakni logika berupa data tetap yang digunakan oleh aplikasi namun tidak berjalan di dalamnya.

Kelima tipe tersebut akan diisikan sesuai dengan kebutuhan pada aplikasi yang akan diteliti. Dalam analisis CFP, kelima tipe tersebut akan dikategorisasi kedalam 3 jenis pembobotan,

yakni *Simple*, *Average* dan *Complex* dengan nilai kompleksitas pembobotan CFP seperti pada **Tabel 2.2**.

Tabel 2. 2 Tabel Bobot Kompleksitas Crude *Function points*

Tipe Fungsi	Bobot Kompleksitas		
	Simple	Average	Complex
External Inputs	3	4	6
External Output	4	5	7
Internal Logical File	7	10	15
External Interface Files	5	7	10
External Inquiry	3	4	6

Nilai-nilai Bobot dari setiap komponen diatas adalah ketetapan atau konstanta yang dibuat oleh *Function point* Internasional User Group (IFPUG).

### **Langkah 2. Menghitung *Relative Complexity Adjustment Factor (RCAF)***

RCAF berfungsi untuk menghitung kesimpulan kompleksitas teknis dari suatu sistem *software* dari perspektif subjek yang paling berpengaruh terhadap usaha pengembangan yang dibutuhkan yang didasarkan pada 14 karakteristik sebagai berikut.

1. *Data Communications* – Didefinisikan sebagai tingkat kebutuhan komunikasi langsung antara aplikasi dengan *processor*

2. *Distributed Functions* – Didefinisikan sebagai tingkat kebutuhan transfer data antara komponen – komponen aplikasi
3. *Performances Objectives* – Didefinisikan sebagai tingkat *response time* dan *throughput* yang perlu dipertimbangkan dalam pengembangan aplikasi
4. *Heavily Used Configuration* – Didefinisikan sebagai tingkat kebutuhan, dimana *setting* konfigurasi computer berpengaruh terhadap pengembangan aplikasi
5. *Transaction Rate* – Didefinisikan sebagai tingkat transaksi bisnis yang berpengaruh terhadap pengembangan aplikasi
6. *On Line Data Entry* – Didefinisikan sebagai tingkat kebutuhan *input* data secara interaktif
7. *End User Efficiency* – Didefinisikan sebagai tingkat kemudahan penggunaan aplikasi
8. *On Line Update* – Didefinisikan sebagai tingkat kebutuhan ILF diperbaharui secara interaktif
9. *Complex Processing* – Didefinisikan sebagai tingkat kesulitan logika proses yang mempengaruhi proses *development*
10. *Reusability* – Didefinisikan sebagai tingkat kebutuhan aplikasi dan ode program dirancang dan dikembangkan untuk bisa digunakan pada aplikasi lain
11. *Installation Ease* – Didefinisikan sebagai tingkat kemudahan koversi ke sistem baru yang berpengaruh pada proses *development*
12. *Operational Ease* – Didefinisikan sebagai tingkat kemudahan aplikasi dalam aspek-aspek operasional seperti *startup*, *backup*, dan proses *recovery*
13. *Multiple Sites* – Didefinisikan sebagai tingkat kebutuhan aplikasi dapat dioperasionalkan pada lingkungan *hardware* dan *software* yang berbeda-beda
14. *Facilitate Change* – Didefinisikan sebagai tingkat kemudahan aplikasi untuk modifikasi logika proses maupun struktur data

Seluruh karakteristik tersebut akan dinilai dengan menggunakan skala 0-5 dengan ketentuan sebagai berikut.

0 = Tidak Pengaruh

1 = Insidental

2 = Moderat

3 = Rata-Rata

4 = Signifikan

5 = Essential

Masing-masing kriteria tersebut selanjutnya dimasukkan kedalam **Tabel 2.3** yang akan digunakan sebagai pedoman dalam menghitung *tabel* RCAF.

Tabel 2.3 Karakteristik *Software*

No	Karakteristik	Bobot
1.	Tingkat kompleksitas Komunikasi Data	[0/1/2/3/4/5]
2.	Tingkat kompleksitas Pemrosesan Terdistribusi	[0/1/2/3/4/5]
3.	Tingkat kompleksitas Pergormance	[0/1/2/3/4/5]
4.	Tingkat kompleksitas konfigurasi	[0/1/2/3/4/5]
5.	Tingkat frekuensi penggunaan <i>software</i>	[0/1/2/3/4/5]
6.	Tingkat frekuensi <i>Input</i> Data	[0/1/2/3/4/5]
7.	Tingkat Kemudahan penggunaan bagi <i>user</i>	[0/1/2/3/4/5]
8.	Tingkat frekuensi <i>update</i> data	[0/1/2/3/4/5]
9.	Tingkat kompleksitas Prosessing Data	[0/1/2/3/4/5]
10.	Tingkat kemungkinan penggunaan kembali	[0/1/2/3/4/5]

No	Karakteristik	Bobot
11.	Tingkat kemudahan dalam instalasi	[0/1/2/3/4/5]
12.	Tingkat kemudahan operasional <i>software</i> ( <i>backup, recovery, dsb</i> )	[0/1/2/3/4/5]
13.	Tingkat <i>software</i> dibuat untuk multi organisasi/perusahaan/ <i>client</i>	[0/1/2/3/4/5]
14.	Tingkat kompleksitas dalam mengikuti perubahan/fleksibel	[0/1/2/3/4/5]
	<b>Total</b>	?

Pemberian bobot dengan skala 0-5 pada masing-masing factor ini mengacu pada *Standar Documentation* bagi *General System Characteristic (GSC)* yang dipublikasikan oleh *Software Matric*.

### Langkah 3. Menghitung *Function point (FP)*

Setelah perhitungan CFP dan RCAF dilakukan selanjutnya dapat dilakukan perhitungan FP. Perhitungan *Function point* dilakukan dengan memasukkan nilai kedalam persamaan berikut.

$$\mathbf{FP = CFP \times (0.65 \times 0.01 \times RCAF)}$$

Dimana nilai CFP diperoleh dari hasil pada langkah 1 yang merupakan total dari tabel *Function point* analitis dan nilai RCAF diperoleh dari langkah 2 yang merupakan total dari penilaian karakteristik kompleksitas teknis *software*. Adapun angka 0.65 dan 0.01 adalah ketetapan atau konstanta yang dibuat oleh International Funtion Point User Group (IFPUG).

#### Langkah 4. Estimasi *Effort*

Setelah diperoleh hasil perhitungan *Function point*, selanjutnya dilakukan perhitungan usaha dengan mencari tahu jenis *software* yang akan dibangun, kemampuan yang dimiliki oleh perusahaan atau pihak pemegang proyek dan biaya atau gaji dalam satuan Rupiah untuk setiap harinya atau dikenal dengan istilah *ManDays*. Estimasi *effort* dihitung dengan memasukkan nilai pada persamaan berikut.

$$m = f^{3*j} / 27$$

Dengan menggunakan persamaan tersebut, dapat diperoleh jumlah sumber daya manusia yang paling efektif untuk mengerjakan proyek aplikasi yang akan dibangun. Perhitungan ini menggunakan fungsi estimasi eksponensial oleh *Jones* dengan  $f$  merupakan nilai *Function point*,  $j$  merupakan nilai *Jone's First Order* (lihat **Tabel 2.4**).

Tabel 2.4 *Jone's First Order Estimate Exponent*

Kind of <i>software</i>	Organization's Skill/Abilities		
	Best In Class	Average	Worst In Class
<i>System</i>	0.43	0.45	0.48
Business	0.41	0.43	0.46
Shrink-wrap	0.39	0.42	0.45

Jika ingin mengetahui lama pembangunan proyek, dengan memanfaatkan *Jone's First Order Estimate Exponent*, nilai yang sesuai dengan aplikasi dapat dimasukkan kedalam persamaan berikut.

$$s = f^j$$

Dengan  $s$  merupakan *schedule months for optimal schedule*,  $f$  merupakan nilai *effort* dan  $j$  merupakan nilai *Jone's First Order*.

### Langkah 5. Estimasi Biaya

*Effort* yang telah didapatkan pada tahap sebelumnya dibagi menjadi tiga aktivitas yang dijalankan dalam pengembangan perangkat lunak yaitu:

- 1) *Software Development*, meliputi analisis kebutuhan pengguna seperti permintaan dan spesifikasi, desain, implementasi, pengujian integrasi yang mencapai sekitar 47% dari total *effort*;
- 2) *Ongoing Activity*, atau disebut aktivitas yang berkesinambungan seperti manajemen proyek, manajemen konfigurasi, dokumentasi, penerimaan dan penyebaran yang mencapai sekitar 16% dari total *effort*.
- 3) *Quality dan Testing*, aktivitas yang berhubungan dengan kualitas dan pengujian seperti pengujian integrasi, penjaminan kualitas, evaluasi dan pengujian yang mencapai sekitar 37% dari total *effort*. Secara lebih detail pendistribusian persentase *effort* tersebut dapat dilihat pada **Tabel 2.5**.

Tabel 2.5 Distribusi Usaha dari masing-masing Aktivitas

No.	Aktivitas	Persentase (%)
1.	<i>Software Development</i>	
	a. Requirement	7.50%
	b. Spesification & Design	17.50%
	c. Coding	10.00%
	d. Testing	7.00%
	e. Acceptance & Deployment	5.00%
2	<i>Ongoing Activity</i>	
	a. Project management	7.00%
	b. Configuration Management	3.00%
	c. Documentation	3.00%
	d. Training & Support	3.00%
3	<i>Quality and Testing</i>	

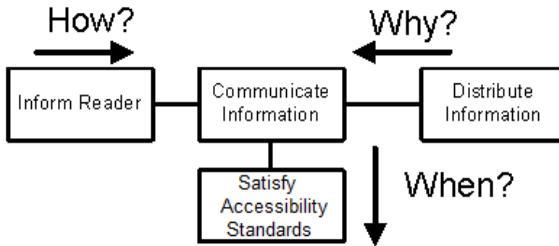
No.	Aktivitas	Persentase (%)
	1. Quality Assurance & Control	12.34%
	2. <i>Evaluation &amp; Testing</i>	24.66%
	<b>Total</b>	<b>100%</b>

Berdasarkan pembagian tersebut, kita dapat mengetahui aktivitas apa saja yang diperlukan dalam pembangunan sebuah aplikasi. Dengan menentukan masing-masing jabatan yang bertanggungjawab pada masing-masing aktivitas dan mengetahui gaji per-bulan dari jabatan tersebut kita dapat mengetahui biaya pembangunan aplikasi dengan mengalikan seluruh data yang dimiliki yakni jumlah SDM (*effort*), jumlah waktu pengerjaan (s) dan gaji bulanan tiap jabatan yang sebelumnya telah diubah kedalam gaji per-jam. Seluruh hasil yang diperoleh dari masing-masing aktivitas akan ditotal untuk dijadikan harga pembangunan aplikasi.

### 2.1.8.3 F.A.S.T (*Function Analysis System Technique*)

#### A. Pengertian FAST Diagram

F.A.S.T. digunakan pada fase ke-dua dalam siklus *Value engineering* yakni "*Function Analysis Phase*" yang dapat menunjukkan objek permasalahan serta mengidentifikasi batasan dari proyek melalui *input*-an yang diterima dari partisipan. Teknik ini disajikan dalam bentuk grafik sistematis yang menunjukkan hubungan antara fungsi sebuah proyek, produk dan proses atau layanan dengan menjawab pertanyaan "Bagaimana fungsi tersebut dapat didapat?", "Mengapa fungsi tersebut digunakan?", "Kapan fungsi tersebut digunakan, dan apa mekanisme penggunaannya (langkah proses, *hardware*, *software*?)".



Gambar 2.9 F.A.S.T Diagram

F.A.S.T memiliki 2 orientasi yakni orientasi teknis dan orientasi pelanggan. Tidak ada benar atau salah dalam penyajian F.A.S.T diagram, yang ada adalah valid atau tidaknya diagram untuk merepresentasikan logika permasalahan. F.A.S.T diagram akan menghasilkan *Output* yang berbeda-beda tergantung pengetahuan dan cakupan yang dimiliki oleh partisipan terkait permasalahan yang sedang dibahas.

### B. Keuntungan Menggunakan FAST

Penggunaan F.A.S.T dapat mendukung komunikasi antar tim [26]. Hal ini akan membantu tim untuk memperoleh beberapa hal berikut ini :

- Mengorganisir daftar random dari fungsi-fungsi yang ada. FAST memperlihatkan hubungan spesifik dari semua fungsi. Hal ini dapat meyakinkan atau menjamin bahwa fungsi-fungsi tersebut memiliki hubungan satu sama lain.
- Mengidentifikasi fungsi yang hilang. Logika How/Why tidak akan mengecek bila fungsi-fungsi itu hilang. FAST mengklarifikasi arti dan deskripsi fungsi kata kerja/benda, dan menyediakan cara-cara yang dapat menggantikannya untuk menjelaskan fungsi-fungsi yang ada.
- Mensimplifikasikan daftar fungsi yang semua dimunculkan.
- Membantu mencari fungsi duplikat atau deksripsi fungsi yang memiliki arti sama. Sehingga dapat dengan mudah

mengeliminir mereka yang tidak perlu atau tidak membantu kemungkinan lain yang sangat perlu.

- Mengidentifikasi fungsi dasar, proses dan produk dari proyek.
- Membantu menyusun cakupan studi.  
Menyusun scop adalah bebas tergantung kesepakatan tim. Fungsi yang berada disebelah kanan garis, diasumsikan sebagai kondisi ayng sudah ada dan bukan untuk dipelajari.
- Memperdalam pemahaman terhadap masalah. FAST dapat memperlihatkan masalah secar objektif yaitu dengan berbasiskan total atau sistem, daam hubungan permasalahan yang lebih kecil lagi dan dalam hal hubungan antara logika masing-masing permasalahan.
- Mendemonstrasikan bahwa sebuah tim analisa telah dibuat.

### **C. Cara Membangun FAST Diagram**

Pembangunan F.A.S.T diagram dimulai dengan menentukan fungsi. Pembangunan ini didasarkan pada pemberian jawaban terhadap beberapa pertanyaan berikut ini [27]:

1. “Bagaimana fungsi tersebut diperoleh?”
2. “Mengapa fungsi tersebut bisa dijalankan?”

Serta menjawab pertanyaan berikut sebagai pertanyaan detail terkait pertanyaan nomor 2.

- a. “Kapan fungsi tersebut digunakan?”

Selanjutnya jawaban dari pertanyaan tersebut akan diidentifikasi dengan cara sebagai berikut :

1. Memperluas fungsi berdasarkan fungsi “Bagaimana” dan “Mengapa”.
2. Bangun dasar pertanyaan “Bagaimana” (panah dari kiri ke kanan) dengan bertanya “Bagaimana fungsi tersebut dapat dicapai?”. Letakkan jawaban tersebut kedalam kata kerja aktif dan kata benda yang terukur.

3. Lakukan percobaan dengan pertanyaan “Mengapa” (panah dari kanan ke kiri) dengan bertanya “Mengapa fungsi ini dilakukan?”
4. Ketika logikanya tidak sesuai, lakukan identifikasi terhadap fungsi yang hilang atau mengalami redundansi atau sesuaikan dengan permintaan.
5. Untuk mengidentifikasi fungsi lain secara bersamaan, tanyakan “kapan fungsi ini selesai dan apa yang menjadi keluaran dari pengimplementasian fungsi ini?”
6. Fungsi permintaan yang lebih tinggi (bagian kiri diagram) menggambarkan apa yang sedang dilakukan dan fungsi yang lebih rendah (bagian kanan diagram) menggambarkan bagaimana pencapaiannya.
7. “Kapan” tidak mengacu kepada waktu, tapi mengacu kepada fungsi yang terjadi secara bersamaan dengan atau sebagai hasil relasi satu dan yang lain.

Pembangunan FAST diagram terbagi atas 2 kategori yakni *Technical oriented* (**Gambar 2.9**) dan *Task Oriented* (**Gambar 2.10**). Dalam FAST Diagram terdapat 2 klasifikasi yang berbeda untuk mendefinisikan fungsi, yakni :

1. *Basic Function*

Mendesripsikan karakteristik atau tugas yang menjadi alasan pertama untuk keberadaan item tersebut, atau dengan kata lain, ini merupakan alasan mengapa produk atau proses tersebut didesain.

2. *Secondary Function*

Adalah mereka yang fungsinya didesain disebabkan oleh adanya *basic Function*. Ini adalah fungsi yang secara langsung berpengaruh bagi ketercapaian *basic Function*. Secara lebih lanjut dapat dibagi kedalam beberapa kategori yang diketanal sebagai *sub-groups*.

- *Dependent Function*

Sebuah fungsi yang keberadaannya bergantung pada fungsi lain. Menjadi nyata ketika sebuah metode spesifik telah dipilih.

- *Independent Function*

Keberandaannya tidak tergantung kepada fungsi lain atau metode yang dipilih untuk membangun fungsi ini.

- *Support Function*

Sebuah fungsi yang membantu *critical Function* dalam melakukan tugasnya sehingga dapat dilakukan dengan cara yang dapat diandalkan dan diterima.

Keterangan lain terkait pada pembangunan FAST Diagram adalah sebagai berikut [12]:

1. *Critical Path Function*

Merupakan fungsi yang secara spesifik mendeskripsikan bagaimana atau mengapa fungsi lain dibutuhkan

2. *Higher Order Function*

Fungsi ini adalah alasan untuk keberadaan *lower order Function*. Pada FAST Diagram mereka diletakkan pada bagian paling kiri dari diagram.

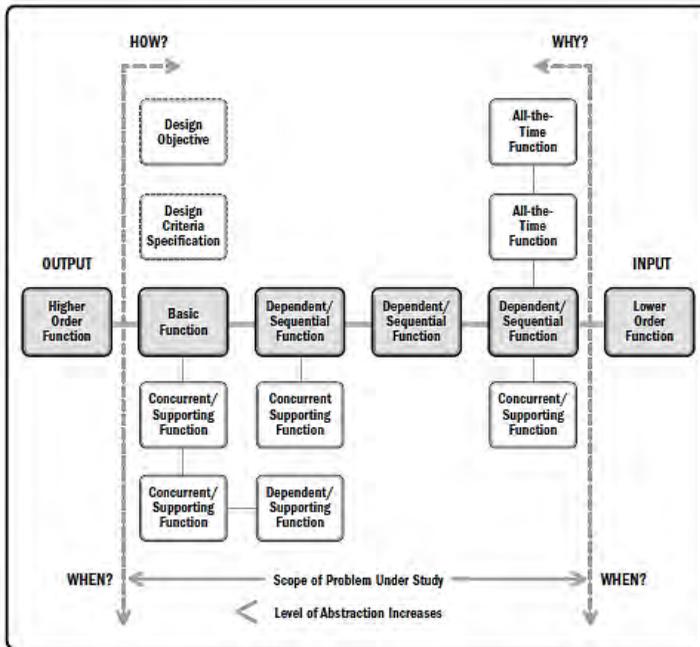
3. *Lower Order Function*

Berfungsi untuk melayani *higher order Function*. Keberadaannya didasarkan pada relevansinya dengan *higher order Function*. Pada FAST Diagram mereka diletakkan pada bagian paling kanan dari diagram.

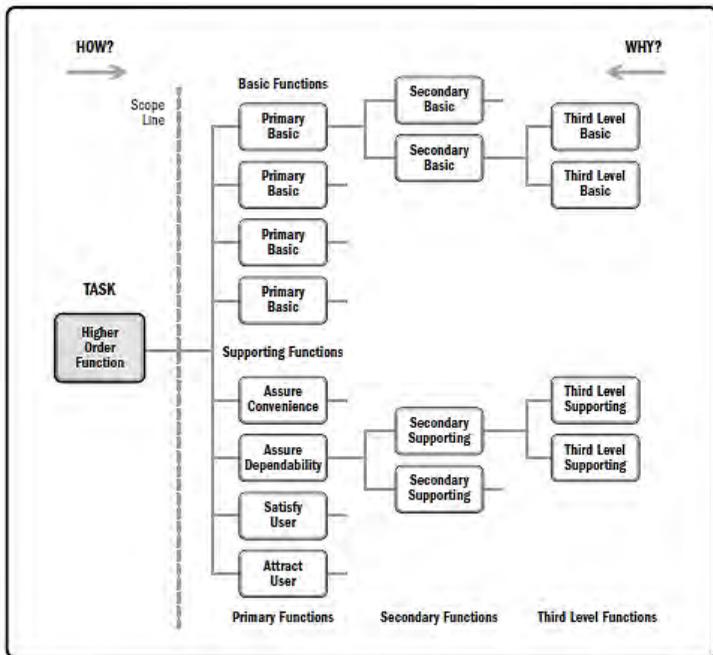
Pada FAST Diagram terdapat beberapa *note* yang perlu dipahami :

Tidak ada kata benar pada FAST Diagram untuk sebuah produk, proses, *system* atau layanan. Karena mereka didasarkan pada :

- Fokus analisis
- Teknologi atau fokus pelanggan
- Objektif yang sedang dipelajari untuk pengimplementasiannya – penghematan biaya, peningkatan proses atau penyelesaian masalah teknik.



Gambar 2.10 *Technical Oriented FAST*



Gambar 2.11 *Customer* (Task) Oriented FAST

## D. Cara Membangun FAST Diagram

### Contoh 1.

Berdasarkan studi kasus proyek konstruksi yang dikerjakan oleh mahasiswa teknik sipi Yogyakarta, salah satu masalah yang ingin mereka analisis adalah item pekerjaan pemasangan pintu lapis teakwood [28]. Analisis ini dilakukan dengan berorientasi pada teknik sehingga mengikuti template *Technical-Oriented FAST* (**Gambar 2.14**).

Tujuan awal pemasangan pintu lapis teakwood ini adalah agar ruangan yang menggunakan pintu ini dapat terjaga privasinya. Selanjutnya dilakukan analisis lebih mendetail dengan menjawab pertanyaan dasar sebagai berikut sebagai dasar pembangunan FAST Diagram:

1. “Bagaimana fungsi tersebut diperoleh?”

Penjagaan privasi pada bangunan tersebut diperoleh dengan cara mengontrol akses bau, akses audio, akses visual, dan akses fisik.

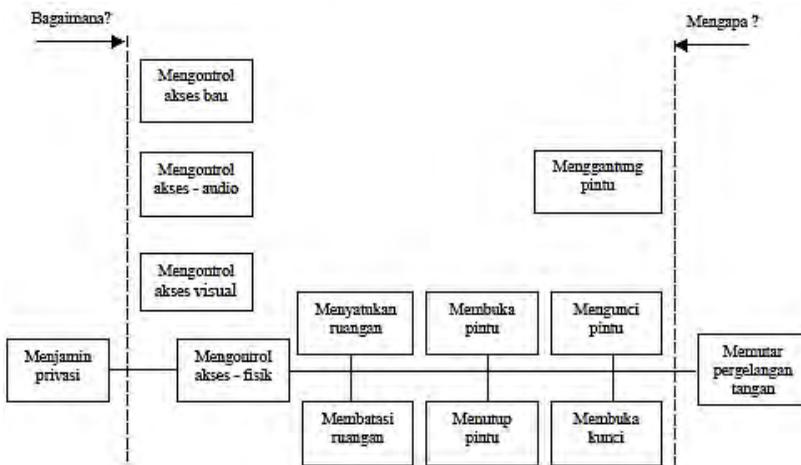
2. “Mengapa fungsi tersebut bisa dijalankan?”

Penjagaan privasi bisa dijalankan karena adanya mekanisme, yakni menggantungkan pintu dan memutar pergelangan tangan untuk menjalankan prosedur terkait pintu tersebut.

Untuk mendetailkan ketercapaian fungsi, diperlukan jawaban atas pertanyaan berikut:

- a. “Kapan fungsi tersebut digunakan?”

Fungsi memutar pergelangan tangan dijalankan ketikan fungsi lain pintu dijalankan yakni menyatukan dua ruangan yang terpisah oleh dinding atau tembok dan untuk membatasi ruangan yang pada awalnya adalah satu, namun ingin dibagi menjadi beberapa ruangan yang lebih kecil. Selain itu pintu teakwood ini juga digunakan jika kita menginginkan adanya komponen membuka, menutup, mengunci dan membuka kunci pada pintu ruangan bangunan. Secara lebih jelas dapat dilihat pada FAST Diagram (**Gambar 2.11**).



Gambar 2.12 Contoh FAST Diagram pada Pintu Teakwood

### Contoh 2.

Untuk semakin memperjelas, contoh berikutnya adalah pada penggunaan PC Proyektor. Tujuan awal penggunaan PC Proyektor adalah untuk membagikan informasi [12]. Selanjutnya dilakukan analisis lebih mendetail dengan menjawab pertanyaan dasar sebagai berikut sebagai dasar pembangunan FAST Diagram:

1. “Bagaimana fungsi tersebut diperoleh?”

Pembagian informasi dapat dilakukan jika proyek mampu menampilkan gambar dan memfokuskan gambar serta memperoleh listrik dan sinyal.

2. “Mengapa fungsi tersebut bisa dijalankan?”

Fungsi ini dapat dijalankan karena dua hal ini berikut ini:

1. Proyektor menerima energi listrik dan mengubah energi listrik tersebut sehingga menghasilkan cahaya yang akan diteruskan untuk membuat gambar menjadi fokus.

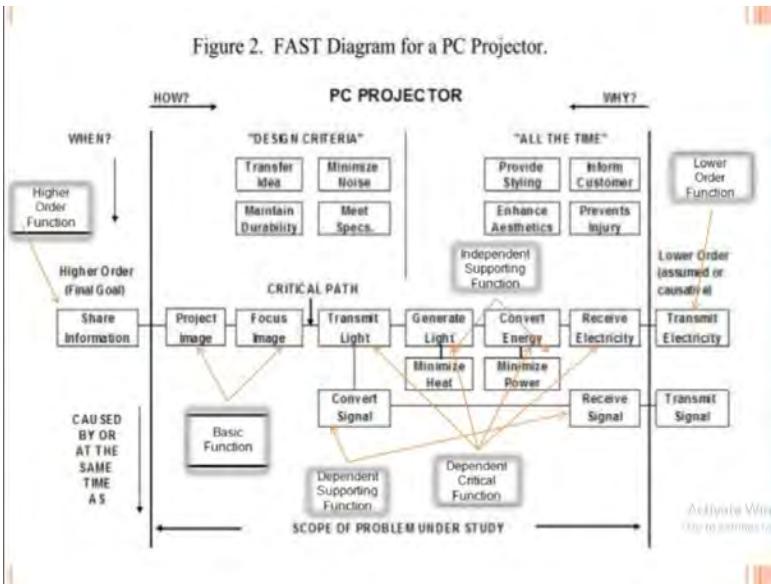
2. Proyektor menerima penerusan signal dan mengkonversinnnya untuk menghasilkan cahaya yang dapat membuat gambar menjadi focus.

Untuk mendetailkan ketercapaian fungsi, diperlukan jawaban atas pertanyaan berikut:

- a. “Kapan fungsi tersebut digunakan?”

Fungsi pengubahan energi listrik dapat digunakan ketika tenaga dalam keadaan *minimize*, dan energi cahaya dapat dihasilkan ketika *heat minimal*. Selain itu, desain kriteria seperti *transfer idea*, *minimize noise*, waktu *maintenance*, dan spesifikasi yang dibutuhkan sesuai. Fungsi ini dapat dilakukan ketika kita ingin memberikan informasi kepada *customer*, memberi pembuktian tentang suatu desain, dan lain-lain.

Secara lebih jelas dapat dilihat pada FAST Diagram (Gambar 2.12).



Gambar 2.13 Contoh FAST Diagram pada PC Proyektor

#### 2.1.8.4 Pencarian Indeks Nilai

Indeks nilai adalah hubungan antara kelayakan dengan biaya. Artinya bahwa biaya yang tepat dapat menjadi salah satu ukuran apakah produk yang dibeli atau disewa memiliki kualitas yang sesuai dengan harganya. Saat pengembangan produk baru atau produk yang sudah ada namun kurang memuaskan, diperlukan analisis minimum pengeluaran sumber daya yang dibutuhkan sampai pada tingkat yang dapat diterima oleh tim atau pihak terkait. Hal ini menyebabkan perubahan di beberapa bagian produk termasuk perubahan biaya. Perubahan biaya tersebut perlu dihitung dan disesuaikan dengan ketentuan yang diberlakukan untuk mengetahui apakah perubahan tersebut perlu dipertimbangkan atau tidak [13].

Sebelum menentukan dan memberikan rekomendasi pemilihan skenario yang optimal, diperlukan perhitungan indeks nilai.

$$\text{Nt/Np} = \frac{\text{Biaya awal} - \text{Biaya akhir}}{\text{Total Anggaran}} \times 100\%$$

Perhitungan skenario dengan membandingkan nilai tukar (Nt) atau nilai perubahan yang diperoleh setelah skenario dijalankan dengan total nilai awal (Np). Berdasarkan rumus tersebut terdapat beberapa ketentuan yang perlu diperhatikan [15] :

1.  $\text{Nt/Np} < 1\%$ , maka *Value engineering* tidak layak dilakukan karena akan menyebabkan kerugian
2.  $\text{Nt/Np} = 1\%$ , maka *Value engineering* tidak layak dipertimbangkan karena upaya akan menghasilkan break even
3.  $\text{Nt/Np} > 1\%$ , maka *Value engineering* layak dipertimbangkan untuk dilakukan.

#### 2.1.8.5 Teknik Brainstorming Osborn

Brainstorming adalah proses kelompok yang dikembangkan pada tahun tujuh puluhan oleh Alex Osborn pada tahun 1971. Teknik ini dimulai dari adanya permasalahan yang memerlukan

identifikasi. Identifikasi tersebut dilakukan secara berkelompok dalam sebuah pertemuan dan memberikan kesempatan bagi seluruh anggota untuk menyampaikan pendapat masing-masing terkait permasalahan, hingga ditemukan solusi yang kreatif [29].

### **2.1.9 Hubungan Proyek, Kontrak dan *Value engineering***

Sebuah proyek akan dijalankan saat kontrak telah disetujui oleh pihak klien dan perusahaan pemegang proyek. Tidak jarang pengerjaan proyek menggunakan teknik skenario yang artinya memperlakukan satu hal yang sama dengan berbagai usaha yang berbeda. Namun hasil yang diperoleh tidaklah selalu baik. Hal ini memberikan resiko kurangnya waktu maupun biaya yang telah disetujui pada kontrak [29].

*Value engineering* menjadi solusi terbaik yang dapat dipilih dalam penyelesaian permasalahan ini. Metode ini dapat memberikan perubahan baik dari segi proses, struktur, aliran informasi, biaya pengadaan dan lain-lain dari kontrak yang telah disetujui terkait proyek yang sedang dijalankan. Hasil akhir penerapan metode ini berupa dokumen usulan perubahan nilai atau *Value Engineering Change Proposals* (VECP) yang selanjutnya akan diserahkan kepada pihak yang berkepentingan untuk memperoleh persetujuan [29]. Secara spesifik, terdapat keterkaitan antara tantangan yang dihadapi dalam pengerjaan proyek dengan siklus pada *Value engineering* (**Gambar 2.15**).

<b>Project Management Challenge</b>	<b>Value Engineering Solution</b>
Making the transition from practitioner to business partner	Pre-Workshop activities
Ability to properly define expectations (Vision and Scope)	Function analysis techniques
Determining value of a project for prioritization and ranking against other projects	The Value Study cycle
Maximizing Return on Investment for the Project Sponsors	Development and Presentation phases of the Workshop
Maximizing Resources to accomplish the project objectives	Post-workshop activities
Minimizing Cost while maximizing value	The Value study cycle
Minimizing Risk	Pre-Workshop activities
Ability to adapt and innovate	The Creative phase

Gambar 2.14 Hubungan Tantangan Proyek dengan *Value engineering*

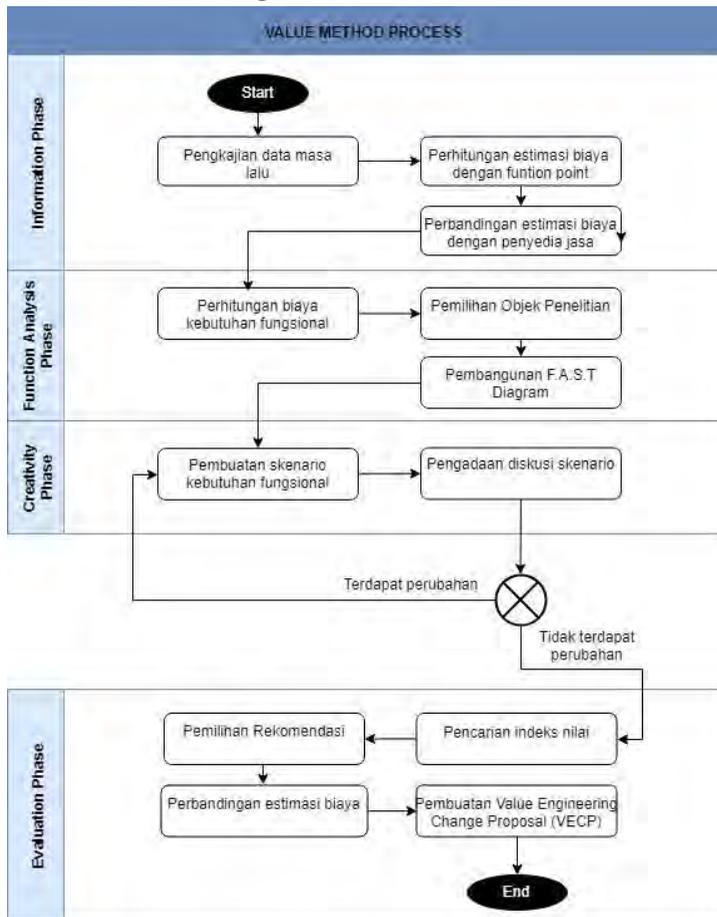
*Halaman ini sengaja dikosongkan*

## BAB III METODOLOGI

Pada bab ini akan menjelaskan tentang metode yang digunakan dalam menyelesaikan Tugas Akhir ini.

### 3.1 Langkah-Langkah Penelitian

#### 3.1.1 Flowchart Langkah Penelitian



Gambar 3.1 Flowchart Langkah-Langkah Penelitian

### 3.1.2 *Input, Proses, Output* Langkah-Langkah Penelitian

Langkah penelitian ini dilakukan secara terstruktur dan sistematis sebagai berikut :

Tabel 3.1 *Input, Proses dan Output* penelitian

Langkah-Langkah Penelitian			
<i>Input</i>		<i>Proses</i>	<i>Output</i>
<b>Pre-Analysis</b>	<i>Orientation</i>	<ul style="list-style-type: none"> <li>• Membaca dan memahami paper dan buku terkait <i>Value engineering</i></li> <li>• Membaca dan memahami KAK</li> <li>• Melakukan identifikasi permasalahan yang terjadi.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>BAB II</b> Dasar Teori</li> <li>• <b>BAB III</b> Metodologi Penelitian</li> <li>• <b>BAB I</b> Latar Belakang permasalahan</li> <li>• Data masa lalu</li> </ul>

## Langkah-Langkah Penelitian

		<i>Input</i>	<i>Proses</i>	<i>Output</i>
<i>Value Analysis</i>	<i>Information Phase</i>	<ul style="list-style-type: none"> <li>• <b>BAB II</b> Dasar Teori</li> <li>• <b>BAB III</b> Metodologi Penelitian</li> <li>• <b>BAB I</b> Latar Belakang permasalahan dan tujuan yang ingin dicapai.</li> <li>• Data masa lalu</li> <li>• Standar yang digunakan perusahaan atau klien</li> <li>• Peraturan dan kebijakan yang berlaku didalam lingkungan perusahaan atau klien</li> <li>• Informasi anggaran total proyek</li> </ul>	<ul style="list-style-type: none"> <li>• Mengkaji data masa lalu</li> <li>• Melakukan perhitungan estimasi biaya pembangunan aplikasi dengan <i>function point</i></li> <li>• Melakukan perbandingan estimasi biaya dengan penyedia jasa</li> </ul>	<ul style="list-style-type: none"> <li>• <b>BAB I</b> Ketentuan objek yang akan diteliti, sasaran yang akan dicapai dan ruang lingkup atau batasan penelitian.</li> <li>• Usulan kebutuhan fungsional pihak ketiga</li> <li>• Estimasi biaya pembangunan aplikasi</li> <li>• Perbandingan Estimasi harga dengan penyedia jasa lain</li> </ul>

Langkah-Langkah Penelitian				
<i>Input</i>		<i>Proses</i>	<i>Output</i>	
	<i>Function Analysis Phase</i>	<ul style="list-style-type: none"> <li>• <b>BAB I</b> Ketentuan objek yang akan diteliti, sasaran yang akan dicapai dan ruang lingkup atau batasan penelitian.</li> <li>• Usulan kebutuhan fungsional pihak ketiga</li> <li>• Estimasi biaya pembangunan aplikasi</li> <li>• Perbandingan Estimasi harga dengan penyedia jasa lain</li> </ul>	<ul style="list-style-type: none"> <li>• Melakukan perhitungan biaya masing-masing kebutuhan fungsional</li> <li>• Melakukan pemilihan objek penelitian</li> <li>• Pembangunan FAST Diagram</li> </ul>	<ul style="list-style-type: none"> <li>• Estimasi biaya masing-masing kebutuhan fungsional</li> <li>• Perhitungan bobot masing-masing kebutuhan fungsional</li> <li>• Penentuan objek penelitian</li> <li>• Diagram F.A.S.T objek penelitian</li> </ul>

### Langkah-Langkah Penelitian

		<i>Input</i>	<i>Proses</i>	<i>Output</i>
<i>Creativity Phase</i>		<ul style="list-style-type: none"> <li>• Estimasi biaya masing-masing kebutuhan fungsional</li> <li>• Perhitungan bobot masing-masing kebutuhan fungsional</li> <li>• Penentuan objek penelitian</li> <li>• Diagram F.A.S.T objek penelitian</li> </ul>	<ul style="list-style-type: none"> <li>• Membuat skenario perubahan fungsi-fungsi aplikasi.</li> <li>• Melakukan brainstorming dengan pihak ketiga</li> </ul>	<ul style="list-style-type: none"> <li>• Daftar berbagai hasil skenario yang berhasil dibuat oleh penulis</li> <li>• Daftar tambahan skenario oleh pihak ketiga (Jika ada)</li> <li>• Catatan hasil brainstorming terkait skenario</li> </ul>

Langkah-Langkah Penelitian				
<i>Input</i>		<i>Proses</i>	<i>Output</i>	
	<i>Evaluation Phase</i>	<ul style="list-style-type: none"> <li>• Daftar berbagai hasil skenario yang berhasil dibuat oleh penulis</li> <li>• Daftar tambahan skenario oleh pihak ketiga (Jika ada)</li> <li>• Catatan hasil brainstorming terkait skenario</li> </ul>	<ul style="list-style-type: none"> <li>• Pembuatan diagram F.A.S.T (Jika diperlukasn)</li> <li>• Melakukan pencarian indeks nilai</li> <li>• Melakukan pemilihan rekomendasi atau alternatif solusi yang akan diimplementasikan berdasarkan hasil analisis bersama pihak ketiga</li> <li>• Melakukan penyusunan VECP</li> </ul>	<ul style="list-style-type: none"> <li>• Buku Tugas Akhir</li> </ul>

### 3.1.2.1 Pre-Analysis

Pada tahap ini akan dijelaskan mengenai tahapan yang dilakukan penulis sebelum memulai level *Value-Analysis*.

#### 0. *Orientation*

Pada tahap ini dilakukan pengumpulan sumber informasi atau pengetahuan tentang *Value engineering*, serta informasi pihak yang akan terlibat dalam proyek yang akan diadakan, baik suplier, proyek manajer dan tim analisa. Penulis akan mengeksplere internet, beberapa situs kumpulan paper terpercaya seperti sciencedirect.com, libgen.io, dan lain-lain, mencari paper dengan keyword *Value engineering*, membaca, memahami dan melakukan review informasi sehingga menjadi sebuah pengetahuan tentang *Value engineering* bagi penulis. Kemudian review tersebut akan penulis tuliskan lagi kedalam bahasa yang lebih formal untuk dijadikan sebagai Dasar Teori pada BAB II. Selanjutnya penulis akan melakukan pertemuan dengan pihak terkait untuk melakukan wawancara atau menggali informasi permasalahan apa yang terjadi saat ini dan bagaimana hal tersebut bisa terjadi untuk dijadikan sebagai latar belakang pada BAB I. Selanjutnya penulis akan meminta data yang telah disetujui, untuk dibaca, dipahami dan dijadikan sebagai data masa lalu yang akan menjadi salah satu bahan pertimbangan pada pelaksanaan tahap selanjutnya. Setelah memperoleh pengetahuan, data dan informasi, penulis menyusun metode penelitian tentang *Value engineering* terkait permasalahan yang sedang dihadapi oleh pihak terkait dan menjadi isi dari BAB III.

### 3.1.2.2 Value-Analysis

Tahapan ini akan menjelaskan mengenai tahapan utama pada Tugas Akhir ini.

#### 1. *Information Phase*

Pada tahap ini, penulis ingin mengetahui apakah biaya yang ditawarkan oleh pihak ketiga terbilang mahal atau tidak.

Pertama penulis akan mengkaji data masa lalu yang penulis peroleh dari pihak ketiga. Untuk memperoleh seluruh data yang diperlukan dalam pengerjaan Tugas Akhir ini. Adapun data yang ingin diperoleh adalah :

1. Waktu pembangunan aplikasi
2. Jumlah SDM yang membangun aplikasi
3. Biaya pembangunan keseluruhan aplikasi
4. Harga aplikasi per-fitur
5. Perbandingan aplikasi dengan aplikasi lain
6. Detail tugas masing-masing SDM dalam pengerjaan aplikasi

Jika seluruh kebutuhan lengkap maka penulis dapat langsung melanjut pada tahapan berikutnya, namun jika kebutuhan tidak lengkap, maka terdapat beberapa teknik yang dapat diterapkan untuk melengkapi, diantaranya adalah :

1. *Function point*. Merupakan teknik yang dapat dimanfaatkan untuk perhitungan biaya pembangunan keseluruhan aplikasi dan harga aplikasi per-fitur
2. *Benchmarking*. Merupakan teknik yang dapat digunakan untuk melakukan perbandingan dengan pihak lain untuk menentukan apakah harga usulan pihak ketiga terbilang mahal atau tidak.

## **2. *Function Analysis Phase***

Pada tahap ini penulis ingin mengetahui objek atau kebutuhan fungsional apa saja yang akan menjadi objek kajian pada tahapan-tahapan selanjutnya. Untuk mengetahui hal tersebut, pertama penulis akan melakukan perhitungan biaya pada masing-masing kebutuhan fungsional berdasarkan usulan pihak ketiga. Perhitungan ini menggunakan teknik *Function point* sama seperti sebelumnya. Selanjutnya penulis akan melakukan perhitungan bobot masing-masing kebutuhan fungsional tersebut untuk melihat fungsi apa saja yang membutuhkan biaya yang besar. Namun dalam menentukan

pilihan, penulis membutuhkan dasar atau syarat. Adapun objek yang penulis pilih adalah kebutuhan fungsional dengan tipe *complex* dan *highly complex*. Setelah objek dipilih, sebagai tahap terakhir fase ini, penulis membangun F.A.S.T Diagram pada masing-masing objek yang terpilih untuk melihat keterkaitan komponen dalam sistem antara satu dan yang lain, untuk membuat fungsi tersebut dapat berjalan.

### **3. *Creativity Phase***

Pada fase ini penulis akan membuat skenario-skenario terhadap objek yang terpilih pada tahapan *Function Analysis Phase*. Adapun ketentuan pembuatan skenario adalah hal yang dapat menghemat biaya atau mengurangi harga pembangunan aplikasi namun tidak mengubah fungsi dan keterkaitan komponen sistem yang membuat fungsi tersebut berjalan. Skenario yang penulis dapatkan tidak selalu berhasil menghemat biaya namun ada kemungkinan skenario yang didapat justru menghasilkan biaya yang lebih besar daripada estimasi sebelumnya. Pembuatan skenario tidak memiliki batasan jumlah.

Pada tahapan ini, penulis juga melibatkan aktivitas brainstorming dengan pihak ketiga untuk memastikan bahwa skenario yang penulis usulkan diterima oleh pihak ketiga. Tahapan ini juga berguna untuk menampung usulan dari pihak ketiga terkait skenario pengurangan biaya pembangunan aplikasi.

Jika ternyata terdapat usulan baru maupun perubahan dari pihak ketiga, maka tahapan ini akan berulang dari mulai pembuatan skenario hingga melakukan brainstorming dengan pihak ketiga hingga terdapat kesepakatan antara penulis dengan pihak ketiga untuk melanjutkan proses ke tahapan berikutnya.

#### 4. *Evaluation Phase*

Pada fase ini penulis akan merangkum skenario dengan membandingkan biaya sebelum dan sesudah dilakukan perhitungan biaya berdasarkan skenario. Tahapan ini akan melibatkan teknik pencarian indeks nilai. Teknik ini berfungsi untuk menilai skenario. Penilaian tersebut akan menghasilkan keterangan atau status skenario, apakah layak dipertimbangkan, tidak layak atau hasil yang diperoleh sama seperti usulan. Selanjutnya dengan mengikuti prinsip hubungan waktu dan biaya yang optimal pada proyek, penulis akan memilih skenario dengan waktu paling cepat dan biaya paling murah untuk dijadikan sebagai rekomendasi yang paling optimal bagi pengerjaan proyek ini.

Hasil analisis tersebut akan didokumentasikan sebagai *Value Engineering Change Proposals* (VECP) sehingga dapat dijadikan pedoman pemegang proyek dalam mengimplementasikan proyeknya sesuai dengan budget dan timeline pengerjaan proyek. Pada tahap ini penulis dapat memberikan masukan kepada pihak terkait serta mendokumentasikan hasil perencanaan yang dilakukan kedalam buku tugas akhir.

#### 3.2 **Timeline Pengerjaan Tugas Akhir**

Pada bagian ini akan disajikan jadwal yang akan digunakan oleh penulis dalam mengerjakan tugas akhir ini. Jadwal akan disusun berdasarkan aktivitas pada tahap *Value Analysis* pada metode penelitian, karena tahapan *post Analysis* telah dilewati. Jadwal disusun dalam satuan minggu selama 12 minggu atau 3 bulan. Jadwal kegiatan dapat dilihat pada tabel berikut.

Tabel 3.2 Tabel Timeline Pengerjaan Tugas Akhir

No	Aktivitas	Minggu											
		1	2	3	4	5	6	7	8	9	10	11	12
<b>Value Analysis</b>													
<b>Information Phase</b>													
1	Pengkajian data masa lalu												
2	Penghitungan estimasi biaya dengan <i>function point</i>												
3	Perbandingan estimasi biaya dengan penyedia jasa												
<b>Function Analysis Phase</b>													
4	Perhitungan biaya kebutuhan fungsional												
5	Pemilihan objek penelitian												
6	Pembangunan F.A.S.T Diagram												
<b>Creativity Phase</b>													
7	Pembuatan skenario kebutuhan fungsional												
8	Pengadaan diskusi skenario												
<b>Evaluation Phase</b>													
9	Pencarian indeks nilai												
10	Pemilihan rekomendasi												
11	Perbandingan estimasi biaya												
12	Pembuatan VECF												

*Halaman ini sengaja dikosongkan*

## BAB IV PERANCANGAN

Pada bab ini akan dijelaskan mengenai rancangan penelitian. Tugas Akhir. Perancangan ini diperlukan sebagai panduan dalam melakukan penelitian Tugas Akhir hingga menghasilkan VECP (*Value Engineering Change Proposal*).

### 4.1 Penjelasan Studi Kasus

Aplikasi XYZ merupakan aplikasi *web-base* yang dapat digunakan pada piranti android dan IOS yang dibangun dalam mendukung keberlangsungan salah satu *Event* Olahraga terbesar 2018. Aplikasi ini ditujukan untuk meng-*organising* sebagai alternatif pendamping berupa layanan akses otomatis bagi para Atlet, *Official* dan *Press Event* Olahraga 2018 tersebut dari mulai saat mereka datang, tinggal atau menetap, saat melaksanakan pertandingan hingga kepulangan ke negara masing-masing. Aplikasi ini direncanakan akan dibangun dalam kurun waktu 5 bulan terhitung sejak Februari hingga Juli 2018. Fungsi utama dari aplikasi ini adalah memberikan kemudahan bagi pengguna dalam berbagai hal terlebih terkait dengan keuangan. Aplikasi ini akan dibangun berdasarkan fungsi utama sebagai *payment gateway* yang akan dapat menyimpan uang pengguna menjadi “jembatan” yang siap digunakan dalam segala transaksi keuangan selama menjadi peseta, sehingga akan mempermudah setiap transaksi kapanpun dan dimanapun seperti transportasi online, pembelian tiket pertandingan, hotel, dan lain-lain. Terdapat beberapa fungsi yang berhasil didefinisikan oleh pihak ketiga dalam tujuannya membangun aplikasi *payment gateway* ini (**Tabel 4.1**).

Tabel 4.1 Kebutuhan Fungsional Usulan Pihak Ketiga

No	Kebutuhan Fungsional
1.	Login
2.	Aplikasi dapat menampilkannya informasi saldo
3.	Aplikasi dapat menampilkan aktivitas transaksi yang telah dilakukan <i>customer</i>

No	Kebutuhan Fungsional
4.	Aplikasi dapat melakukan pemesanan transportasi online, tiket venue tertentu dan transaksi lain seperti hotel, pembelian barang, dan lain-lain
5.	Aplikasi dapat menampilkan informasi pendamping dan penjemput atlet
6.	Aplikasi dapat menampilkan informasi hotel yang berada disekitar lokasi <i>Event</i>
7.	Aplikasi dapat menampilkan informasi <i>mall</i> di sekitar lokasi pelaksanaan <i>Event</i> yang menjadi <i>official partner</i> dan diskon produk tertentu yang ada selama <i>Event</i> berlangsung
8.	Aplikasi dapat menampilkan informasi jadwal dan lokasi pertandingan
9.	Aplikasi dapat dimanfaatkan untuk menyampaikan keluhan atau menanyakan hal terkait layanan yang tersedia pada aplikasi
10.	Aplikasi dapat memberikan petunjuk lokasi suatu
11.	Aplikasi dapat ditampilkan dalam berbagai bahasa, minima dua bahasa yakni Bahasa Inggris dan Bahasa Indonesia

Berdasarkan beberapa kebutuhan tersebut, pihak ketiga merumuskan beberapa fitur yang dapat memenuhinya, diantaranya: Fitur fintech ini akan didukung dengan fasilitas “rekening koran” yang dapat memberikan informasi saldo dan rekaman transaksi yang dilakukan. Adapun fitur lain yang diusulkan adalah sebagai berikut [7].

- Fitur yang dapat memudahkan Atlet dalam mendapatkan informasi akomodasi seperti informasi pendamping atau penjempu, hotel, diskon *mall* untuk atlet dengan *Merk* tertentu (*Official Brand Partner*), dan lain-lain
- Fitur yang dapat memudahkan Atlet dalam mendapatkan informasi pertandingan seperti informasi jadwal pertandingan dan *venue* atlet, informasi jadwal tanding harian, dan lain-lain
- Fitur penghubung atlet dengan *help desk* secara pesan online (*chat*) maupun verbal (*voice*), sehingga jika ada

keluhan, bantuan dan lain-lain dapat segera didapatkan atau diberikan solusi.

- Fitur lain yang dapat memberikan kemudahan bagi pengguna, seperti *maps* seperti *waze*, map seperti *google maps* dan kemampuan multi bahasa dari aplikasi.

## 4.2 *Value engineering*

### 4.2.1 *Pre-Analysis*

Tahapan ini bertujuan agar penulis dan pembaca memahami apa itu *Value engineering* dan objek apa yang akan penulis analisa dalam Tugas Akhir ini. Dalam menjalankan tahapan *pre-Analysis* ini penulis tidak membutuhkan persiapan khusus. Persiapan pada tahap ini hanya berupa persiapan penggalian informasi tentang *Value engineering* dan memahaminya untuk pembangunan dasar teori pada bab 2, metode pada bab 3 dan penggalian informasi dari pihak ketiga untuk mengisi kebutuhan bab 1. Pada tahap ini penulis mempersiapkan paketan internet, *device* dan email kampus yang dapat membantu penulis mengakses repositori dari Institut Teknologi Sepuluh Nopember maupun universitas lain.

### 4.2.2 *Analysis*

#### 4.2.2.1 *Information Phase*

Pada tahapan ini, persiapan yang penulis lakukan adalah memahami KAK (Kerangka Acuan Kerja) untuk dicari tahu apakah seluruh kebutuhan pengerjaan tugas akhir telah tersedia dalam KAK atau tidak. Adapun beberapa kebutuhan dari KAK adalah sebagai berikut :

1. Waktu pembangunan aplikasi
2. Jumlah SDM yang membangun aplikasi
3. Biaya pembangunan keseluruhan aplikasi
4. Harga aplikasi per-fitur
5. Perbandingan aplikasi dengan aplikasi lain
6. Detail tugas masing-masing SDM dalam pengerjaan aplikasi

Jika terdapat ketidaklengkapan pada kebutuhan tersebut, terdapat beberapa teknik pengolahan informasi yang penulis dapat gunakan untuk melengkapi informasi tersebut.

### ➤ **Perhitungan Estimasi Biaya dengan *Function point***

Teknik ini berguna dalam melengkapi informasi terkait waktu, jumlah SDM yang optimal dalam pembangunan aplikasi, biaya keseluruhan aplikasi berdasarkan *ManDays*, dan harga pembangunan aplikasi per-fungsi. Adapun langkah yang dilakukan dalam implementasi teknik ini adalah sebagai berikut.

#### **Langkah 1. Menghitung CFP (*Crude Function point*)**

Berdasarkan sumber yang penulis peroleh, dalam mengimplementasikan perhitungan ini penulis harus melakukan penentuan bobot kompleksitas didasarkan pada jumlah tipe elemen data tiap fungsi dan jumlah referensi tipe file terlebih dahulu. Sehingga penulis mencari tahu penentuan bobot tersebut. Adapun penentu bobot kompleksitas dan definisinya adalah sebagai berikut [25]:

- *Data Element Type* merupakan *EIF* yang tidak berulang dan diidentifikasi *user* sebagai *EIF* yang unik.
- *Record element type* merupakan *eksternal interface file* yang berada dalam ILF atau EIF.
- *File type reference* merupakan *Eksternal Interface File* yang dibaca oleh fungsi transaksional. Fungsi ini mempresentasikan fungsionalitas yang disediakan untuk user dalam melakukan pemrosesan terhadap data menggunakan *website*.

Cara paling mudah untuk mengidentifikasi jumlah *file type reference* adalah dengan mengidentifikasi *internal logical files*, jumlah kedua hal ini selalu sama. Kriterianya mengikuti (**Tabel 4.2**) [25].

Tabel 4.2 Perhitungan Bobot Kompleksitas

<b><i>Internal Logical Files dan External Interface Files</i></b>			
		<b>Elemen Data</b>	
<b><i>Record Element Type</i></b>	1-19	20-50	51+
1	<i>Low</i>	<i>Low</i>	<i>Avg</i>
2-5	<i>Low</i>	<i>Avg</i>	<i>High</i>
6+	<i>Avg</i>	<i>High</i>	<i>High</i>
<b><i>External Output dan External Inquiry</i></b>			
		<b>Elemen Data</b>	
<b><i>File Type Reference</i></b>	1-5	6-19	20+
0 atau 1	<i>Low</i>	<i>Low</i>	<i>Avg</i>
2-3	<i>Low</i>	<i>Avg</i>	<i>High</i>
4+	<i>Avg</i>	<i>High</i>	<i>High</i>
<b><i>Internal Input</i></b>			
		<b>Elemen Data</b>	
0 atau 1	<i>Low</i>	<i>Low</i>	<i>Avg</i>
2-3	<i>Low</i>	<i>Avg</i>	<i>High</i>
3+	<i>Avg</i>	<i>High</i>	<i>High</i>

Berdasarkan kategori tersebut, penulis dapat menyesuaikan aktivitas aplikasi dengan kriteria yang ada untuk memperoleh kategori masing-masing aktivitas. Dalam melaksanakan langkah ini, persiapan yang penulis lakukan pertama adalah dengan melakukan perancangan tabel template hasil pengelompokan *inputan* (**Tabel 4.3**).

Tabel 4.3 Template Hasil Pengelompokan *Inputan*

<b>Hasil Pengelompokan <i>Inputan</i></b>	
<b>Nama</b>	<b>Jenis</b>
<b>Eksternal <i>Input</i></b>	
<b>Eksternal <i>Output</i></b>	
<b>Eksternal inquiry</b>	
<b>File Logic</b>	
<b>Interface Eksternal</b>	

Pengelompokan tabel akan dibagi kedalam 3 kategori yakni Simple (S), Average (A) dan Complex (C) dengan kriteria yang telah dijelaskan pada bab IV.

Keterangan :

Nama : Diisi nama aktivitas *inputan* pada masing-masing kriteria. Contoh : Pada bagian *Eksternal Input* diisikan Informasi *mall*, informasi jadwal pertandingan, dan lain-lain

Jenis : Diisi dengan inisial jenis *inputan*, dengan ketentuan inisial : Simple → S, Average → A, dan Complec → C.

Selanjutnya hasil dari pengelompokan tersebut akan diolah kembali. Pada tahap ini penulis melakukan persiapan dengan membangun tabel template hasil perhitungan CFP (**Tabel 4.4**).

Tabel 4.4 Template Hasil Perhitungan CFP

Tipe Fungsi	Bobot Kompleksitas								
	Simple			Average			Complex		
	Count	Bobot	Point	Count	Bobot	Point	Count	Bobot	Point
External Inputs		3			4			6	
External Output		4			5			7	
Internal Logical File		7			10			15	
External Interface Files		5			7			10	
External Inquiry		3			4			6	
<b>TOTAL</b>									<b>?</b>

Keterangan :

- Count : Diisi dengan jumlah *input* yang sesuai dengan kolom. Contoh : Jika terdapat 5 *inputan* yang pengerjaannya masuk kedalam kategori simple, maka pada kolom count simple isikan angka 5.
- Bobot : Nilai bobot sudah merupakan suatu ketetapan.
- Point : Diisi dengan nilai hasil perkalian count dengan bobot.

Total : Diisi dengan jumlah seluruh point dari seluruh kategori. Sehingga bagian ini hanya diisikan disalah satu kolom saja yang bertanda tanda tanya (?).

## Langkah 2. Menghitung Relative Complexity Adjustment Factor (RCAF)

Pada implementasi ini, pemberian nilai bobot kepentingan dengan besar antara 0 sampai 5 untuk masing-masing factor mengacu pada dokumen *Standar Documentation* untuk *General System Characteristics (GSC)* yang dipublikasikan oleh *Software Matric*. Sehingga diperlukan ukuran gmasing-masing factor yang bobot nilai dijelaskan dalam deskripsi untuk menyamakan standar pemberian bobot nilai dan memudahkan untuk digunakan dalam aplikasi berbasis GUI. Sehingga penulis mencari tahu penentuan bobot karakteristik tersebut sebagai berikut [25].

Tabel 4.5 Standar Documentation GSC Data Communication

Faktor	Data Communication
Nilai	Deskripsi
0	Aplikasi merupakan kumpulan proses yang diproses dalam satu perangkat yang berdiri sendiri
1	Aplikasi merupakan kumpulan proses yang memiliki entri data ATAU proses <i>printing</i> secara <i>remote</i>
2	Aplikasi merupakan kumpulan proses yang memiliki entri data DAN proses <i>printing</i> secara <i>remote</i>
3	Aplikasi termasuk pada pengumpulan data secara <i>online</i> atau TP ( <i>teleprocessing</i> ) tampilan antarmuka ke kumpulan proses atau <i>query</i> sistem

<b>Faktor</b>	<b>Data Communication</b>
<b>Nilai</b>	<b>Deskripsi</b>
4	Aplikasi lebih dari sekedar tampilan antarmuka tapi mendukung hanya satu tipe protocol komunikasi TP
5	Aplikasi lebih dari sekedar tampilan antarmuka dan mendukung lebih dari satu tipe protocol komunikasi TP

Tabel 4.6 Standar Documentation GSC Distributed Processing

<b>Faktor</b>	<b>Distributed Data Processing</b>
<b>Nilai</b>	<b>Deskripsi</b>
0	Aplikasi tidak membantu <i>transfer</i> data atau pengolah fungsi antar komponen
1	Aplikasi mempersiapkan data untuk proses <i>end user</i> pada komponen sistem lain seperti PC speardsheets dan PC DBMS
2	Data disiapkan untuk ditransfer, kemudian di transfer dan diproses pada komponen lain dari sistem (bukan untuk pengolahan <i>end user</i> )
3	Pengolahan dan transfer data terdistribusi secara <i>online</i> dan dalam satu arah saja
4	Pengolahan dan transfer data terdistribusi secara <i>online</i> dan dalam banyak arah
5	Pemrosesan fungsi dilakukan secara dinamik terhadap komponen sistem yang paling sesuai

Tabel 4.7 Standar Documentation GSC Performance

<b>Faktor</b>	<b>Performance</b>
<b>Nilai</b>	<b>Deskripsi</b>

0	Tidak ada persyaratan kinerja khusus yang dinyatakan oleh <i>user</i>
1	Persyaratan kinerja dan desain dinyatakan dan ditinjau tapi tidak ada tindakan khusus yang diperlukan
2	Waktu respon sangat penting selama waktu sibuk. Tidak ada desain khusus untuk penggunaan CPU yang dibutuhkan. Batas waktu proses adalah untuk hari kerja berikutnya
3	Waktu respon sangat penting setiap jam bisnis. Tidak ada desain khusus untuk penggunaan CPU yang dibutuhkan. Memproses tenggang waktu diperlukan dengan sistem yang terhambat
4	Persyaratan kinerja pengguna cukup ketat untuk membutuhkan tugas analisis kinerja di fase desain
5	Alat analisis kinerja digunakan di tahap desain, pengembangan dan / atau implementasi untuk mendapatkan persyaratan kinerja yang dinyatakan oleh pengguna

Tabel 4.8 Standar Documentation GSC Heavily Used Configuration

<b>Faktor</b>	<b>Heavily Used Configuration</b>
<b>Nilai</b>	<b>Deskripsi</b>
0	Tidak ada batasan operasional baik secara eksplisit maupun implisit yang disertakan
1	Pembatasan operasional memang ada, tapi kurang ketat dibanding sebuah aplikasi yang memiliki khas lain. Tidak ada upaya khusus yang diperlukan untuk memenuhi pembatasan.

2	Beberapa pertimbangan keamanan atau waktu disertakan
3	Kebutuhan prosesor khusus untuk bagian tertentu dari suatu aplikasi
4	Pembatasan operasi yang ditetapkan memerlukan batasan khusus aplikasi di pusat prosesor
5	Adanya kendala khusus pada aplikasi yang terdistribusi pada komponen sistem

Tabel 4.9 Standar Documentation GSC Transaction Rate

<b>Faktor</b>	<b>Transaction Rate</b>
<b>Nilai</b>	<b>Deskripsi</b>
0	Tidak ada periode transaksi puncak yang diantisipasi
1	Periode transaksi puncak (misal bulanan, tahunan) diantisipasi
2	Periode transaksi puncak mingguan diantisipasi
3	Periode transaksi puncak harian diantisipasi
4	Transaksi tinggi dinyatakan oleh pengguna dalam aplikasi dan memerlukan tugas analisis kinerja pada tahapan perancangan
5	Transaksi tinggi dinyatakan oleh pengguna dalam aplikasi dan memerlukan tugas analisis kinerja dan memerlukan penggunaan alata analisis kinerja dalam desain tahap pengembangan dan / atau instalasi

Tabel 4.10 Standar Documentation GSC Online Data Entry

<b>Faktor</b>	<b>Online Data Entry</b>
<b>Nilai</b>	<b>Deskripsi</b>

0	Semua transaksi diproses dalam mode batch
1	1% sampai 7% transaksi adalah data entri yang interaktif
2	8% sampai 15% transaksi adalah data entri yang interaktif
3	16% sampai 23% transaksi adalah data entri yang interaktif
4	24% sampai 30% transaksi adalah data entri yang interaktif
5	Lebih dari 30% transaksi adalah data entri yang interaktif

Fungsi online yang diberikan menekankan desain untuk efisiensi pengguna akhir dimana desainnya meliputi :

- Bantuan navigasi (misalnya tombol fungsi, lompatan, menu yang dibuat secara dinamis)
- Menu
- Bantuan dan dokumen online
- Gerakan kursor otomatis
- Menggulir
- Pencetakan jarak jauh (melalui transaksi online)
- Tombol fungsi yang telah ditentukan sebelumnya
- Sistem batch dikirim dari transaksi online
- Pemilihan kursor data pada layar
- Penggunaan video terbalik, sorotan, warna yang menggarisbawahi dan indicator lainnya
- Dokumentasi pengguna hardcopy transaksi online
- Antarmuka mouse
- Pop-up jendela
- Sedikit layar untuk menyelesaikan fungsi bisnis
- Dukungan bilingual (dua bahasa)
- Dukungan multibahasa (lebih dari dua bahasa)

Tabel 4.11 Standar Documentation GSC End-user efisiensi

<b>Faktor</b>	<b>End user efficiency</b>
<b>Nilai</b>	<b>Deskripsi</b>
0	Bukan dari salah satu diatas
1	1 sampai 3 dari diatas
2	4 sampai 5 dari diatas
3	6 atau lebih hal diatas tapi tidak ada persyaratan khusus oleh pengguna yang berkaitan dengan efisiensi
4	6 atau lebih persyaratan diatas dan dinyatakan untuk efisiensi pengguna akhir membutuhkan tugas desain (misalnya memaksimalkan default, penggunaan template)
5	6 atau lebih persyaratan diatas dan dinyatakan untuk efisiensi pengguna akhir yang membutuhkan pemakaian alat khusus dan proses untuk menunjukkan bahwa tujuan telah tercapai

Tabel 4.12 Standar Documentation GSC Online Update

<b>Faktor</b>	<b>Online Update</b>
<b>Nilai</b>	<b>Deskripsi</b>
0	Tidak ada
1	Pembaruan online menyertakan satu sampai tiga file. Volume pembaruan rendah dan pemulihan mudah dilakukan
2	Pembaruan online menyertakan empat atau lebih file. Volume pembaruan rendah dan pemulihan mudah
3	Pembaruan online file logis internal utama disertakan

4	Perlindungan terhadap data yang hilang sangat penting dan telah dirancang dan deprogram secara khusus dalam sistem
5	Volume tinggi membawa pertimbangan biaya ke dalam proses pemulihan yang otomatis dengan menyertakan intervensi operator

Proses yang kompleks merupakan karakter dari suatu aplikasi. Berikut merupakan komponennya kompleksitas :

- Sensitive (contoh yaitu proses audit) dan /atau proses aplikasi yang spesifik
- Proses logis yang ekstensif
- Proses matematika yang ekstensif
- Banyak sekali proses yang dihindari sehingga menghasilkan transaksi yang tidak lengkap dan harus diolah kembali
- Pengolahan yang kompleks untuk menangani beberapa kemungkinan masukan atau keluaran.

Tabel 4.13 Standar Documentation GSC Complex Processing

<b>Faktor</b>	<b>Complex Processing</b>
<b>Nilai</b>	<b>Deskripsi</b>
0	Bukan dari salah satu di atas
1	Salah satu di atas
2	Dua di atas
3	Tiga di atas
4	Empat di atas
5	Kelima hal di atas

Tabel 4.14 Standar Documentation GSC Reuseability

<b>Faktor</b>	<b>Reusability</b>
<b>Nilai</b>	<b>Deskripsi</b>
0	Tidak ada kode yang dapat digunakan kembali

<b>Faktor</b>	<b>Reusability</b>
<b>Nilai</b>	<b>Deskripsi</b>
1	Kode dapat digunakan kembali digunakan dalam aplikasi
2	Kurang dari 10% aplikasi dianggap lebih dari satu kebutuhan pengguna
3	10% atau lebih dari aplikasi dipertimbangkan lebih dari satu kebutuhan pengguna
4	Aplikasi ini dikemas secara khusus dan / atau didokumentasikan untuk memudahkan penggunaan ulang dan aplikasi disesuaikan oleh pengguna
5	Aplikasi ini dikemas secara khusus dan / atau didokumentasikan untuk memudahkan penggunaan ulang dan aplikasi disesuaikan untuk digunakan dengan cara pemeliharaan parameter pengguna

Tabel 4. 15 Standar Documentation GSC Installation Ease

<b>Faktor</b>	<b>Installation Ease</b>
<b>Nilai</b>	<b>Deskripsi</b>
0	Tidak ada pertimbangan khusus yang dikemukakan oleh pengguna dan tidak perlu persiapan khusus untuk instalasi
1	Tidak ada pertimbangan khusus yang dikemukakan oleh pengguna tapi perlu pengaturan khusus untuk instalasi
2	Persyaratan konversi dan pemasangan dinyatakan oleh pengguna dan panduan konversi dan pemasangan disediakan dan diuji. Dampak konversi pada proyek tidak dianggap penting
3	Persyaratan konversi dan pemasangan dinyatakan oleh pengguna dan panduan

<b>Faktor</b>	<b>Installation Ease</b>
<b>Nilai</b>	<b>Deskripsi</b>
	konversi dan pemasangan disediakan dan diuji. Dampak konversi pada proyek ini dianggap penting
4	Selain 2 diatas, konversi dan instalasi otomatis disediakan dan diuji
5	Selain 3 diatas, otomatis konersi dan instalasi otomatis disediakan dan diuji

Tabel 4.16 Standar Documentation GSC Operational Ease

<b>Faktor</b>	<b>Operational Ease</b>
<b>Nilai</b>	<b>Deskripsi</b>
0	Tidak ada pertimbangan operasional khusus selain yang normal. Prosedur <i>back up</i> dinyatakan oleh pengguna
1 - 4	Satu, beberapa atau semua item berlaku untuk aplikasi. Setiap item ada benarnya kecuali yang dinyatakan lain. Proses <i>start up</i> , <i>back up</i> dan <i>recovery</i> yang efektif tapi membutuhkan intervensi operator Proses <i>start up</i> , <i>back up</i> dan <i>recovery</i> yang efektif tapi tidak memerlukan intervensi operator Aplikasi ini meminimalkan kebutuhan akan rekaman Aplikasi ini meminimalkan kebutuhan penanganan kertas
5	Aplikasi dirancang untuk operasi tanpa pengawasan. Operasi tanpa pengawasan berarti tidak adanya intervensi operator dalam pengoperasian sistem selain untuk <i>start up</i> atau <i>shut down</i> aplikasi. Pemulihan kesalahan otomatis adalah fitur dari aplikasi

Tabel 4.17 Standar Documentation GSC Multiple Sites

<b>Faktor</b>	<b>Multiple Sites</b>
<b>Nilai</b>	<b>Deskripsi</b>
0	Persyaratan pengguna tidak perlu mempertimbangkan kebutuhan lebih dari satu pengguna/situs instalasi
1	Kebutuhan beberapa situs dipertimbangkan dalam desain dan aplikasi dirancang untuk beroperasi hanya dibawah lingkungan perangkat keras dan lunak
2	Kebutuhan beberapa situs dipertimbangkan dalam desain dan aplikasi dirancang untuk beroperasi hanya dibawah yang serupa lingkungan perangkat keras dan /atau perangkat lunak
3	Kebutuhan beberapa situs dipertimbangkan dalam desain dan aplikasi ini dirancang untuk beroperasi di bawah yang berbeda dari perangkat keras dan /atau perangkat lunak
4	Dokumentasi dan rencana dukungan disediakan dan diuji untuk mendukung aplikasi di beberapa situs seperti nomor 1 dan 2
5	Dokumentasi dan rencana dukungan disediakan dan diuji untuk mendukung aplikasi di beberapa situs dan aplikasi seperti nomor 3

Aplikasi ini telah dirancang secara khusus, dikembangkan dan didukung untuk memudahkan perubahan. Karakteristik berikut dapat mengajukan permohonan untuk aplikasi:

- Permintaan dan fasilitas laporan yang ILFeksibel yang dapat menangani permintaan sederhana sebagai contoh logika hanya diterapkan pada satu file logis internal

- Permintaan dan fasilitas laporan yang ILFeksible yang dapat menangani permintaan dengan kompleksitas rata-rata sebagai contoh lebih dari satu file logis internal
- Permintaan dan fasilitas laporan yang ILFeksibel yang dapat menangani permintaan yang rumit misalnya pada satu atau lebih file logis internal
- Data kontrol bisnis disimpan dalam *tabel* yang dikelola oleh pengguna secara online tapi perubahannya berlaku pada hari kerja berikutnya
- Data kontrol bisnis disimpan dalam *tabel* yang dikelola oleh pengguna secara online dan perubahan berlaku segera

Tabel 4.18 Standar Documentation GSC Facilitate Change

<b>Faktor</b>	<b>Facilitate Change</b>
<b>Nilai</b>	<b>Deskripsi</b>
0	Bukan dari salah satu di atas
1	Salah satu diatas
2	Dua diatas
3	Tiga diatas
4	Empat diatas
5	Semua hal diatas

Pada tahap ini implementasi ini, hambatan yang penulis dapatkan adalah saat ingin menentukan berapa bobot yang sesuai dengan aplikasi. Hingga akhirnya dalam memastikannya, penulis melakukan *Benchmarking* dengan pihak ketiga. Sehingga hasil yang nantinya akan penulis dapatkan dapat diterima oleh pihak ketiga, karena sebelumnya nilai yang tertera sudah melalui proses *Benchmarking*.

Pada tahap ini persiapan yang penulis lakukan adalah dengan melakukan perancangan tabel template hasil perhitungan RCAF (**Tabel 4.19**).

Tabel 4.19 Template Hasil Perhitungan RCAF

No	Karakteristik	Bobot
1.	Tingkat kompleksitas Komunikasi Data	[0/1/2/3/4/5]
2.	Tingkat kompleksitas Pemrosesan Terdistribusi	[0/1/2/3/4/5]
3.	Tingkat kompleksitas Performance	[0/1/2/3/4/5]
4.	Tingkat kompleksitas konfigurasi	[0/1/2/3/4/5]
5.	Tingkat frekuensi penggunaan <i>software</i>	[0/1/2/3/4/5]
6.	Tingkat frekuensi <i>Input Data</i>	[0/1/2/3/4/5]
7.	Tingkat Kemudahan penggunaan bagi <i>user</i>	[0/1/2/3/4/5]
8.	Tingkat frekuensi <i>update data</i>	[0/1/2/3/4/5]
9.	Tingkat kompleksitas Processing Data	[0/1/2/3/4/5]
10.	Tingkat kemungkinan penggunaan kembali	[0/1/2/3/4/5]
11.	Tingkat kemudahan dalam instalasi	[0/1/2/3/4/5]
12.	Tingkat kemudahan operasional <i>software (backup, recovery, dsb)</i>	[0/1/2/3/4/5]
13.	Tingkat <i>software</i> dibuat untuk multi organisasi/perusahaan/cLEIFnt	[0/1/2/3/4/5]
14.	Tingkat kompleksitas dalam mengikuti perubahan/ILFeksibel	[0/1/2/3/4/5]
<b>TOTAL</b>		

Keterangan :

Total : Diisi dengan menjumlahkan seluruh bobot kriteria dari 1 hingga 14 yang dilingkari (O) atau sesuai dengan kriteria aplikasi yang akan dibangun.

### Langkah 3. Menghitung *Function point (FP)*

Langkah 3 bertujuan untuk mencari tahu nilai *Function point*. Pada tahap ini penulis tidak melakukan perancangan khusus,

karena perhitungan hanya perlu memasukkan nilai kedalam persamaan yang sudah dijelaskan pada dasar teori. Adapun penulisan hasil *Function point* tidak perlu dituliskan secara khusus kedalam sebuah tabel.

#### Langkah 4. Estimasi *Effort*

Langkah 4 bertujuan untuk mencari tahu estimasi *effort* yakni mencari tahu jumlah SDM dan waktu optimal yang diperlukan untuk membangun aplikasi ini. Pada tahap ini penulis tidak melakukan perancangan khusus, karena perhitungan hanya perlu memasukkan nilai kedalam persamaan yang sudah dijelaskan pada dasar teori. Adapun penulisan hasil wstimasi *effort* tidak perlu dituliskan secara khusus kedalam sebuah tabel.

#### Langkah 5. Estimasi Biaya

Langkah ini adalah untuk menghitung estimasi biaya total pembangunan aplikasi. Pada perhitungan estimasi biaya total, sama seperti langkah 1 dan 2, penulis melakukan persiapan dengan melakukan perancangan tabel untuk menuliskan hasil perhitungan. Adapun hasil akan melibatkan kelompok aktivitas yang telah dijelaskan pada bab 2 dan standar gaji yang akan dicarikan oleh penulis untuk dimasukkan kedalam tabel hasil **(Tabel 4.20)**.

Tabel 4.20 Template Hasil Perhitungan Estimasi Biaya Total

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement					
Spesification & Design					
Coding					
Testing					

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
Acceptance and Deployment					
<b>Ongoing Activity</b>					
Project management					
Configuration Management					
Documentation					
Training and Support					
<b>Quality and Testing</b>					
Integration Testing					
Quality Assurance and Control					
<i>Evaluation and Testing</i>					
<b>TOTAL</b>					<b>?</b>

Keterangan :

- Jabatan : Diisi dengan salah satu sumber daya manusia yang bertanggungjawab terhadap aktivitas tersebut didasarkan pada pembagian kategori SDM pada Tabel 2.6. Contoh : *System Analyst*.
- Person : Diisi dengan jumlah sumber daya yang bertanggungjawab pada masing-masing aktivitas tersebut. Contoh : 2 orang.
- Days : Diisi dengan jumlah waktu yang dibutuhkan untuk menyelesaikan masing-masing aktivitas tersebut dalam satuan hari. Contoh : 20 hari.

- Standar Gaji : Diisikan dnegan nilai gaji per-hari mengikuti jabatan yang diisikan pada “jabatan” dan didasarkan pada tabel daftar gaji pada Tabel 2.6. Contoh : Rp 500.000.
- Total : Diisikan dengan hasil perkalian antara jumlah person, jumlah hari dan gaji harian yang diterima pada masing-masing aktivitas. Contoh : Jumlah SDM yang diperlukan 2 orang, selama 20 hari dengan gaji per-hari Rp 500.000 (20 hari kerja/bulan). Maka total yang harus dibayarkan dengan mengali seluruh data menjadi Rp 20.000.000 untuk satu jabatan tersebut sampai proyek selesai.
- Total : Diisikan dengan menjumlahkan total dari seluruh aktivitas yang telah didefinisikan sebelumnya. Ini akan menjadi estimasi biaya yang dipakai untuk membangun aplikasi tersebut.

Sesuai dengan perencanaan tersebut, adapun gaji yang akan digunakan sebagai dasar penentuan harga aplikasi berpacu pada *Indonesian Sallary Guide 2018* (**Tabel 4.21**) yang dikeluarkan oleh *Kelly Services 2018*.

Tabel 4.21 Daftar Tarif/Gaji Tenaga IT

Jabatan	Pendidikan/ Pengalaman	Gaji/bulan	Gaji/hari	Gaji/jam
<i>Project manger</i>	Sarjana (min 5-10 th)	Rp 20.000.000	Rp 1.000.000	Rp 125.000
<i>Business/System analyst</i>	Sarjana (min 3-6 th)	Rp 7.000.000	Rp 350.000	Rp 43.750
<i>Technical/Non-Technical Team leader</i>	Sarjana (min 5-8 th)	Rp 15.000.000	Rp 750.000	Rp 93.750
<i>Programmer</i>	Sarjana (min 3-8 th)	Rp 8.000.000	Rp 400.000	Rp 50.000
<i>Software QA/Test Analyst</i>	Sarjana (min 3-5th)	Rp 8.000.000	Rp 400.000	Rp 50.000

Selanjutnya penulis dapat melakukan perhitungan dan memasukkan hasil perhitungan kedalam tabel hasil yang telah dijelaskan pada tahap perancangan.

Dalam mengimplementasikan *Function point* dalam Tugas Akhir ini, penulis melakukan perhitungan harga aplikasi dengan memanfaatkan harga dari *Indonesian Sallary Guide 2018*. Namun disebabkan oleh adanya timeline pengerjaan dari pihak ketiga, sehingga perlu ada tambahan langkah berikutnya yakni konversi total waktu pengerjaan *Function point* kedalam waktu pengerjaan pihak ketiga.

### **Langkah 6. Konversi Waktu Pengerjaan dengan Estimasi Pihak Ketiga**

Konversi ini dilakukan agar penulis dapat membandingkan harga yang diperoleh dari *Function point* dengan harga pihak ketiga, untuk mengetahui mahal atau tidaknya produk tersebut. Adapun konversi ini dilakukan dengan 2 jenis analisis, yakni pemanfaatan waktu lembur dan penambahan jumlah SDM.

#### **A. Pemanfaatan Waktu Lembur**

Pada langkah ini, penulis perlu mencari tahu persamaan yang dapat dimanfaatkan untuk menghitung waktu lembur. Adapun persamaan yang dapat mencari waktu lembur seperti pada persamaan berikut.

$$\text{Waktu Kerja} = \frac{\text{Jam Kerja Awal}}{\text{Total jam kerja awal}} \times \text{Total Jami Kerja Baru}$$

Melalui persamaan tersebut, masing-masing aktivitas dapat memperoleh hari lembur masing-masing sesuai dengan besar bobot pekerjaan. Aktivitas dengan pekerjaan yang lebih lama akan memperoleh waktu lembur yang lebih banyak pula. Adapun perhitungan waktu lembur menggunakan persamaan sebagai berikut.

$$\text{Jam Lembur} = \text{Total Jam Kerja 8.5 Bulan} - \text{Jam Kerja Biasa di 5 Bulan}$$

## B. Penambahan Jumlah SDM

Pada langkah ini, penulis mempersiapkan tabel perubahan jumlah SDM dan jumlah hari kerja sebagai dampak dari penambahan SDM. Adapun perhitungan dilakukan secara manual oleh penulis, dengan mengalikan atau membagikan jumlah SDM yang sebelumnya dan mengurangi jumlah hari kerja setelah penambahan SDM dilakukan. Adapun hasil akan dimasukkan kedalam tabel penambahan SDM (**Tabel 4.22**).

Tabel 4.22 Template Penambahan Jumlah SDM

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
		BEFORE		AFTER	
Aktivitas	Jabatan	Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
<b>Software Development</b>					
<b>Ongoing Activity</b>					
<b>Quality and Testing</b>					
<b>TOTAL</b>					

Kemudian perhitungan estimasi biaya akan dilakukan dengan cara yang sama dengan perhitungan estimasi biaya sebelumnya dengan mengalikan seluruh jumlah komponen yakni waktu, jumlah SDM dan biaya kerja perhari.

### ➤ Fungsi Pemanding dengan Penyedia Jasa Lain

Tahapan ini dilakukan dengan *Benchmarking*. *Benchmarking* dilakukan dengan memanfaatkan internet untuk menghubungi beberapa penyedia jasa untuk berkonsultasi sehingga dapat

memperoleh estimasi untuk membandingkan harga pembangunan aplikasi dengan fitur yang mirip dengan aplikasi ini. Dalam implementasi *Benchmarking* penulis mempersiapkan list fitur utama sesuai dengan fitur penawaran pada Aplikasi XYZ yang akan dibandingkan (**Tabel 4.23**). Setiap fitur lain yang menjadi tambahan pada aplikasi pembading akan diperhitungkan untuk penentuan harga Aplikasi XYZ.

Tabel 4.23 Fungsi Pembanding

No	Kebutuhan Fungsional
1.	Login
2.	Aplikasi dapat menampilkan informasi saldo
3.	Aplikasi dapat menampilkan aktivitas transaksi yang telah dilakukan <i>customer</i>
4.	Aplikasi dapat melakukan pemesanan transportasi online, tiket vanue tertetu dan transaksi lain seperti hotel, pembelian barang, dan lain-lain
5.	Aplikasi dapat menampilkan informasi pendamping dan penjemput atlet
6.	Aplikasi dapat menampilkan informasi hotel yang berada disekitar lokasi <i>Event</i>
7.	Aplikasi dapat menampilkan informasi <i>mall</i> di sekitar lokasi pelaksanaan <i>Event</i> yang menjadi <i>official partner</i> dan diskon produk tertentu yang ada selama <i>Event</i> berlangsung
8.	Aplikasi dapat menampilkan infromasi jadwal dan lokasi pertandingan
9.	Aplikasi dapat dimanfaatkan untuk menyampaikan keluhan atau menanyakan hal terkait layanan yang tersedia pada aplikasi
10.	Aplikasi dapat memberikan petunjuk lokasi suatu
11.	Aplikasi dapat ditampilkan dalam berbagai bahasa, minima dua bahasa yakni Bahasa Inggris dan Bahasa Indonesia

Fungsi-fungsi tersebut akan menjadi dasar konsultasi penulis dengan beberapa penyedia jasa dalam mencari tahu harga pembangunan berdasarkan pandangan penyedia jasa. Kemudian penulis juga akan melakukan brainstorming dengan

pihak ketiga untuk memastikan harga awal pembangunan dengan menunjukkan hasil benchmark dan pemanfaatan web untuk mencari titik akhir harga pembangunan aplikasi XYZ.

### ➤ **Perbandingan Estimasi Harga dengan Penyedia Jasa Lain**

Teknik ini diperlukan untuk melengkapi kebutuhan terhadap perbandingan harga aplikasi ini dengan aplikasi lain yang memiliki fungsi yang sama. Para tahapan ini penulis perlu melakukan perancangan template hasil *Benchmarking* (**Tabel 4.24**).

Tabel 4.24 Template Tabel Perbandingan Aplikasi

No	Fungsi pembanding	Aplikasi A	Aplikasi B
1.			
2.			
3.			
4.			
5.			
<b>Total</b>			

Keterangan :

- Fungsi pembanding : Diisikan sesuai dengan fungsi pembanding yang sudah didefinisikan pada tabel 4.5.
- Aplikasi A,B : Diisi dengan checklist fungsi mana yang disediakan oleh penyedia jasa yang sesuai dengan fungsi pembanding, jika sesuai akan diisikan dengan tanda centang (  ), jika tidak sesuai akan diisi dengan tanda silang ( X ).
- Total : Diisikan dengan estimasi biaya yang diperoleh dari pihak penyedia jasa.

Selanjutnya, berdasarkan hasil perbandingan tersebut, penulis akan menyatitkan sebuah kesimpulan beserta justifikasinya yang menyatakan apakah biaya pembangunan fungsi aplikasi oleh pihak ketiga tersebut terbilang mahal atau tidak.

#### 4.2.2.2 *Function Analysis Phase*

##### ➤ **Perhitungan Biaya Kebutuhan Fungsional**

Pada tahap ini dilakukan perhitungan kebutuhan fungsional dengan memanfaatkan teknik *Function point*. Berdasarkan hasil *effort* dan waktu yang diperoleh, penulis akan melakukan pendistribusian kepada masing-masing aktivitas dan jabatan yang selanjutnya dapat dihitung untuk mencari tahu biaya masing-masing kebutuhan fungsional. Adapun pendistribusian tersebut akan dibagi kedalam 5 aktivitas dan 5 jabatan yang bertanggungjawab (**Tabel 4.25**).

Tabel 4.25 Template Breakdown effort dan Waktu Pengerjaan

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b><i>Software Development</i></b>					
Requirement	<i>Business analyst</i>			Rp 350.000	
Spesification & Design	<i>System analyst</i>			Rp 350.000	
Coding	<i>Programmer</i>			Rp 400.000	
Testing	<i>Technical tester</i>			Rp 400.000	
Acceptance and Deployment	<i>Technical Team leader</i>			Rp 750.000	
<b>TOTAL</b>					

Keterangan :

- Kelompok aktivitas : Diisikan sesuai dengan pembagian aktivitas pada tahap *software* development yang sudah dijelaskan pada bab 2. Contoh : *Requirement*.
- Jabatan : Diisi dengan jabatan yang akan bertanggungjawab pada aktivitas tersebut. Bagian ini juga telah didefinisikan pada Bab 2. Contoh : *Business analyst*. ✓
- Person : Diisikan dengan jumlah pekerja yang disesuaikan dengan hasil perhitungan *effort* yang diperoleh melalui teknik *Function point*. Contoh : *effort* = 8 orang. Requirement = 1 orang; Coding = 3 orang, dll.
- Days : Diisikan dengan jumlah waktu pengerjaan masing-masing aktivitas, yang disesuaikan dengan total waktu pengerjaan yang diperoleh dari teknik *Function point*. Contoh : Waktu = 80 hari. Requirement = 15 hari; Coding = 35 hari, dll.
- Standar Gaji : Diisikan dengan harga gaji perhari sesuai dengan jabatan masing-masing. Contoh : Rp 400.000.
- Total : Diisikan dengan hasil yang diperoleh dengan mengalikan person, days dan standart gaji. Contoh : person = 2; days = 15 hari; Gaji = Rp 400.000; Total = Rp 12.000.000
- TOTAL : Diisikan dengan jumlah total biaya masing-masing aktivitas.

### ➤ Pemilihan Objek Penelitian

Pada tahap ini, penulis akan melakukan pembentukan FAST Diagram. Namun sebelum membuat FAST Diagram, menggunakan 2 cara analisis yakni : Pertama, melakukan perhitungan bobot, penulis tidak mengalami kesulitan dalam

mengimplementasikannya, data yang penulis miliki, hasil perhitungan tahap sebelumnya dan fasilitas yang penulis miliki seperti kalkulator, cukup untuk memudahkan penulis dalam melakukan perhitungan. Adapun persamaan yang digunakan untuk mencari tahu nilai bobot yakni :

$$\text{Bobot} = \frac{\text{Biaya fitur}}{\text{Biaya total fitur}} \times 100\%$$

Untuk memperoleh nilai biaya fungsi, perhitungan akan dilakukan sama seperti sebelumnya yakni memanfaatkan teknik *Function point*. Biaya total fungsi diperoleh dengan menjumlahkan seluruh biaya per-fungsi. Persiapan yang penulis lakukan sama seperti pada tahapan sebelumnya dengan tambahan, pembuatan template hasil harga per-fungsi yang diperoleh (**Tabel 4.26**).

Tabel 4.26 Template Estimasi Biaya Kebutuhan Fungsional

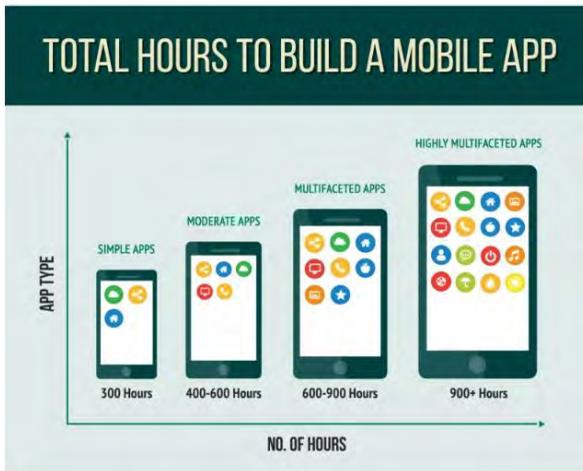
No	Kebutuhan Fungsional	FP	Biaya	Bobot
1.				
2.				
3.				
<b>TOTAL</b>				<b>?</b>

Keterangan :

Kebutuhan Fungsional : Diisikan sesuai dengan kebutuhan fungsional yang ada pada aplikasi ini. Terdapat 10 kebutuhan fungsional yang tercatat dalam KAK. Contoh : Melihat informasi Saldo.

- FP : FP (*Function point*). Diisi dengan nilai *Function point* yang diperoleh dari hasil perhitungan. Contoh : 45.
- Biaya : Diisikan dengan estimasi biaya yang diperoleh dari pihak penyedia jasa. Contoh : Rp 56.000.000.
- Bobot : Diisi dengan hasil pembagian antara biaya per-fungsi dan total harga pembangunan per-fungsi. Contoh: Biaya fungsi Rp 56.000.000, biaya total seluruhnya Rp112.000.000, sehingga dengan mengikuti persamaan bobot fungsi ini adalah 20%.
- Total : Diisikan dengan estimasi biaya yang diperoleh dari hasil brainstorming dengan pihak ketiga tersebut.

Kedua, penulis menentukan objek yang akan diolah disesuaikan dengan syarat waktu pengerjaan aplikasi yang masuk kedalam kategori *complex* dan *highly complex*. (**Gambar 4.1**) [30].



Gambar 4.1 Pengelompokan Tipe Fungsi Aplikasi Mobile

Berdasarkan gambar tersebut, dapat kita lihat bahwa suatu fungsi yang dikatakan *complex* adalah fungsi yang waktu pengerjaannya berkisar antara 600-900 jam dengan estimasi 2-4 tim yang bekerja. Sedangkan *highly complex* adalah fungsi yang pengerjaannya membutuhkan waktu diatas 900 jam dengan estimasi pekerja lebih dari 4 tim. Berdasarkan hasil tersebut, penulis dapat menentukan objek yang digunakan pada FAST Diagram. Kemudian hasil tersebut akan dimasukkan kedalam tabel penentuan objek penelitian (**Tabel 4.27**).

Tabel 4.27 Template Penentuan Objek Penelitian

No	Kebutuhan Fungsional	Days (Jam)	Jumlah SDM	Keterangan

Keterangan :

- No : Diisikan dengan urutan kebutuhan fungsional
- Kebutuhan fungsional : Diisikan sesuai dengan kebutuhan fungsional yang ada pada aplikasi ini. Terdapat 11 kebutuhan fungsional yang tercatat dalam KAK. Contoh : Melihat infomrasi Saldo.
- Days : Diisikan dengan jumlah waktu yang dibutuhkan untuk menyelesaikan masing-masing aktivitas tersebut dalam satuan hari. Contoh : 20 hari.
- Person : Diisi dengan jumlah sumber daya yang bertanggungjawab pada masing-masing aktivitas tersebut. Contoh : 2 orang.

Keterangan : Diisikan dengan status fungsi, apakah akan diterima (menjadi objek yang akan dibahas pada tahapan selanjutnya) atau ditolak (pengolahan tidak dilanjutkan).

### ➤ **Pembangunan FAST Diagram**

Dalam pembangunan FAST Diagram penulis tidak memerlukan persiapan khusus. Penulis hanya perlu memahami unsur-unsur dalam FAST Diagram seperti *high order Function*, *support Function*, *low order Function*, dan lain-lain seperti yang dijelaskan pada dasar teori. Pembangunan FAST Diagram hanya dilakukan pada fungsi *complex* yang dipilih berdasarkan hasil pembobotan.

#### **4.2.2.3 Creativity Phase**

Pada fase ini, dalam melakukan pengolahan data pada skenario 1 (Pemanfaatan waktu lembur) dan 2 (Penambahan jumlah SDM) penulis tidak memerlukan persiapan khusus, penulis memerlukan persiapan berupa pemahaman hal-hal penting terkait dengan pembangunan suatu proyek aplikasi TI, sehingga dapat membuat skenario-skenario yang memungkinkan untuk diimplementasikan dalam proyek, kemudian penulis perlu mengikuti langkah dan format hasil yang sama seperti tahapan sebelumnya, menggunakan tabel perubahan hari kerja, estimasi kerja menggunakan jam lembur, perubahan jumlah SDM dan estimasi biaya perubahan SDM.

Pada skenario 3, sehubungan dengan Pemanfaatan Visual Programming, penulis perlu mengetahui tugas dari masing-masing jabatan yang terlibat dalam *software development* dan kegiatan yang dilakukan pada masing-masing aktivitas sebagai berikut:

- ***Business analyst***

Fokus utama bisnis analis adalah mengidentifikasi peluang untuk mengembangkan proses bisnis dan menggunakan

teknologi untuk menghilangkan masalah terkait produktivitas, *Output*, distribusi dan akhirnya batas bawah. Sehingga, mengetahui bagaimana teknologi dapat menyelesaikan permasalahan bisnis adalah hal yang sangat penting bagi keberhasilan seorang bisnis analis. Berikut merupakan tugas seorang bisnis analis :

1. Menganalisis proses bisnis dalam suatu organisasi atau perusahaan untuk hal yang tidak efisien dan tidak berkecukupan. Membuat rekomendasi dan persetujuan untuk solusi atau pengembangan yang dapat dicapai melalui pengimplementasian teknologi atau penggunaan alternatif dari teknologi yang sudah ada.
2. Bertindak sebagai penghubung antara *stakeholders* seperti management, pelanggan atau *end users*. Dan *software development* atau *Information technology team*.
3. Menganalisis dan mengkomunikasikan kebutuhan *stakeholder* dengan menterjemahkan kebutuhan bisnis kedalam kebutuhan *software*.
4. Mendokumentasikan dan mengevaluasi data dan informasi yang dibutuhkan.
5. Menggunakan permodelan, pengujian dan data untuk meningkatkan arus informasi melalui organisasi untuk meningkatkan keberhasilan proyek.

- ***System analyst***

Sistem analis menggunakan teknologi informasi perusahaan untuk membantu mencapai beberapa sasaran dan tujuan bisnis yang strategis. Mereka dapat merancang dan mengembangkan *system* baru dengan mengkonfigurasi perangkat keras dan perangkat lunak baru, atau menggunakan *system* yang ada dengan cara-cara baru untuk mencapai hasil tambahan. Tugas dari *system* analis adalah :

1. Berkonsultasi dengan manajemen dan pengguna untuk menentukan kebutuhan *system*

2. Merancang sebuah *system* untuk memenuhi tujuan dan sasaran bisnis
3. Menggunakan teknik seperti sampling, model, analisis terstruktur, bersamaan dengan prinsip-prinsip akuntansi, untuk memastikan efisiensi solusi, biaya yang efektif dan layak secara finansial.
4. Mengembangkan diagram spesifikasi dan *flowchart* untuk menjadi acuan *programmer*.
5. Mengawasi implementasi, mengkoordinasikan tes dan mengamati inisiasi *system* untuk memvalidasi kinerja.

- ***Programmer***

Fungsi dasar dari *programmer* adalah untuk mempergunakan pengetahuan teknik pemrograman dan sistem komputer untuk membuat program komputer untuk melakukan bermacam-macam pekerjaan sesuai dengan persetujuan dengan *client*. Tugas utama programmer meliputi:

1. Mengkonfirmasi persyaratan proyek dengan meninjau tujuan program, memasukkan data, dan persyaratan *Output* dengan analis, pengawas, dan klien.
2. Mengatur persyaratan proyek dalam urutan pemrograman dengan menganalisis persyaratan; menyiapkan diagram alur kerja dan diagram menggunakan pengetahuan tentang kemampuan komputer, materi pelajaran, bahasa pemrograman, dan logika.
3. Mengkodekan persyaratan proyek dengan mengubah informasi alur kerja menjadi bahasa komputer.
4. Program komputer dengan memasukkan informasi yang dikodekan.
5. Konfirmasi operasi program dengan melakukan tes; memodifikasi urutan dan / atau kode program.
6. Menyiapkan referensi untuk pengguna dengan menulis instruksi operasi.
7. Mempertahankan catatan sejarah dengan mendokumentasikan pengembangan dan revisi program.

8. Menjaga kepercayaan klien dan melindungi operasi dengan menjaga kerahasiaan informasi.
9. Memastikan pengoperasian peralatan dengan mengikuti petunjuk pabrik; pemecahan masalah malfungsi; menyerukan perbaikan; mengevaluasi peralatan dan teknik baru.
10. Mempertahankan pengetahuan profesional dan teknis dengan menghadiri lokakarya pendidikan; meninjau publikasi profesional; membangun jaringan pribadi; berpartisipasi dalam masyarakat profesional.
11. Berkontribusi untuk upaya tim dengan mencapai hasil terkait sesuai kebutuhan

- ***Technical Tester***

Teknikal terster diperlukan untuk sepenuhnya menguji produk atau sistem untuk memastikan berfungsi dengan baik dan memenuhi kebutuhan bisnis. Tanggung jawab pekerjaan meliputi:

1. Menguji semua aspek produk / sistem seperti fungsi / komponen, sistem, kinerja, regresi, dan layanan.
2. Bekerja dengan tim pengembangan untuk mengidentifikasi dan menangkap kasus uji, memastikan manajemen versi
3. Menyiapkan lingkungan pengujian, merancang rencana uji, mengembangkan kasus uji / skenario / kasus penggunaan, dan mengeksekusi kasus-kasus ini.
4. Memberikan umpan balik tentang kegunaan dan kemudahan servis, melacak hasilnya ke risiko kualitas dan melaporkannya kepada orang-orang yang peduli.

- ***Technical Team leader***

*Technical Team leader* bertanggung jawab untuk mengembangkan dan menginstal perangkat lunak untuk sistem komputer organisasi mereka, serta memecahkan masalah teknis untuk keberhasilan pelaksanaan proyek. Pemimpin tim teknis bekerja dengan perusahaan perangkat lunak dan perusahaan

teknologi informasi dengan deskripsi pekerjaan mereka yang melibatkan penyediaan solusi untuk masalah teknis yang muncul yang menghadapi kemajuan produk dan layanan organisasi mereka. Aktivitas pekerjaan yang dilakukan oleh *Technical Team leader* adalah sebagai berikut:

1. Membuat dan mengembangkan laporan kinerja, metode pengiriman, ruang lingkup kerja, dan catatan tugas umum
2. Pergi ke klien untuk pengarahan proyek, konsultasi, instalasi dan ulasan close out.
3. Bantu dalam mengelola permintaan pelanggan untuk memastikan kepuasan maksimal, dan untuk menjaga kualitas daripada kuantitas
4. Terlibat dalam negosiasi tuntutan dan spesifikasi pekerjaan pelanggan terkait dengan tenaga kerja dan material; dan membantu dalam membuat dokumen teknis yang komprehensif
5. Mengawasi kegiatan antara sumber daya internal dan eksternal, dan memfasilitasi alur kerja yang lancar untuk penyampaian layanan
6. Evaluasi semua barang yang disediakan sesuai pesanan untuk keawetan dan keakuratan
7. Mengevaluasi data proyek untuk akurasi, dan memimpin dalam menetapkan target dan prioritas proyek
8. Selesaikan perselisihan antara anggota tim dan manajemen dan bantu dalam mengatasi masalah di antara anggota tim untuk menghindari perilaku yang tidak dapat diterima
9. Berikan pelatihan dan bimbingan kepada anggota tim untuk membuat mereka lebih baik dalam pekerjaan
10. Up to date dengan praktik lapangan saat ini untuk meningkatkan standar dan pengiriman layanan
11. Tinjau permintaan dan petunjuk teknis pelanggan dan bantu tim internal dan pelanggan untuk mengidentifikasi pesanan terbaik untuk lift, dan metode yang paling efisien dalam mempertahankan dan menggunakan materi yang dikirim

12. Hadiri pelatihan teknis untuk memperoleh lebih banyak pengetahuan tentang prosedur dan aplikasi teknis, dan tuntutan pekerjaan lainnya
13. Tetap berpengetahuan teknologi saat ini dan melakukan penelitian untuk mengidentifikasi tren baru yang dapat digunakan untuk mencapai hasil yang maksimal
14. Melaksanakan tugas-tugas teknis terkait lainnya yang mungkin diperlukan, termasuk mealkukan serah-terima kepada pihak terkait.

Selain itu, penulis perlu mengetahui tanggungjawab yang dipegang pada masing-masing aktivitas yang terlibat dalam *software development* sebagai berikut [31].

- **Requirement**

Tahapan ini merupakan tahapan pertama dalam SDLC (*Software Development Life Cycle*) untuk pembangunan *software*. Fase ini meliputi komunikasi antara *project stakeholders*, pengguna akhir (*customer*) dan tim proyek, karena kebutuhan (fungsional dan non-fungsional) akan dikumpulkan melalui *client*. Fase ini akan meliputi aktivitas :

1. Analisis kelayakan fungsional dan finansial.
2. Mengidentifikasi dan mendokumentasikan kebutuhan dari *stakeholder* melalui wawancara klien, survey, dan lain-lain.
3. Pendefinisian secara jelas dan dokumentasi kebutuhan dalam SRSD (*Software Requirement Specification Document*) yang terdiri dari seluruh kebutuhan produk yang akan dikembangkan.
4. Membuat *prototype* dari proyek, untuk menunjukkan kepada *customer* bagaimana proyek tersebut akan dihasilkan. Performa ini akan dibantu dengan *tools* seperti Azure, dan lain-lain.

- **Specification and Desain**

Dalam tahap ini, berdasarkan pada hasil yang diperoleh dari tahap *requirement* pada SRSD, desain arsitektur ini ditujukan

untuk proyek dan mendokumentasikannya. Fase ini akan meliputi aktivitas :

1. Pembagian kebutuhan kedalam *system* perangkat keras dan perangkat lunak.
2. Mendesain arsitektur *system*.
3. Membuat UML diagram seperti *use case*, *class diagram*, *sequence diagram*, dan *activity diagram*).

- **Coding**

Tahapan ini adalah tahapan paling panjang dalam SDLC. Pada tahap ini proyek akan dibangun dan dikembangkan. Fase ini akan meliputi aktivitas :

1. Penulisan kode actual.
2. Demonstrasi pekerjaan yang telah dilakukan kepada *Business analyst* untuk modifikasi yang sesuai dengan kebutuhan dimasa depan.
3. Pengujian dilakukan yakni dengan melakukan verifikasi kode sesuai dengan kebutuhan.

- **Testing**

Tahapan ini terlibat hampir diseluruh tahapan dalam SDLC. Walaupun fase dari SDLC ini lebih mengarah kepada pengujian sistem yang dilaporkan mengalami *bugs/defect*, dilacak dan diperbaiki. Sistem atau proyek beralih kepada pengujian lingkungan dan berbagai jenis pengujian seperti fungsional, integrasi, sistem, dan penerimaan. Performa ini dilakukan sampai proyek mendapatkan standar kualitas yang telah dijabarkan secara spesifik pada SRS. Fase ini akan meliputi aktivitas :

1. Pengujian *system* secara keseluruhan.
2. Pelaporan berbagai jenis pengujian yang dilakukan dan perbaikan *bugs*.

- **Acceptance and Deployment**

Pada fase SDLC, ketika sistem telah diuji, maka sistem siap untuk dipergunakan secara *live*. Pertama sistem mungkin perlu diberikan batasan *user*, dan dilakukan pengujian lingkungan bisnis secara nyata untuk UAT (*User Acceptance Testing*). Fase ini akan meliputi aktivitas :

1. Memastikan bahwa *system* telah siap untuk diserahkan.
2. Memastikan sistem telah terinstal dan mulai digunakan.
3. Memeriksa kesalahan yang mungkin belum terdeteksi sebelumnya.
4. Memperbaiki *system* dan akhirnya meningkatkan *system* di pusat data

Berdasarkan penjelasan tugas tersebut, penulis dapat mencari hubungan masing-masing aktivitas dan jabatan untuk melakukan analisis apakah jika terdapat kemungkinan pemindahan tugas.

Selain itu penulis coba mencari alternative lain. Dengan menggunakan acuan [wiki.c2.com](http://wiki.c2.com) penulis mendapatkan informasi tentang visual programming. Dimana dalam pembangunan sebuah program, hanya diperlukan desain yang baik dan benar, kemudian running kedalam visual programming dan terbentuklah program. Sumber yakni Doug Serres telah mengimplementasikan salah satu visual programming yakni rasional rose selama 4 tahun, merupakan ekspertis dibidang programming, yang juga banyak memberikan masukan dalam situs <http://wiki.c2.com/> sehubungan dengan programming. Dalam situs tersebut beliau membagi pengalaman membutuhkan waktu 2 bulan untuk melakukan spesifikasi dan desain dan hanya membutuhkan 11 hari untuk membangun aplikasi (**Gambar 4.2**)

Task Name	Duration	Start	Finish
<b>Comp1</b>	<b>9.5 days</b>	<b>Fri 3/11/05</b>	<b>Thu 3/24/05</b>
Design Inspection	0.5 days	Fri 3/11/05	Fri 3/11/05
code	5 days	Fri 3/11/05	Fri 3/18/05
<b>Code Review/R2A</b>	<b>1.5 days</b>	<b>Fri 3/18/05</b>	<b>Mon 3/21/05</b>
Run tool	0.5 days	Fri 3/18/05	Fri 3/18/05
Fix problems	1 day	Mon 3/21/05	Mon 3/21/05
Write/Automate Unit Test:	1 day	Fri 3/18/05	Mon 3/21/05
Run Unit Tests	0.5 days	Mon 3/21/05	Mon 3/21/05
Collect Test Results/Metri	0.5 days	Tue 3/22/05	Tue 3/22/05
Debug/Fix for 100% pass	2 days	Tue 3/22/05	Thu 3/24/05

Gambar 4.2 Pembagian Aktivitas Penggunaan Rasional Rose [32]

Penulis menggunakan *review* ini untuk digunakan sebagai salah satu skenario karena dengan mengurangi waktu kerja dan jumlah SDM bagian coding, juga dapat mengurangi biaya pengeluaran [31].

Hasil akhir pada tahapan ini dimasukkan dalam template hasil pencarian alternative (**Tabel 4.28**).

Tabel 4.28 Template Hasil Pencarian Alternatif Kebutuhan Fungsi

No.	Kebutuhan Fungsional	Alternatif

Keterangan :

No : Diisikan sesuai urutan jenis fungsi. Contoh : 1,2, dst.

Kebutuhan Fungsional : Diisikan sesuai dengan kebutuhan fungsional yang ada pada aplikasi ini. Terdapat 11 kebutuhan fungsional yang tercatat dalam KAK. Contoh : Melihat informasi Saldo.

Alternatif Kebutuhan Fungsional : Diisikan sesuai dengan skenario yang berhasil dibuat. Contoh : Penambahan SDM.

#### 4.2.2.4 Evaluation Phase

Tahapan ini dilakukan setelah diperoleh berbagai alternatif terkait objek penelitian, namun sebelumnya penulis perlu merangkum hasil perolehan biaya berdasarkan skenario yang telah dibuat kedalam **Tabel 4.29**. Masing-masing tabel akan diisikan dengan kategori skenario yang sama. Contoh terdapat 3 skenario, maka tabel akan dibuat sebanyak 3 sesuai dengan skenario.

Tabel 4.29 Template Hasil Perubahan Biaya berdasarkan Skenario

No.	Kebutuhan Fungsional	Biaya awal	Alternatif Kebutuhan Fungsional	Biaya Akhir
		<b>TOTAL :</b>	<b>TOTAL :</b>	

Keterangan :

No : Diisikan sesuai urutan jenis fungsi. Contoh : 1,2, dst.

Kebutuhan Fungsional : Diisikan sesuai dengan kebutuhan fungsional yang ada pada aplikasi ini. Terdapat 11 kebutuhan fungsional yang

tercatat dalam KAK. Contoh : Melihat informasi Saldo.

- Biaya awal : Diisikan biaya awal pembangunan fungsi sebelum dilakukan skenario. Contoh : Rp 20.000.000
- Alternatif Kebutuhan Fungsional : Diisikan sesuai dengan skenario yang berhasil dibuat. Contoh : Penambahan SDM.
- Biaya akhir : Diisikan dengan biaya yang diperoleh setelah dihitung berdasarkan skenario. Contoh : Rp 15.000.000.
- TOTAL : Diisikan dengan jumlah biaya awal dan jumlah biaya akhir.

### ➤ Pencarian Indeks Nilai

Pada tahapan ini, penulis perlu mengetahui nilai perubahan yang diperoleh sebelum dan setelah perhitungan skenario. Masing-masing skenario akan dihitung dan hasil yang diperoleh akan dimasukkan kedalam **Tabel 4.30**, untuk melihat mana skenario yang layak dan tidak layak untuk dipertimbangkan.

Tabel 4.30 Template Hasil Pencarian Indeks Nilai

No	Alternatif Kebutuhan Fungsional	Hasil Pencarian Indeks	Keterangan

Keterangan :

- No : Diisikan sesuai urutan jenis fungsi. Contoh : 1,2, dst.

Alternatif : Diisikan sesuai dengan skenario yang  
Kebutuhan berhasil dibuat. Contoh : Penambahan  
Fungsional SDM.

Hasil : Diisikan dengan nilai yang diperoleh  
Pencarian setelah perhitungan dilakukan. Contoh : 1.  
Indeks

Keterangan : Diisikan dengan keterangan sesuai kategori  
berikut :

 :  $Nt/Np < 1$

 :  $Nt/Np = 1$

 :  $Nt/Np > 1$

Berdasarkan hasil perhitungan indeks, penulis akan mengambil alternative dengan keterangan berwarna hijau atau  $Nt/Np > 1$ .

#### ➤ **Pemilihan Rekomendasi**

Dalam memilih rekomendasi skenario yang paling optimal, penulis menggunakan prinsip hubungan waktu dan biaya. Dalam pengerjaan proyek, waktu tercepat dan biaya yang paling murah akan menjadi rekomendasi yang paling optimal, sehingga dalam pengerjaan Tugas Akhir memilih skenario dengan waktu tercepat dan biaya paling murah sebagai rekomendasi.

*Halaman ini sengaja dikosongkan*

## BAB V IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai proses pelaksanaan penelitian. Tahapan ini berguna untuk mengetahui bagaimana penelitian dilakukan, penerapan strategi pelaksanaan, rancangan, hambatan, dan lain-lain.

### 5.1 *Information Phase*

#### 5.1.1 **Ulasan Dokumen Internal**

Dalam penelitian ini, dokumen internal yang diulas oleh penulis adalah KAK (Kerangka Acuan Kerja). Dengan melakukan ulasan dokumen internal usulan pihak ketiga terdapat beberapa hasil yang diperoleh:

##### 5.1.1.1 **Kebutuhan Fungsional Aplikasi**

Berdasarkan usulan pihak ketiga, setidaknya terdapat 11 kebutuhan fungsional yang ingin ditampilkan dalam aplikasi untuk menjawab kebutuhan pelanggan seperti pada **Tabel 5.1**.

Tabel 5.1 Kebutuhan Fungsional Aplikasi

No	Kebutuhan Fungsional
1.	Login
2.	Aplikasi dapat menampilkan informasi saldo
3.	Aplikasi dapat menampilkan aktivitas transaksi yang telah dilakukan <i>customer</i>
4.	Aplikasi dapat melakukan pemesanan transportasi online, tiket <i>vanue</i> tertentu dan transaksi lain seperti hotel, pembelian barang, dan lain-lain
5.	Aplikasi dapat menampilkan informasi pendamping dan penjemput atlet
6.	Aplikasi dapat menampilkan informasi hotel yang berada disekitar lokasi <i>Event</i>
7.	Aplikasi dapat menampilkan informasi <i>mall</i> di sekitar lokasi pelaksanaan <i>Event</i> yang menjadi <i>official partner</i> dan diskon produk tertentu yang ada selama <i>Event</i> berlangsung
8.	Aplikasi dapat menampilkan informasi jadwal dan lokasi pertandingan

No	Kebutuhan Fungsional
9.	Aplikasi dapat dimanfaatkan untuk menyampaikan keluhan atau menanyakan hal terkait layanan yang tersedia pada aplikasi
10.	Aplikasi dapat memberikan petunjuk lokasi suatu
11.	Aplikasi dapat ditampilkan dalam berbagai bahasa, minima dua bahasa yakni Bahasa Inggris dan Bahasa Indonesia

### 5.1.1.2 Pihak Penanggungjawab Pembangunan Aplikasi

Dalam mewujudkan kebutuhan fungsional tersebut kedalam fungsi aplikasi, pihak ketiga memberikan usulan jabatan dan gaji masing-masing jabatan sebagai acuan penentuan harga pembangunan aplikasi seperti pada **Tabel 5.2**.

Tabel 5.2 Pihak Penanggungjawab Pembangunan Aplikasi

NO	SUMBER DAYA	NN	SATUAN	RATE / SAT	QTY
1	<b>PROJECT MANAGER</b>	PM	ManDays	Rp 500,000	1 orang
2	<b>BUSINESS ANALYST</b>	BA	ManDays	Rp 300,000	2 orang
3	<b>Team MANAGER</b>	TM	ManDays	Rp 300,000	1 orang
4	Technical - Team Leader	TTL	ManDays	Rp 250,000	1 orang
5	System Analyst	SA	ManDays	Rp 225,000	2 orang
6	Programmer	PR	ManDays	Rp 150,000	4 orang
7	Technical SUPPORT	TS	ManDays	Rp 100,000	2 orang
8	Technical TESTER	Tst	ManDays	Rp 100,000	2 orang
9	<b>Non Technical - Team Leader</b>	NTM	ManDays	Rp 250,000	1 orang
10	Support	Sup	ManDays	Rp 100,000	2 orang
11	Others	Oth	ManDays	Rp 75,000	2 orang
12	SECURITY Specialist	SSp	Mandays	Rp 250,000	1 orang

Berdasarkan **Tabel 5.2** pihak ketiga memberikan perbedaan warna pada masing-masing *field* artinya adalah sebagai berikut.

1. Merah – *Project manger*, merupakan pihak tertinggi yang mengatur jalannya proyek, mengawasi dan menjadi pihak yang akan bertanggungjawab dengan seluruh kinerja karyawan pada proyek tersebut.
2. Abu-Abu – *Business analyst*, merupakan pihak yang akan menerima konsultasi dari karyawan pada tingkatan dibawahnya yakni *Team manager* dan *security specialist*.

3. Hijau – *Team manager*, merupakan pihak yang akan bertanggungjawab, terhadap kinerja *Technical* dan *Non-Technical Team leader*, melakukan pengawasan serta menerima konsultasi dari mereka.
4. Cream – *Technical Team leader*, merupakan pihak yang akan bertanggungjawab pada pihak dibawahnya, sekaligus melakukan pengawasan dan menerima konsultasi secara langsung dari *system analyst*, *programmer*, *Technical support* dan *Technical tester*.
5. Biru – *Non-Technical Team leader*, merupakan pihak yang akan bertanggungjawab, mengawasi secara langsung dan menerima konsultasi terkait kebutuhan yang berhubungan dengan *support* dan *others*.
6. Ungu – *security specialist*, merupakan pihak yang bersama dengan *Team manager* akan bertanggungjawab akan bidang masing-masing kepada *Business analyst*.

### 5.1.1.3 Tahapan Pengerjaan Aplikasi

Tahapan pengerjaan aplikasi berdasarkan usulan pihak ketiga terdiri atas 12 aktivitas seperti pada **Tabel 5.3**.

Tabel 5.3 Tahapan Pengerjaan Aplikasi

NO	TAHAPAN PEKERJAAN
1.	Analisa Kebutuhan Sistem
2.	Analisa Kebutuhan Aplikasi
3.	Desain
4.	Pengkodean
5.	Pengujian
6.	Pengujian Internal
7.	Pengujian Eksternal
8.	Finalisasi Pengujian
9.	Pelatihan
10.	Serah Terima

NO	TAHAPAN PEKERJAAN
11.	Pemeliharaan (termasuk Jaminan Security)
12.	Proyek Selesai

#### 5.1.1.4 Waktu Pengerjaan Aplikasi

Pembangunan aplikasi memiliki waktu pengerjaan selama 7 bulan dengan pembagian 5 bulan waktu pembangunan aplikasi dan 2 bulan waktu pengembangan aplikasi. Dalam melakukan perhitungan pada tahapan selanjutnya, penulis menggunakan perbandingan biaya pembangunan selama 5 bulan dengan total biaya Rp 439.300.000. Adapaun detail pembagian kerja usulan pihak ketiga adalah seperti pada **Tabel 5.4**.

Tabel 5.4 Detail Waktu Pengerjaan Pembangunan Aplikasi

NO	TAHAPAN KEGIATAN	TAHAP BULAN MINGGU	TAHAP																				TOTAL Mandays
			FEBRUARI				MARET				APRIL				MEI				JUNI				
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	PROJECT MANAGER	55,000,000.00	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	110
2	BUSINESS ANALYST	12,000,000.00	3	3	3	3	2	2															20
3	TEAM MANAGER	33,000,000.00	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	110
4	Technical - Team Leader	27,500,000.00	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	110
5	System Analyst	25,200,000.00			5	7	5	5	5	7	5	5	5	7									56
6	Programmer	60,000,000.00			5	7	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	100
7	Technical SUPPORT	8,800,000.00													5	5	5	7	5	5	5	7	44
8	Technical TESTER	17,600,000.00					5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	88
9	Non Technical - Team Leader	27,500,000.00	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	110
10	Support	20,000,000.00			5	7	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	100
11	Others	16,500,000.00	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	5	5	5	7	110
12	SECURITY Specialist	1,000,000.00																			2	2	4
	TOTAL BIAYA PRODUCTION (belum termasuk Security Spc)	303,100,000.00																					

TAHAPAN KEGIATAN	TAHAP BULAN MINGGU	TAHAP							
		JULI				AGUSTUS			
		1	2	3	4	5	6	7	8
PROJECT MANAGER	22,000,000	5	5	5	7	5	5	5	7
BUSINESS ANALYST	1,200,000	1	1						
TEAM MANAGER	13,200,000	5	5	5	7	5	5	5	7
Technical - Team Leader	11,000,000	5	5	5	7	5	5	5	7
System Analyst	19,800,000	5	5	5	7	5	5	5	7
Programmer	26,400,000	5	5	5	7	5	5	5	7
Technical SUPPORT	3,200,000	2	2	2	2	2	2	2	2
Technical TESTER	8,800,000	5	5	5	7	5	5	5	7
Non Technical - Team Leader	4,000,000	2	2	2	2	2	2	2	2
Support	8,800,000	5	5	5	7	5	5	5	7
Others	6,300,000	7	7	7	7	7	7		
SECURITY Specialist	10,500,000	7	7	7	7	7	7		
TOTAL BIAYA DEPLOYMENT (belum termasuk Security Spc)	124,700,000.00								

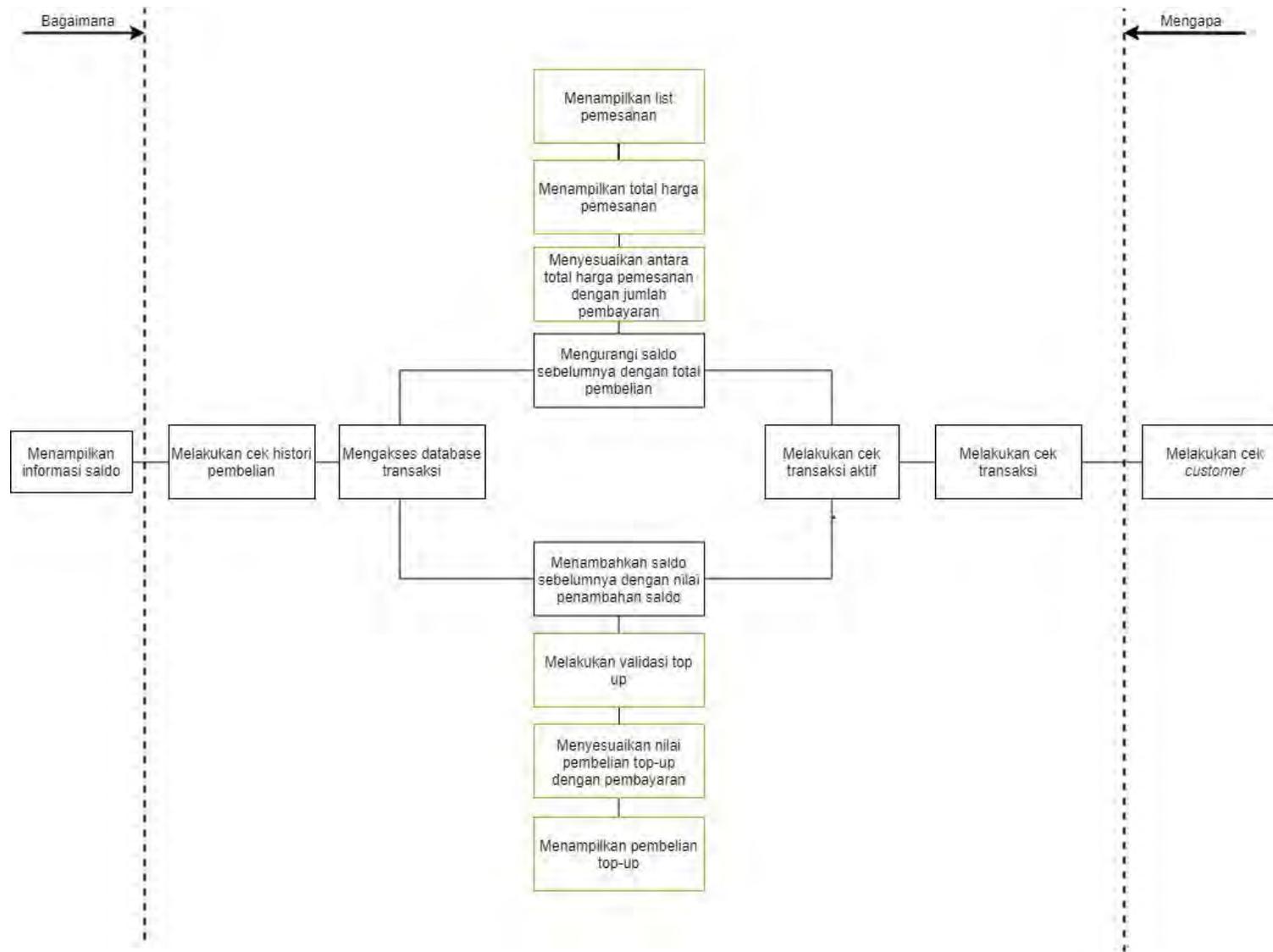
*Halaman ini sengaja dikosongkan*

## 5.2 *Function Analysis Phase*

Berdasarkan hasil perhitungan estimasi biaya per-fungsinyang akan dihitung, terdapat beberapa kebutuhan fungsional yang akan dijadikan sebagai objek penelitian. Pengolahan objek penelitian akan melibatkan pembangunan F.A.S.T Diagram untuk melihat keterkaitan antar komponen dalam sistem untuk mewujudkan fungsi tersebut. Dalam implementasinya, penulis menggunakan aplikasi draw.io sebagai *tools* dalam melakukan pembangunan F.A.S.T Diagram.

Pada contoh proyek konstruksi, objek yang diteliti adalah pintu, dan segala hal yang dimasukkan kedalam diagram adalah hal yang berkaitan erat dengan berjalannya fungsi pintu, begitu juga pada penggunaan layar proyektor, hanya memasukkan segala hal yang merupakan berkaitan erat dengan berjalannya fungsi layar proyektor. Berdasarkan contoh baru dan hasil diskusi penulis memperoleh kesimpulan bahwa perancangan FAST Diagram pada pembangunan perangkat lunak hanya akan melibatkan hal yang berkaitan erat dengan berjalannya fungsi perangkat lunak yang menjadi obkjek tersebut. Sehingga pada pembanguna FAST Diagram, penulis hanya akan memasukkan aktivitas yang sistem lakukan. Aktivitas yang dilakukan oleh *customer* dan pihak lain tidak dimasukkan kedalam FAST Diagram, hanya dijelaskan pada keterangan FAST Diagram saja. Berikut ini merupakan contoh pembangunan F.A.S.T Diagram.

*Halaman ini sengaja dikosongkan*



Gambar 5.1 Contoh F.A.S.T Diagram pada Pembangunan Fungsi Aplikasi

*Halaman ini sengaja dikosongkan*

### 5.3 Hambatan dan Rintangan

Dalam pengerjaan tugas akhir ini, penulis memiliki beberapa hambatan dalam pengerjaannya :

1. *Value engineering* merupakan topik baru pada Jurusan Sistem Informasi. Teknik ini adalah teknik yang biasa dilakukan oleh jurusan Teknik Sipil dalam membangun penghematan biaya konstruksi. Penulis mengambil teknik ini untuk diimplementasikan pada proyek pembangunan aplikasi, sehingga tidak ada acuan atau referensi yang bisa dijadikan acuan pengerjaan Tugas Akhir ini.
2. KAK (Kerangka Acuan Kerja) yang penulis dapatkan memiliki beberapa kekurangan informasi, diantaranya biaya pembangunan aplikasi tidak secara langsung didefinisikan pada KAK, melainkan didasarkan pada pemberian gaji karyawan atau pihak yang terlibat dalam pembangunan aplikasi. Sehingga penulis perlu melakukan perhitungan estimasi biaya aplikasi berdasarkan *mandays* untuk memperoleh kesimpulan apakah pembangunan aplikasi ini terbilang mahal atau tidak.
3. KAK (Kerangka Acuan Kerja) juga tidak menjabarkan pemabagian tugas dari masing-masing jabatan dan kegiatan apa saja yang dilakukan pada masing-masing aktivitas, sehingga penulis perlu mencari pembagian tugas tersebut untuk bisa menjawab kebutuhan dalam penggunaan teknik *Function point* pada pengerjaan tugas akhir ini.

Namun demikian penulis tetap terbantu dengan adanya komunikasi dengan pihak ketiga. Setiap pengerjaan atau progress yang penulis lakukan dapat langsung diperlihatkan kepada pihak ketiga untuk dilakukan koreksi dan masukan jika terdapat kekurangan atau ketidaksesuaian hasil dengan kebutuhan fungsional.

*Halaman ini sengaja dikosongkan*

## **BAB VI**

### **HASIL DAN PEMBAHASAN**

Bab ini akan menjelaskan hasil yang diperoleh pada setiap tahapan yang diimplementasikan oleh penulis dari mulai tahapan *Information Phase*, *Function Analysis Phase*, *Creativity Phase*, dan *Evaluation Phase* untuk menjawab rumusan masalah yang telah penulis jabarkan pada BAB I.

#### **6.1 Information Phase**

##### **6.1.1 Pengkajian Data Masa Lalu**

Studi kasus yang dipakai dalam penelitian ini adalah aplikasi XYZ, yang merupakan aplikasi yang bersifat *organizing* sebagai alternatif pendamping bagi para atlet pada *Event* olahraga 2018. Aplikasi ini berbasis mobile yang siap digunakan pada Android maupun iOS (Iphone). Aplikasi ini dibangun dengan didasarkan pada tujuan *payment gateway*. Berdasarkan tujuan tersebut, pihak ketiga memberikan beberapa usulan fitur dan manfaat yang bisa diperoleh untuk mewujudkan tujuan tersebut. Selain itu, pihak ketiga juga menyertakan usulan aliran informasi serta proses perjalanan fitur berhubungan dengan arsitektur aplikasinya (Kerangka Acuan Kerja. 2018).

#### **A. Fitur Aplikasi XYZ**

Berdasarkan kebutuhan fungsional usulan pihak ketiga, maka usulan tersebut dapat dirangkum kedalam beberapa kebutuhan fitur sebagai berikut.

##### **1. Fitur e-Wallet**

Fitur ini merupakan fitur yang berfungsi seperti “dompet” yang dapat menyimpan uang para atlet. Namun dalam hal ini uang tersebut bersifat elektronik. Kelebihan fitur ini adalah menyediakan fasilitas “rekening koran” yang dapat memberikan informasi saldo yang dimiliki dan rekaman transaksi yang telah dilakukan. Fitur ini akan mendukung fungsi fintech yang dapat membantu atlet dalam melakukan

transaksi dengan para penyedia layanan sesuai kebutuhan selama *Event* dilaksanakan. Adapun transaksi yang dimaksud meliputi :

- Penggunaan alat transportasi Online seperti Go-Jek/Grab-Bike/Uber, Go-car/Uber/Grab-Car/Go-Bird.
- Pembelian tiket pertandingan pada vanue tertentu. Hal ini jika atlet tersebut ingin melihat cabang olahraga lain.
- Pelaksanaan transaksi pembelian lainnya seperti pembayaran transaksi selama di hotel, pembelian barang di Merchant/*Mall* yang sedang menawarkan diskon bagi para atlet, dan lain-lain.

## 2. Fitur Informasi Akomodasi Atlet

Fitur ini memberikan informasi akomodasi bagi atlet selama *Event* dilaksanakan. Informasi pada fitur ini terkait dengan informasi diluar pertandingan. Informasi tersebut meliputi :

- Informasi pendamping atau penjemput atlet saat hendak atau sedang berpergian ketempat-tempat tertentu selama *Event* dilangsungkan.
- Informasi hotel. Baik informasi harga, fasilitas maupun lokasi hotel yang berada di daerah *Event* dilaksanakan.
- Informasi *mall* yang menjadi *official* partner.
- Informasi diskon pada produk dengan *merk* tertentu yang biasanya banyak diberikan selama *Event* olahraga diberlangsungkan.

### 3. Fitur Informasi Pertandingan

Fitur ini memberikan informasi bagi atlet selama *Event* dilaksanakan. Informasi pada fitur ini berkaitan dengan pelaksanaan pertandingan yang akan diikuti oleh masing-masing atlet. Informasi ini meliputi :

- Informasi jadwal pertandingan dan lokasi pertandingan atlet yang bersangkutan.
- Informasi LO dan penjemputan untuk pertandingan yang dilangsungkan setiap harinya
- Informasi jadwal pertandingan secara keseluruhan setiap harinya.
- Informasi pembelian tiket dan transaksi tiket tersebut.

### 4. Fitur Help-Desk

Sama seperti *help desk* pada umumnya, fitur ini membantu pengguna dalam menyampaikan keluhan atau menanyakan hal terkait dengan layanan yang disediakan pada aplikasi. Fitur ini akan menyediakan 2 tipe layanan yakni pesan online dengan fitur *chat* dan verbal atau *voice* dengan fasilitas di telepon.

### 5. Dukungan Utama lain adalah :

Beberapa fitur ini membantu terwujudnya tujuan aplikasi dan diharapkan dapat memberikan kemudahan bagi pengguna dalam melaksanakan *Event* olahraga 2018. Fitur dukungan tersebut antara lain:

- Peta (*Maps*)

Fitur ini sama seperti fitur *maps* pada umumnya yakni berfungsi untuk memberikan informasi petunjuk arah untuk melakukan perjalanan yang akan dilakukan oleh pengguna, seperti *Waze* atau *google street*. Selain

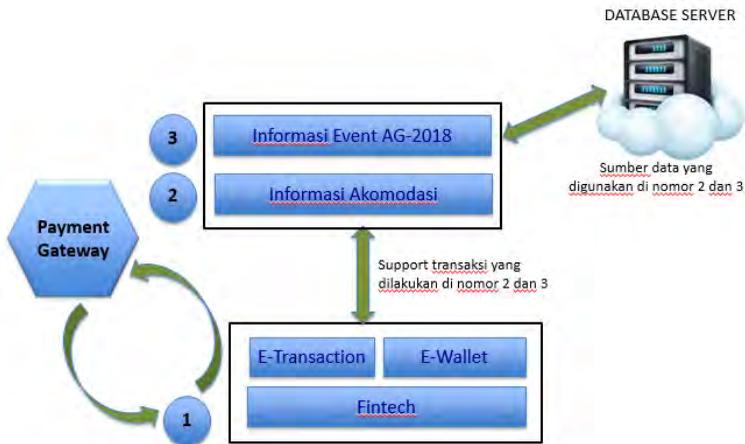
itu fitur ini juga akan memberikan informasi mengenai okasi tertentu yang dibutuhkan oleh pengguna.

- Multi bahasa.

Aplikasi ini juga akan dilengkapi dengan kemampuan multi bahasa sehingga memungkinkan untuk digunakan oleh semua atlet yang berasal dari negara yang berbeda. Namun pada aplikasi ini bahasa yang menjadi prioritas adalah bahasa Inggris dan bahasa Indonesia.

## B. Arsitektur Aplikasi XYZ

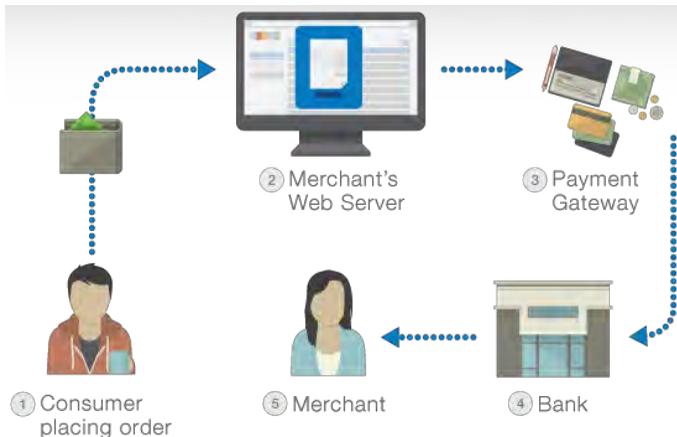
Berdasarkan fungsi utama aplikasi XYZ sebagai *payment gateway*, pihak ketiga membutuhkan database server sebagai sumber data sekaligus tempat penyimpanan setiap transaksi maupun penggunaan data pada setiap penggunaan fitur-fitur pada aplikasi seperti **Gambar 6.1**.



Gambar 6.1 Arsitektur Aplikasi XYZ

Berdasarkan arsitektur tersebut, setiap penggunaan aplikasi yang melibatkan e-wallet dan e-transaction akan memiliki hubungan timbal balik dengan aplikasi sebagai *payment gateway*. Dimana setiap transaksi akan melewati *ILFow payment gateway* agar transaksi dapat berjalan. Dan untuk keseluruhan penggunaan fitur akan melakukan hubungan timbal balik dengan database server yang artinya, database dapat memberikan data yang tersedia atau telah *diinputkan* oleh pengembang sebagai informasi untuk digunakan oleh pengguna, dan pengguna dapat memberikan data baru untuk disimpan dalam database server sebagai log atau histori.

Berdasarkan fitur yang disediakan oleh aplikasi untuk melaksanakan fungsi fintech dalam melakukan transaksi dibutuhkan kerjasama dengan beberapa penyedia layanan hingga membentuk sebuah aliran informasi seperti **Gambar 6.2**. Adapun penyedia layanan yang dibutuhkan adalah hotel, transportasi online, bank dan lain sebagainya.



Gambar 6.2 Aliran Informasi Aplikasi

Berdasarkan aliran tersebut, *customer* atau user yang ingin melakukan transaksi dapat menggunakan fitur e-wallet untuk

melakukan pemesanan melalui aplikasi. Aplikasi akan melanjutkan pemesanan secara otomatis pada website penyedia layanan. Kemudian *customer* akan melakukan pembayaran melalui aplikasi dan aplikasi akan meneruskan pembayaran kepada pihak bank dan kemudian melakukan konfirmasi pada pedagang mengenai pembelian layanan atau jasa tersebut.

### C. Proses Penggunaan Fitur

Dalam menjalankan fitur aplikasi terdapat 2 proses yang digunakan sehubungan dengan arsitektur aplikasinya.

1. Proses ini berkaitan dengan aplikasi, database server dan fungsi utama aplikasi yakni *payment gateway*. Adapun hubungan tersebut dapat dilihat pada **Gambar 6.3**.



Gambar 6.3 Proses Pelaksanaan Transaksi

Proses ini terjadi pada saat pengguna ingin melakukan registrasi atau login setiap kali ingin masuk kedalam aplikasi serta setiap kali pengguna ingin melakukan transaksi. Transaksi yang dilakukan akan memnawa aplikasi menjalankan fungsinya sebagai *payment gateway* yang akan menjembatani user dengan penyedia layanan, dimana data awal terkait layanan tersebut telah direkam kedalam database server. Hasil transaksi yang terjadi akan direkam kembali sebagai *log* dan disimpan kedalam database server.

2. Proses ini berkaitan dengan aplikasi, database server dan *cloud*. Adapun hubungan tersebut dapat dilihat pada **Gambar 6.4**.



Gambar 6.4 Proses Pencarian Informasi Umum

Proses ini akan terjadi saat pengguna ingin mengetahui informasi umum. Baik itu informasi lokasi pertandingan, jadwal maupun lokasi tempat tertentu yang ingin dituju. Menggunakan aplikasi ini, pengguna dapat mencari informasi yang diinginkan, dan akan secara langsung terhubung keinformasi terkait apa yang dicari tersebut dengan memanggil informasi yang direkam dalam database server. Dan sama seperti sebelumnya, aktivitas user akan terekam dan menjadi data *log* pada database server.

#### D. Tampilan Aplikasi yang Diharapkan

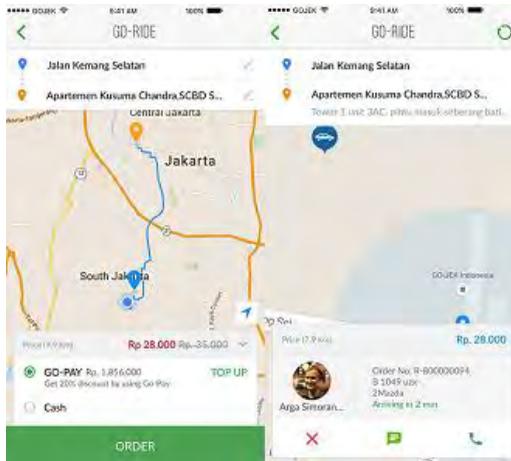
Pembangunan aplikasi ini terinspirasi dari salah satu aplikasi transportasi online yakni Go-Jek. Tampilan pada aplikasi ini akan mengikuti beberapa desain yang sama seperti pada aplikasi Go-Jek. Desain tampilan yang akan dibuat terinspirasi dari Go-Jek dapat dilihat pada **Gambar 6.5**.



Gambar (a)



Gambar (b)



Gambar (c)

Gambar 6.5 Gambar (a), (b) dan (c) Gambar Tampilan Aplikasi yang Diharapkan

Berdasarkan **Gambar (a)** pengembang ingin membangun aplikasi yang memiliki tampilan dengan model iconic seperti pada Go-Jek. Berdasarkan **Gambar (b)** pengembang ingin membangun aplikasi yang memiliki fitur cek saldo dan memiliki dapat ditampilkan secara TOP UP. Berdasarkan **Gambar (c)** pengembang ingin terdapat tampilan peta info lokasi seperti pada gojek serta peta panduan perjalanan (rute). Namun untuk peta sendiri terdapat 2 opsi, yakni menggunakan yang sudah ada pada *google* atau membuat sendiri.

### 6.1.2 Perhitungan Estimasi Biaya dengan *Function point*

Estimasi harga yang diberikan oleh pihak ketiga didasarkan pada besarnya gaji sumber daya manusia sesuai dengan jabatan. Namun dalam penyelesaian tugas akhir ini, penulis memerlukan perhitungan estimasi biaya pembangunan aplikasi berdasarkan fitur yang ingin dibangun, sehingga penulis menggunakan metode *Function point* untuk mengkonversi *effort* kedalam biaya. Adapun tahap awal yang dilakukan oleh penulis adalah pembuatan Data Flow Diagram (DFD) untuk mempermudah penulis menterjemahkan kebutuhan *eksternal input*, *eksternal Output*, *eksternal inquiry*, *file logic* dan *interface eksternal* untuk keperluan perhitungan CFP (Lampiran A).

Berdasarkan DFD tersebut, aliran data adalah sebagai berikut.

1. Jika *customer* ingin melakukan pendaftaran, *customer* akan *menginputkan* informasi yang diminta saat pendaftaran (data akun) kemudian masuk kedalam database *customer*, selanjutnya akan di cek oleh admin dan divalidasi kedalam database.
2. Untuk informasi lokasi, *customer* akan berhubungan langsung dengan *google maps*. *Customer* hanya perlu memasukkan kata kunci lokasi untuk mengetahui keberadaan lokasi.
3. Centre aplikasi berada pada admin, sehingga untuk memasukkan informasi berupa informasi pertandingan, LO, akomodasi, transportasi, *mall*, produk penyedia layanan dan lain-lain dilakukan oleh admin.

4. Untuk *input* informasi terkait keberlangsungan olahraga, pihak *Event* olahraga akan berhubungan dengan admin, pihak *Event* Olahraga akan memberikan informasi kepada admin untuk diinputkan kedalam database *Event* olahraga untuk selanjutnya dapat dilihat oleh *Customer*.
5. Dalam melakukan transaksi, *customer* akan berhubungan dengan pihak Penyedia Jasa dan Bank. Dalam melakukan pembelian, *customer* akan memilih layanan yang akan dibeli, kemudian dibayarkan kepada pihak bank, dan akan divalidasi oleh pihak penyedia jasa, dan tercatat kedalam database transaksi. *Customer* dapat mencari tahu histori transaksinya dengan menggunakan fitur yang akan berhubungan dengan database transaksi tersebut.
6. Sehubungan dengan promo, *customer* dapat menggunakan fitur yang akan berhubungan dengan database promo.
7. Dalam melakukan setiap aktivitas transaksi, *customer* dapat melakukan top up saldo, admin akan melakukan validasi top up yang dilakukan dengan pihak bank dan mengupdate nya kedalam database transaksi, untuk dapat dilihat oleh *customer*.
8. Dalam menyampaikan keluhan, *customer* akan berhubungan dengan *help desk*. *Customer* dapat menyampaikan keluhan secara langsung kepada pihak *help desk* melalui telp atau sms untuk ditanggapi oleh pihak *Help desk*.

Selanjutnya penulis akan memulai perhitungan estimasi biaya dengan menggunakan metode *Function point*. Perhitungan tersebut terdiri atas 5 langkah berikut.

### **Langkah 1. Menghitung Crude *Function point* (CFP)**

Dari gambar diagram arus tersebut hasil perhitungan nilai yang diperlukan untuk menghitung CFP, yakni jumlah masing-masing komponen dari 5 hal berikut ini.

- Jumlah eksternal *input* : 10
- Jumlah eksternal *Output* : 5
- Jumlah eksternal inquiry : 15
- Jumlah file logic : 5
- Jumlah interface eksternal : 0

Masing-masing komponen tersebut kemudian dikelompokkan kedalam 3 jenis pembobotan dengan hasil seperti pada **Tabel 6.1** berikut ini.

Tabel 6.1 Hasil Pengelompokan *Inputan*

<b>Hasil Pengelompokan <i>Inputan</i></b>	
<b>Nama</b>	<b>Jenis</b>
<b>Eksternal <i>Input</i></b>	
Informasi <i>mall</i>	S
Informasi produk penyedia layanan dan jasa	C
Informasi Pertandingan	S
Informasi LO	S
Informasi Akomodasi	S
Informasi Transportasi	A
Kata kunci lokasi	C
Data akun	S
Transaksi pembelian	C
Transaksi	C
<b>Eksternal <i>Output</i></b>	
Promo	S
Rekening koran	C
Histori pembelian	C
Informasi Lokasi	C
Profil <i>customer</i>	A
<b>Eksternal inquiry</b>	
Penyampaian informasi <i>Event</i>	C
<i>Input Event</i>	S
Cek <i>customer</i>	S
Pendaftaran	A
Akses peta	C

<b>Hasil Pengelompokan <i>Inputan</i></b>	
<b>Nama</b>	<b>Jenis</b>
Melihat even	S
Penyampaian Keluhan	S
Top up saldo	A
Pembelian	C
Cek promo	S
Cek transaksi	S
Pembayaran	C
Validasi top up	A
Validasi penjualan	A
Penjualan	C
<b>Internal Logic File</b>	
<i>Event</i> Olahraga	S
Promo Penyedia Jasa	A
Transaksi	C
<i>Customer</i>	S
Database Bahasa	A
<b>Eksternal Interface Files</b>	

Setelah memperoleh jumlah dari masing-masing kategori, selanjutnya dilakukan perhitungan dengan menggunakan CFP berdasarkan usulan fitur pihak ketiga dengan hasil seperti pada **Tabel 6.2**.

Tabel 6.2 Hasil Perhitungan CFP

<b>Tipe Fungsi</b>	<b>Bobot Kompleksitas</b>								
	<b>Simple</b>			<b>Average</b>			<b>Complex</b>		
	Count	Bobot	Point	Count	Bobot	Point	Count	Bobot	Point
External <i>Inputs</i>	5	3	<b>15</b>	1	4	<b>4</b>	4	6	<b>24</b>
External <i>Output</i>	1	4	<b>4</b>	1	5	<b>5</b>	3	7	<b>21</b>

Tipe Fungsi	Bobot Kompleksitas								
	Simple			Average			Complex		
	Count	Bobot	Point	Count	Bobot	Point	Count	Bobot	Point
Internal File Logic	2	7	14	2	10	20	1	15	15
External Interface Files	0	5	0	0	7	0	0	10	0
External Inquiry	6	3	9	4	4	16	5	6	30
<b>TOTAL</b>									<b>177</b>

Berdasarkan hasil perhitungan, diperoleh nilai CFP adalah 177.

## Langkah 2. Menghitung Relative Complexity Adjustment Factor (RCAF)

Hasil perhitungan RCAF dimasukkan langsung kedalam tabel sesuai dengan karakteristik yang dimiliki aplikasi. Adapun kriteria tersebut dijelaskan pada **BAB IV**. Pemberian nilai bobot kepentingan dengan besar antara 0 sampai 5 untuk masing-masing factor mengacu pada dokumen *Standar Documentation* untuk *General System Characteristics (GSC)* yang dipublikasikan oleh *Software Matric*. Untuk masing-masing factor, bobot nilai dijelaskan dalam deskripsi untuk menyamakan standar pemberian bobot nilai dan memudahkan untuk digunakan dalam aplikasi berbasis GUI. Hasil yang diperoleh berdasarkan fitur usulan pihak ketiga seperti pada **Tabel 6.3**.

Tabel 6.3 Hasil Perhitungan RCAF

No	Karakteristik	Bobot
1.	Tingkat kompleksitas Komunikasi Data	[0/1/2/3/4/5]

No	Karakteristik	Bobot
2.	Tingkat kompleksitas Pemrosesan Terdistribusi	[0/1/2/3/4/5]
3.	Tingkat kompleksitas Performance	[0/1/2/3/4/5]
4.	Tingkat kompleksitas konfigurasi	[0/1/2/3/4/5]
5.	Tingkat frekuensi penggunaan <i>software</i>	[0/1/2/3/4/5]
6.	Tingkat frekuensi <i>Input Data</i>	[0/1/2/3/4/5]
7.	Tingkat Kemudahan penggunaan bagi <i>user</i>	[0/1/2/3/4/5]
8.	Tingkat frekuensi <i>update data</i>	[0/1/2/3/4/5]
9.	Tingkat kompleksitas Prossessing Data	[0/1/2/3/4/5]
10.	Tingkat kemungkinan penggunaan kembali	[0/1/2/3/4/5]
11.	Tingkat kemudahan dalam instalasi	[0/1/2/3/4/5]
12.	Tingkat kemudahan operasional <i>software (backup, recovery, dsb)</i>	[0/1/2/3/4/5]
13.	Tingkat <i>software</i> dibuat untuk multi organisasi/perusahaan/cIEIFnt	[0/1/2/3/4/5]
14.	Tingkat kompleksitas dalam mengikuti perubahan/ILFeksibel	[0/1/2/3/4/5]
<b>Total</b>		<b>46</b>

Berdasarkan hasil perhitungan, nilai RCAF adalah 46.

### Langkah 3. Menghitung *Function point* (FP)

Perhitungan *Function point* dilakukan dengan memasukkan nilai kedalam persamaan yang sudah dijelaskan pada **BAB IV**. Hasil perhitungan tersebut adalah sebagai berikut.

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times RCAF) \\
 &= 177 \times (0.65 + (0.01 \times 46)) \\
 &= 177 \times (0,65 + 0,46) \\
 &= 177 \times 1,11 \\
 &= 196,47
 \end{aligned}$$

#### Langkah 4. Estimasi Effort

Mengetahui estimasi *effort* maka perlu menggunakan fungsi estimasi eksponensial oleh *Jone's*

Kind of Software	Organization's Skill/Abilities		
	Best In Class	Average	Worst In Class
System	0.43	0.45	0.48
Business	0.41	0.43	0.46
Shrink-wrap	0.39	0.42	0.45

Gambar 6.6 Estimasi Eksponensial oleh *Jone's*

$$\begin{aligned}
 \text{Effort} &= FP^{3*j} / 27 \\
 &= 196,47^{3*0.41} / 27 \\
 &= 196,47^{1,23} / 27 \\
 &= 661,84 / 27 \\
 &= 24 \text{ orang}
 \end{aligned}$$

$$\begin{aligned}
 S &= f^j \\
 &= 196,47^{0,41} \\
 &= 8.7 \text{ bulan} \\
 &= 174 \text{ hari (20 hari kerja per-bulan)}
 \end{aligned}$$

#### Langkah 5. Estimasi Biaya Total

Setelah diperoleh estimasi sumber daya manusia dan hari yang efektif dalam penyelesaian pembangunan aplikasi tersebut, selanjutnya dilakukan pembagian sumber daya manusia dan hari disesuaikan dengan jabatan yang bertanggungjawab untuk dikalikan sehingga memperoleh total estimasi biaya. Hasil perhitungan seperti pada **Tabel 6.4**.

Tabel 6.4 Hasil Perhitungan Estimasi Biaya Total

<b>Kelompok aktivitas</b>	<b>Jabatan</b>	<b>Person</b>	<b>Days</b>	<b>Standar Gaji (per-hari)</b>	<b>Total</b>
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	2	12	Rp 350.000	Rp 8.400.000
Spesification & Design	<i>System analyst</i>	2	26	Rp 350.000	Rp 18.200.000
Coding	<i>Programmer</i>	2	15	Rp 400.000	Rp 12.000.000
Testing	<i>Technical tester</i>	2	10	Rp 400.000	Rp 8.000.000
Acceptance and Deployment	<i>Technical Team leader</i>	2	14	Rp 750.000	Rp 21.000.000
<b>Ongoing Activity</b>					
Project management	<i>Project manger</i>	2	14	Rp 1.000.000	Rp 28.000.000
Configuration Management	<i>Technical Team leader</i>	1	9	Rp 750.000	Rp 6.750.000
Documentation	<i>Non Technical Team leader</i>	1	9	Rp 750.000	Rp 6.750.000
Training and Support	<i>Technical support</i>	2	7	Rp 400.000	Rp 5.600.000
<b>Quality and Testing</b>					
Integration Testing	<i>Technical tester</i>	2	14	Rp 400.000	Rp 11.200.000
Quality Assurance and Control	<i>Project manger</i>	2	14	Rp 1.000.000	Rp 28.000.000
Evaluation and Testing	<i>Technical tester</i>	4	30	Rp 400.000	Rp 48.000.000
<b>TOTAL</b>					<b>Rp 201.900.000</b>

Berdasarkan perhitungan menggunakan metode *Function point*, estimasi biaya pembangunan aplikasi adalah sebesar Rp 201.900.000.

## Langkah 6. Konversi Hasil *Function point* dengan Estimasi Pihak ketiga.

### A. Memanfaatkan Waktu Lembur

Jika dibandingkan dengan estimasi biaya yang terdapat pada lampiran, pihak ketiga memberikan harga pembangunan aplikasi sebesar Rp 439.300.000 dengan estimasi waktu 5 bulan, dengan estimasi hari kerja adalah 20 hari per-bulan setara dengan 100 hari dan estimasi jam kerja per-hari adalah 8 jam atau setara dengan 800 jam . Untung menentukan apakah biaya tersebut mahal atau tidak, kita perlu mengkonversi harga *Function point* dari 8.7 bulan atau 174 hari ke dalam kurun waktu 5 bulan atau 100 hari, atau pekerja harus melakukan lembur sebanyak 74 hari kerja atau setara dengan 592 jam. Untuk melakukan perhitungan, perlu dilakukan perhitungan pembagian hari kerja dan cara perhitungan upah lembur pada hari kerja.

Sebelum menghitung waktu kerja, perlu dilakukan perubahan jam kerja biasa menjadi 5 bulan. Pembagian jam kerja dihitung dengan menggunakan persamaan sebagai berikut.

$$\text{Waktu Kerja} = \frac{\text{Jam kerja Awal}}{\text{Total Jam kerja awal}} \times \text{Total Jam Kerja Baru}$$

Contoh perhitungannya adalah sebagai berikut

Pada aktivitas requirement, *Business analyst* memiliki jam kerja awal adalah 12 hari untuk 8.7 bulan. Jika ingin dikonversi kedalam 5 bulan, maka :

$$\begin{aligned} \text{Waktu Kerja} &= \frac{96}{1392} \times 800 \\ &= 56 \text{ Jam} \end{aligned}$$

Perubahan hari kerja secara keseluruhan dapat dilihat pada **Tabel 6.5** sebagai berikut.

Tabel 6.5 Perhitungan Hari Kerja Biasa dalam 5 Bulan

PERUBAHAN HARI KERJA DALAM 5 BULAN										
Aktivitas	Jabatan	Jam Kerja / hari	Hari kerja 8.7 bulan	Total jam kerja 8.7 bulan	Total jam kerja 5 bulan	Bobot	Pembulatan (jam)	Hari kerja (hari)	Pembulatan (hari)	Perubahan dalam Jam
<b>Software Development</b>										
Requirement	<i>Business analyst</i>	8	96	1392	800	55.17	55	6,875	7	56
Spesification & Design	<i>System analyst</i>	8	208	1392	800	119.54	120	15	15	120
Coding	<i>Programmer</i>	8	120	1392	800	68.97	69	8,625	9	72
Testing	<i>Technical tester</i>	8	80	1392	800	45.98	46	5,75	6	48

PERUBAHAN HARI KERJA DALAM 5 BULAN										
Aktivitas	Jabatan	Jam Kerja / hari	Hari kerja 8.7 bulan	Total jam kerja 8.7 bulan	Total jam kerja 5 bulan	Bobot	Pembulatan (jam)	Hari kerja (hari)	Pembulatan (hari)	Perubahan dalam Jam
Acceptance and Deployment	<i>Technical Team leader</i>	8	112	1392	800	64.37	64	8	8	64
<b>Ongoing Activity</b>										
Project management	<i>Project manger</i>	8	112	1392	800	64.37	64	8	8	64
Configuration Management	<i>Technical Team leader</i>	8	56	1392	800	32.18	32	4	5	40

PERUBAHAN HARI KERJA DALAM 5 BULAN										
Aktivitas	Jabatan	Jam Kerja / hari	Hari kerja 8.7 bulan	Total jam kerja 8.7 bulan	Total jam kerja 5 bulan	Bobot	Pembulatan (jam)	Hari kerja (hari)	Pembulatan (hari)	Perubahan dalam Jam
Documentation	Non Technical Team leader	8	56	1392	800	32.18	32	4	5	40
Training and Support	Technical support	8	56	1392	800	32.18	32	4	4	32
Quality and Testing										
Integration Testing	Technical tester	8	112	1392	800	64.37	64	8	8	64
Quality Assurance and Control	Project manager	8	112	1392	800	64.37	64	8	8	64

<b>PERUBAHAN HARI KERJA DALAM 5 BULAN</b>										
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jam Kerja / hari</b>	<b>Hari kerja 8.7 bulan</b>	<b>Total jam kerja 8.7 bulan</b>	<b>Total jam kerja 5 bulan</b>	<b>Bobot</b>	<b>Pembulatan (jam)</b>	<b>Hari kerja (hari)</b>	<b>Pembulatan (hari)</b>	<b>Perubahan dalam Jam</b>
<i>Evaluation and Testing</i>	<i>Technical tester</i>	8	240	1392	800	137.93	138	17,25	17	136
<b>TOTAL</b>						<b>781,61</b>	<b>780</b>	<b>97.5</b>	<b>100</b>	<b>800</b>

Berdasarkan perubahan kerja pada **Tabel 6.5** dilakukan perhitungan estimasi biaya kerja biasa menjadi seperti pada **Tabel 6.6**.

Tabel 6.6 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 5 Bulan

Kelompok aktivitas	Jabatan	Person	Hours	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	2	56	Rp 43.750	Rp 4.900.000
Spesification & Design	<i>System analyst</i>	2	120	Rp 43.750	Rp 10.500.000
Coding	<i>Programmer</i>	2	72	Rp 50.000	Rp 7.200.000
Testing	<i>Technical tester</i>	2	48	Rp 50.000	Rp 4.800.000
Acceptance and Deployment	<i>Technical Team leader</i>	2	64	Rp 93.750	Rp 12.000.000
<b>Ongoing Activity</b>					
Project management	<i>Project manger</i>	2	64	Rp 125.000	Rp 16.000.000
Configuration Management	<i>Technical Team leader</i>	1	40	Rp 93.750	Rp 3.750.000
Documentation	Non <i>Technical Team leader</i>	1	40	Rp 93.750	Rp 3.750.000
Training and Support	<i>Technical support</i>	2	32	Rp 50.000	Rp 1.600.000
<b>Quality and Testing</b>					
Integration Testing	<i>Technical tester</i>	2	64	Rp 50.000	Rp 6.400.000

Kelompok aktivitas	Jabatan	Person	Hours	Standar Gaji (per-hari)	Total
Quality Assurance and Control	<i>Project manger</i>	2	64	Rp 125.000	Rp 16.000.000
<i>Evaluation and Testing</i>	<i>Technical tester</i>	4	136	Rp 50.000	Rp 27.200.000
<b>TOTAL</b>					<b>Rp 115.700.000</b>

Selanjutnya dilakukan perhitungan biaya dengan memanfaatkan waktu lembur. Hari kerja biasa yang dimanfaatkan adalah 800 jam, sehingga diperlukan pemanfaatan hari kerja selama 592 jam. Secara lebih mudah perubahan hari kerja dapat dilihat pada **Tabel 6.7**.

Tabel 6.7 Perubahan Hari Kerja

Keterangan	Bulan	Hari	Jam
Estimasi <i>Function point</i>	8.7 Bulan	174 Hari	1392 Jam
Estimasi Pihak Ketiga	5 Bulan	100 Hari	800 Jam
Lembur	3.7 Bulan	74 Hari	592 Jam

Dalam perhitungan pembagian hari lembur, bobot masing-masing sumber daya tidak sama. Sumber daya manusia dengan hari kerja yang lebih banyak akan memiliki waktu lembur yang lebih lama. Untuk menghitung waktu lembur masing-masing sumber daya manusia menggunakan persamaan berikut :

Lembur = Total Jam Kerja 8.7 Bulan – Jam Kerja Biasa di 5 Bulan

Contoh perhitungannya adalah sebagai berikut :

Pada aktivitas requirement, *Business analyst* memiliki jam kerja awal adalah 96 hari dalam total pengerjaan selama 8.7 bulan dan perhitungan setelah dilakukan perubahan menjadi 56 hari kerja biasa. Jika ingin menghitung waktu lembur, maka :

$$\begin{aligned} \text{Lembur} &= 96 - 56 \\ &= 40 \text{ Hari} \end{aligned}$$

Pemanfaatan waktu lembur secara keseluruhan dapat dilihat pada **Tabel 6.8** berikut.

Tabel 6.8 Perhitungan Jam Lembur

<b>PERHITUNGAN HARI LEMBUR</b>		
<b>Jam kerja di 8.7 bulan</b>	<b>Perubahan Jam Kerja ke 5 Bulan</b>	<b>Jam Lembur 5 Bulan</b>
96	56	40
208	120	88
120	72	48
80	48	32
112	64	48
112	64	48
56	40	32
56	40	32
56	32	24
112	64	48
112	64	48
240	136	104
<b>TOTAL</b>		<b>592</b>

Berdasarkan perubahan kerja pada **Tabel 6.8** dilakukan perhitungan estimasi biaya kerja biasa menjadi seperti pada **Tabel 6.9**.

Contoh perhitungan :

Pada aktivitas requirement, *Business analyst* memiliki jam lembur 5 hari, dimana perharinya terhitung 8 jam, sehingga perhitungan biaya lembur pada hari pertama adalah sebagai berikut.

$$\begin{aligned} &= 1 \times 1,5 \times \frac{1}{173} 7.000.000 \\ &= 121.387 \end{aligned}$$

Perhitungan pada hari berikut mengikuti persamaan kedua menjadi berikut ini :

$$\begin{aligned} &= 39 \times 2 \times \frac{1}{173} 7.000.000 \\ &= 6.312.139 \end{aligned}$$

Perubahan hari kerja secara keseluruhan dapat dilihat pada **Tabel 6.9** sebagai berikut.

Tabel 6.9 Perhitungan Estimasi Biaya Lembur selama 5 Bulan

Kelompok aktivitas	Jabatan	Person (Orang)	Hours (Jam)	Standar Gaji (per-jam)	Upah/bulan (Rp)	Gaji lembur jam pertama (Rp)	Gaji lembur jam selanjutnya (Rp)
<b>Software Development</b>							
Requirement	<i>Business analyst</i>	2	40	Rp 43750	7.000.000	Rp 121.387	Rp 6.312.139
Spesification & Design	<i>System analyst</i>	2	88	Rp 43750	7.000.000	Rp 121.387	Rp 14.080.924
Coding	<i>Programmer</i>	2	48	Rp 50000	8.000.000	Rp 138.728	Rp 8.693.642
Testing	<i>Technical tester</i>	2	32	Rp 50000	8.000.000	Rp 138.728	Rp 5.734.104
Acceptance and Deployment	<i>Technical Team leader</i>	2	48	Rp 93750	15.000.000	Rp 260.116	Rp 16.300.578
<b>Ongoing Activity</b>							
Project management	<i>Project manger</i>	2	48	Rp 125000	20.000.000	Rp 346.821	Rp 21.734.104

<b>Kelompok aktivitas</b>	<b>Jabatan</b>	<b>Person (Orang)</b>	<b>Hours (Jam)</b>	<b>Standar Gaji (per-jam)</b>	<b>Upah/bulan (Rp)</b>	<b>Gaji lembur jam pertama (Rp)</b>	<b>Gaji lembur jam selanjutnya (Rp)</b>
Configuration Management	<i>Technical Team leader</i>	1	24	Rp 93750	15.000.000	Rp 130.058	Rp 5.375.722
Documentation	<i>Non Technical Team leader</i>	1	24	Rp 93750	15.000.000	Rp 130.058	Rp 5.375.722
Training and Support	<i>Technical support</i>	1	24	Rp 50000	8.000.000	Rp 138.728	Rp 4.254.335
Integration Testing	<i>Technical tester</i>	2	48	Rp 50000	8.000.000	Rp 138.728	Rp 8.693.642
Quality Assurance and Control	<i>Project manger</i>	2	48	Rp 125000	20.000.000	Rp 346.821	Rp 21.734.104
<i>Evaluation and Testing</i>	<i>Technical tester</i>	4	88	Rp 50000	8.000.000	Rp 277.457	Rp 38.104.046
<b>TOTAL</b>						<b>Rp 2.289.017</b>	<b>Rp 156.393.064</b>

Berdasarkan hasil perhitungan konversi hari kerja, biaya pengerjaan aplikasi ini adalah dengan menjumlahkan nilai estimasi biaya + biaya lembur hari kerja. Secara numerik sebagai berikut.

Biaya pembangunan aplikasi :

$$= \text{Rp } 115.700.000 + \text{Rp } 2.289.017 + \text{Rp } 156.393.064$$

$$= \text{Rp } 170.252.081$$

Berdasarkan hasil perhitungan *Function point* dan konversi waktu pengerjaan diperoleh bahwa harga pembangunan aplikasi adalah Rp 170.252.081.

## B. Penambahan SDM

Selain melakukan konversi dengan waktu lembur, penulis juga melakukan perhitungan dengan menambah jumlah SDM untuk memenuhi permintaan pembangunan aplikasi selama 5 bulan atau setara dengan 100 hari. Perubahan penambahan SDM tersebut dapat dilihat dari **Tabel 6.10**.

Tabel 6.10 Penambahan Jumlah SDM

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
Aktivitas	Jabatan	BEFORE		AFTER	
		Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	2	12	4	6
Spesification & Design	<i>System analyst</i>	2	26	4	13
Coding	<i>Programmer</i>	2	15	4	8
Testing	<i>Technical tester</i>	2	10	4	5

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
Aktivitas	Jabatan	BEFORE		AFTER	
		Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
Acceptance and Deployment	<i>Technical Team leader</i>	2	14	4	7
<b>Ongoing Activity</b>					
Project management	<i>Project manger</i>	2	14	3	8
Configuration Management	<i>Technical Team leader</i>	1	9	2	4
Documentation	<i>Non Technical Team leader</i>	1	9	1	9
Training and Support	<i>Technical support</i>	2	7	3	5
<b>Quality and Testing</b>					
Integration Testing	<i>Technical tester</i>	2	14	4	7
Quality Assurance and Control	<i>Project manger</i>	2	14	3	8
Evaluation and Testing	<i>Technical tester</i>	4	30	6	20
<b>TOTAL</b>		<b>174</b>		<b>100</b>	

Berdasarkan perubahan jumlah SDM tersebut, maka estimasi biaya pembangunan aplikasi menjadi seperti pada **Tabel 6.11**.

Tabel 6.11 Perhitungan Estimasi Biaya dengan Penambahan SDM

PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM					
Aktivitas	Jabatan	Jumlah pekerja (Orang)	Hari Kerja (Hari)	Standar gaji/hari (Rp)	Hasil (Rp)
<i>Software Development</i>					

PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM					
Aktivitas	Jabatan	Jumlah pekerja (Orang)	Hari Kerja (Hari)	Standar gaji/hari (Rp)	Hasil (Rp)
Requirement	<i>Business analyst</i>	4	6	Rp 350.000	Rp 8.400.000
Spesification & Design	<i>System analyst</i>	4	13	Rp 350.000	Rp 18.200.000
Coding	<i>Programmer</i>	4	8	Rp 400.000	Rp 12.800.000
Testing	<i>Technical tester</i>	4	5	Rp 400.000	Rp 8.000.000
Acceptance and Deployment	<i>Technical Team leader</i>	4	7	Rp 750.000	Rp 21.000.000
Ongoing Activity					
Project management	<i>Project manger</i>	3	8	Rp 1.000.000	Rp 24.000.000
Configuration Management	<i>Technical Team leader</i>	2	4	Rp 750.000	Rp 6.000.000
Documentation	<i>Non Technical Team leader</i>	1	9	Rp 750.000	Rp 6.750.000
Training and Support	<i>Technical support</i>	3	5	Rp 400.000	Rp 6.000.000
Quality and Testing					
Integration Testing	<i>Technical tester</i>	4	7	Rp 400.000	Rp 11.200.000
Quality Assuance and Control	<i>Project manger</i>	3	8	Rp 1.000.000	Rp 24.000.000
<i>Evaluation and Testing</i>	<i>Technical tester</i>	6	20	Rp 400.000	Rp 48.000.000
<b>TOTAL</b>					<b>Rp 194.350.000</b>

Total Pembangunan aplikasi dengan menambahkan jumlah SDM yang terlibat adalah senilai Rp 194.350.000.

### C. Pemanfaatan Visual Programming

Skenario ketiga adalah pemanfaatan visual programming. Dengan adanya visual programming, waktu pengerjaan yang dipegang oleh *programmer* dapat dikurangi. Hasil yang diperoleh seperti pada **Tabel 6.12**.

Tabel 6.12 Penggunaan Visual Programming (Informasi Saldo)

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	2	12	Rp 350.000	Rp 8.400.000
Spesification & Design	<i>System analyst</i>	2	26	Rp 350.000	Rp 18.200.000
Coding	<i>Programmer</i>	1	11	Rp 400.000	Rp 4.400.000
Testing	<i>Technical tester</i>	2	10	Rp 400.000	Rp 8.000.000
Acceptance and Deployment	<i>Technical Team leader</i>	2	14	Rp 750.000	Rp 21.000.000
<b>Ongoing Activity</b>					
Project management	<i>Project manger</i>	2	14	Rp 1.000.000	Rp 28.000.000
Configuration Management	<i>Technical Team leader</i>	1	9	Rp 750.000	Rp 6.750.000
Documentation	<i>Non Technical Team leader</i>	1	9	Rp 750.000	Rp 6.750.000
Training and Support	<i>Technical support</i>	2	7	Rp 400.000	Rp 5.600.000
<b>Quality and Testing</b>					
Integration Testing	<i>Technical tester</i>	2	14	Rp 400.000	Rp 11.200.000

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
Quality Assurance and Control	<i>Project manger</i>	2	14	Rp 1.000.000	Rp 28.000.000
<i>Evaluation and Testing</i>	<i>Technical tester</i>	4	30	Rp 400.000	Rp 48.000.000
<b>TOTAL</b>					<b>Rp 194.300.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 194.300.000, lebih murah dari harga awal.

### 6.1.3 Perbandingan Estimasi Biaya dengan Penyedia Jasa

Berdasarkan estimasi biaya yang diberikan oleh pihak ketiga, penulis melakukan perbandingan harga dengan beberapa penyedia jasa untuk mencari tahu apakah estimasi harga oleh pihak ketiga terbilang mahal atau tidak. Penulis melakukan *review* pada beberapa website penyedia jasa berikut, serta memberikan pertanyaan melalui *e-mail*, langsung kepada programmer dari masing-masing penyedia jasa tentang biaya pembanguna aplikasi dengan fungsi yang sesuai dengan usulan pihak ketiga yang telah dijelaskan pada penjelasan studi kasus. Sebagai pembanding, penulis menggunakan beberapa pembanding berupa aplikasi transportasi online dan marketplace. Hasil yang diperoleh dapat dilihat dalam **Tabel 6.13**.

Tabel 6.13 Hasil Perbandingan Aplikasi

No	Fitur yang akan diandingkan	App A (topbisnis.org)	App B (jagowebdesain.com)
1.	Login	✓	✓

No	Fitur yang akan diandingkan	App A (topbisnis.org)	App B (jagowebdesain.com)
2.	Aplikasi dapat menampilkannya informasi saldo	✓	✓
3.	Aplikasi dapat menampilkan aktivitas transaksi yang telah dilakukan <i>customer</i>	✓	✓
4.	Aplikasi dapat melakukan pemesanan transportasi online, tiket vanue tertentu dan transaksi lain seperti hotel, pembelian barang, dan lain-lain	✓	✓
5.	Aplikasi dapat menampilkan informasi pendamping dan penjemput atlet	✓	✓
6.	Aplikasi dapat menampilkan informasi hotel yang berada disekitar lokasi <i>Event</i>	✓	✓
7.	Aplikasi dapat menampilkan informasi <i>mall</i> di sekitar lokasi pelaksanaan <i>Event</i> yang menjadi <i>official partner</i> dan diskon produk tertentu yang ada selama <i>Event</i> berlangsung	✓	✓
8.	Aplikasi dapat menampilkan infromasi jadwal	✓	✓

No	Fitur yang akan diandingkan	App A (topbisnis.org)	App B (jagowebdesain.com)
	dan lokasi pertandingan		
9.	Aplikasi dapat dimanfaatkan untuk menyampaikan keluhan atau menanyakan hal terkait layanan yang tersedia pada aplikasi	✓	✓
10.	Aplikasi dapat memberikan petunjuk lokasi suatu	✓	✓
11.	Aplikasi dapat ditampilkan dalam berbagai bahasa, minima dua bahasa yakni Bahasa Inggris dan Bahasa Indonesia	✓	✓
<b>Total Harga</b>		<b>Rp 149.99 JT</b>	<b>Rp 150 JT</b>

### A. Aplikasi A

Aplikasi ini merupakan aplikasi yang ditawarkan oleh penyedia jasa di [www.topbisnis.org](http://www.topbisnis.org). Aplikasi ini merupakan aplikasi multi device yang dibangun secara profesional dengan tujuan bisnis. Website penyedia jasa ini memiliki copyright 2014-2018. Kelebihan penyedia jasa ini adalah memberikan satu paketan khusus bagi para pebisnis profesional dengan detail aplikasi *multi device* dan beberapa detail lain seperti *security*, *optimasi* dan lain-lain. Pembangunan aplikasi ini ditawarkan dengan harga Rp. 149.990.000. Adapun detail yang diperoleh adalah sebagai berikut (TopBisnis.2018.)

### ➤ **Detail spesifikasi Ojek Online**

Aplikasi ini menyediakan paketan aplikasi dengan ketentuan sebagai berikut :

- **Fitur : Unlimited**  
Yang artinya kita dapat meminta fitur dengan fungsi apa saja untuk dimasukkan kedalam aplikasi, namun perihal pengerjaannya, kita serahkan sepenuhnya kepada pihak penyedia jasa.
- **Design : Premium + Profesional**  
Desain yang disediakan berupa desain icon, tampilan untuk aplikasi dan website itu sendiri.
- **Pembuatan aplikasi dalam bentuk website**
- **Custom : High Quality**
- **Multi device**  
Artinya dapat digunakan pada android, iphone serta windows phone.
- **Memberikan tambahan dukungan berupa *maintanance*, optimasi, *security* profesional, *support* selama 15 hari, garansi 100% untuk *custom*, dan dapat melakukan *upload* ke android, iphone secara *limited*, dan *device* jenis lain selama 30 hari.**

### ➤ **Lama Waktu Pembuatan Aplikasi**

Dalam pengerjaannya, topbisnis membutuhkan estimasi waktu sekitar 1-3 bulan.

## **B. Aplikasi B**

Aplikasi ini merupakan aplikasi yang ditawarkan oleh jagowebdesign.com. Aplikasi ini merupakan aplikasi multidevice. Sama seperti aplikasi C, aplikasi ini juga merupakan aplikasi profesional yang cocok bagi para pebisnis. Adapun harga yang ditawarkan adalah Rp. 150.000.000 dengan detail sebagai berikut (Jagowebdesain.com).

### ➤ Detail spesifikasi Ojek Online

Aplikasi ini menyediakan paketan aplikasi dengan ketentuan sebagai berikut :

- Fitur : 20-30 Fitur
- Medium Custom
- Menyertakan dukungan seperti : home, about, contact, email, *maps*/lokasi, call.
- Pembuatan aplikasi dalam bentuk website
- Multi device  
Artinya dapat digunakan pada android, iphone serta windows phone
- Menyediakan fasilitas link ke sosial media
- Diupload ke playstore dan iphone market
- Design icon
- Memberikan tambahan dukungan berupa *maintanance*, optimasi, *security*, *support*, dan dapat melakukan *upload* ke android dan iphone.
- Dan untuk perpanjangan berikutnya dengan harga Rp 2.000.000

### ➤ Detail spesifikasi Ojek Online

Dalam pengerjaannya, jagowebdesain membutuhkan estimasi waktu sekitar 3-6 bulan.

Berdasarkan fitur yang ditawarkan oleh pihak ketiga dan pengolahan informasi yang penulis lakukan, hasil yang diperoleh dirangkum dalam **Tabel 6.14** berikut.

Tabel 6.14 Perbandingan Hasil Perhitungan Estimasi Biaya

Teknik	Effort (Orang)	Jumlah Waktu Pengerjaan (Hari)	Estimasi Biaya
<i>Function point</i>	23	170	Rp 196.100.000

Pemanfaatan Waktu Lembur	23	100	Rp 259.406.936
Penambahan SDM	40	100	Rp 189.300.000
Topbisnis.org	-	60	Rp 150.000.000
Jagowebde-sain.com	-	120	Rp 150.000.000

Jika dilakukan perbandingan dengan nilai yang ditawarkan oleh pihak ketiga yakni Rp 439.300.000 (belum termasuk security) kesimpulan yang dapat penulis ambil adalah biaya pembangunan aplikasi oleh pihak ketiga terbilang **Lebih Mahal**. Berdasarkan kesimpulan tersebut, maka aplikasi XYZ tepat untuk dijadikan sebagai studi kasus dalam rekayasa nilai yang berfokus pada penghematan biaya ini, karen jika aplikasi bernilai lebih murah, maka tidak perlu melakukan rekayasa nilai.

## 6.2 *Function Analysis Phase*

Sebelum melakukan pembangunan FAST Diagram pada tahapan selanjutnya, maka perlu dilakukan pembobotan terhadap kebutuhan fungsional aplikasi. Hasil pembobotan akan menentukan apa saja yang akan dimasukkan kedalam FAST Diagram dan menjadi komponen yang akan diolah pada tahapan-tahapan selanjutnya.

### 6.2.1 **Perhitungan Biaya Kebutuhan Fungsional**

Berdasarkan kebutuhan pembangunan FAST Diagram tersebut dilakukan pendefinisian *Functional requirement* atau kebutuhan fungsional dengan membuat list kebutuhan fungsional aplikasi. Adapun kebutuhan yang dituliskan adalah kebutuhan khusus yang disampaikan oleh pihak ketiga pada usulannya sebagai berikut.

1. Login. Hal ini memerlukan :

- EI : *Inputan* data akun (S)
- EO : Profil *Customer* (A)

- ILF : *Customer* (S)
  - EIF : -
  - EQ : Pendaftaran (A)
2. Aplikasi dapat menampilkannya informasi saldo. Hal ini memerlukan :
- EI : Transaksi pembelian (C), transaksi (C)
  - EO : Histori pembelian (C), Rekening koran (C)
  - ILF : Transaksi (C), *Customer* (S)
  - EIF : -
  - EQ : Validasi pop up (S), Top up saldo (A), Cek *Customer* (S)
3. Aplikasi dapat menampilkan aktivitas transaksi yang telah dilakukan *customer*. Hal ini memerlukan :
- EI : Transaksi pembelian (C), transaksi (C)
  - EO : Histori pembelian (C), Rekening koran (C)
  - ILF : Transaksi (C), *Customer* (S)
  - EIF : -
  - EQ : Cek *customer* (S), Cek Transaksi (S)
4. Aplikasi dapat melakukan pemesanan transportasi online, tiket vanue tertentu dan transaksi lain seperti hotel, pembelian barang, dan lain-lain. Hal ini memerlukan :
- EI : Informasi penyedia layanan dan jasa (C), Kata kunci lokasi (C), Transaksi pembelian (C), Transaksi (C)
  - EO : Informasi Lokasi (C), Histori Pembelian (C), Promo (S),
  - ILF : Transaksi (C), *Customer* (S)
  - EIF : -
  - EQ : Penjualan (C), Akses peta (C), Pembayaran (C), Cek Transaksi (S), Cek Promo (S), Cek *Customer* (S)

5. Aplikasi dapat menampilkan informasi pendamping dan penjemput atlet . Hal ini memerlukan :
  - EI : Informasi LO (S), Informasi Akomodasi (S), Informasi Transportasi (A)
  - EO : -
  - ILF : *Event Olahraga* (S), *Customer* (S)
  - EIF : -
  - EQ : *Input Event* (S), *Penyampaian informasi Event* (C), *Cek Customer* (S)
  
6. Aplikasi dapat menampilkan informasi hotel yang berada disekitar lokasi *Event*. Hal ini memerlukan :
  - EI : Informasi penyedia produk penyedia layanan dan jasa (C), Kata kunci lokasi (C)
  - EO : Informasi Lokasi (C), Promo (S)
  - ILF : Transaksi (C)
  - EIF : -
  - EQ : Akses peta (C), Cek Promo (S)
  
7. Aplikasi dapat menampilkan informasi *mall* yang menjadi *official partner* di sekitar lokasi pelaksanaan *Event* dan diskon produk tertentu yang ada selama *Event* berlangsung. Hal ini memerlukan :
  - EI : Informasi *mall* (S), Informasi produk penyedia layanan dan jasa (C), Kata kunci lokasi (C)
  - EO : Informasi Lokasi (C), Promo (S)
  - ILF : Promo penyedia jasa (A)
  - EIF : -
  - EQ : Akses peta (C), Cek Promo (S)
  
8. Aplikasi dapat menampilkan informasi jadwal dan lokasi pertandingan. Hal ini memerlukan :
  - EI : Informasi pertandingan (S)
  - EO : -

- ILF : *Event* olahraga (S)
  - EIF : -
  - EQ : *Melihat Event* (S)
9. Aplikasi dapat dimanfaatkan untuk menyampaikan keluhan atau menanyakan hal terkait layanan yang tersedia pada aplikasi. Hal ini memerlukan :
- EI : -
  - EO : -
  - ILF : *Event Olahraga* (S), *Customer* (S)
  - EIF : -
  - EQ : *Penyampaian Keluhan* (S), *Cek Customer* (S)
10. Aplikasi dapat memberikan petunjuk lokasi suatu tempat. Sama seperti akses peta, hal ini juga memerlukan kata kunci lokasi untuk mengakses peta yang tersimpan pada cloud.
- EI : *Kata Kunci Lokasi* (C)
  - EO : *Informasi lokasi* (C)
  - ILF : *Event Olahraga* (S)
  - EIF : -
  - EQ : *Akses Peta* (C)
11. Aplikasi dapat ditampilkan dalam berbagai bahasa, minima dua bahasa yakni Bahasa Inggris dan Bahasa Indonesia.
- EI : -
  - EO : -
  - ILF : *Database bahasa*(A)
  - EIF : -
  - EQ : -

Dalam melakukan perhitungan, penulis menggunakan perhitungan *Function point* dengan langkah yang sama seperti sebelumnya, penulis akan menghitung CFP masing-masing kebutuhan fungsional seperti pada Tabel **Tabel 6.15**.

Tabel 6.15 Perhitungan CFP Kebutuhan Fungsional

No	Kebutuhan Fungsional	EI		EO		ILF		EIF		EQ		Total
		C	D	C	D	C	D	C	D	C	D	
1.	Login	1	3	1	5	1	7			1	4	<b>18</b>
2.	Aplikasi dapat menampilkannya informasi saldo	2	6	2	7	1	7			2	3	<b>58</b>
						1	15			1	4	
3.	Aplikasi dapat menampilkan aktivitas transaksi yang telah dilakukan <i>customer</i>	2	6	2	7	1	7			2	3	<b>54</b>
						1	15					
4.	Aplikasi dapat melakukan pemesanan transportasi online, tiket vanue tertentu dan transaksi lain seperti hotel, pembelian barang, dan lain-lain	4	6	1	4	1	7			3	3	<b>91</b>
				2	7	1	15			3	6	
5.	Aplikasi dapat menampilkan informasi pendamping dan penjemput atlet	2	3			2	7			2	3	<b>36</b>
		1	4							1	6	
6.	Aplikasi dapat menampilkan informasi hotel yang berada disekitar lokasi <i>Event</i>	2	6	1	4	1	15			1	3	<b>47</b>
				1	7					1	6	

No	Kebutuhan Fungsional	EI		EO		ILF		EIF		EQ		Total
		C	D	C	D	C	D	C	D	C	D	
7.	Aplikasi dapat menampilkan informasi <i>mall</i> di sekitar lokasi pelaksanaan <i>Event</i> yang menjadi <i>official partner</i> dan diskon produk tertentu yang ada selama <i>Event</i> berlangsung	1	3	1	4	1	10			1	3	45
		2	6	1	7					1	6	
8.	Aplikasi dapat menampilkan informasi jadwal dan lokasi pertandingan	1	3			1	7			1	3	13
9.	Aplikasi dapat dimanfaatkan untuk menyampaikan keluhan atau menanyakan hal terkait layanan yang tersedia pada aplikasi					2	7			2	3	20
10	Aplikasi dapat memberikan petunjuk lokasi suatu	1	6	1	7	1	7			1	6	26
11	Aplikasi dapat ditampilkan dalam berbagai bahasa, minima dua bahasa yakni Bahasa Inggris dan Bahasa Indonesia	-	-	-	-	1	10			-	-	10

Berdasarkan perhitungan CFP dan hasil RCAF, selanjutnya dilakukan perhitungan FP dengan menggunakan rumus yang sama :

$$\text{FP} = \text{CFP} \times (0.65 + 0.01 \times \text{RCAF})$$

Dengan CFP diperoleh seperti pada **Tabel 5.14**, nilai RCAF akan menggunakan hasil yang sama seperti saat menghitung karakteristik keseluruhan aplikasi, yakni **46**. Nilai 0.65 dan 0.01 merupakan nilai default. Selanjutnya penulis melakukan perhitungan FP dengan persamaan yang sama seperti menghitung Estimasi biaya sebelumnya. Hasil yang diperoleh adalah sebagai berikut.

### 6.2.1.1 Login

Dalam perhitungan sebelumnya, perhitungan terkait login yang diperoleh adalah sebagai berikut.

- CFP : 18
- RCAF : 46
- J : 0.41

Selanjutnya dilakukan perhitungan *Function point* sebagai berikut:

$$\begin{aligned} \text{FP} &= \text{CFP} \times (0.65 + 0.01 \times \text{RCAF}) \\ &= 18 \times (0.65 + (0.01 \times 46)) \\ &= 18 \times (0,65 + 0,46) \\ &= 18 \times 1,11 \\ &= 19,918 \end{aligned}$$

Untuk mencari tahu *effort* dan waktu pengerjaan memerlukan nilai FP atau *f (Function point)* dan *j* (estimasi eksponensial oleh Jone's) dengan menggunakan persamaan seperti sebelumnya, nilai yang diperoleh adalah sebagai berikut.

$$\begin{aligned} \text{Effort} &= \text{FP}^{3*j} / 27 \\ &= 19,98^{3*0,41} / 27 \\ &= 19,98^{1,23} / 27 \\ &= 39,79 / 27 \\ &= 3 \text{ Orang} \end{aligned}$$

$$\begin{aligned}
 S &= f^j \\
 &= 19,98^{0,41} \\
 &= 3.4 \text{ bulan} \\
 &= 68 \text{ hari (20 hari kerja per-bulan)}
 \end{aligned}$$

Jika dilakukan *breakdown* maka kebutuhan akan 3 orang tersebut adalah seperti pada **Tabel 6.16**.

Tabel 6.16 Breakdown effort dan Waktu Pengerjaan Login

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	1	14	Rp 350.000	Rp 4.900.000
Spesification & Design	<i>System analyst</i>	1	14	Rp 350.000	Rp 4.900.000
Coding	<i>Programmer</i>	1	40	Rp 400.000	Rp 16.000.000
Testing	<i>Technical tester</i>	0	0	Rp 400.000	Rp 0
Acceptance and Deployment	<i>Technical Team leader</i>	0	0	Rp 750.000	Rp 0
<b>TOTAL</b>					<b>Rp 25.800.000</b>

### 6.2.1.2 Menampilkan Informasi Saldo

Dalam perhitungan sebelumnya, perhitungan terkait informasi saldo yang diperoleh adalah sebagai berikut.

- CFP : 58
- RCAF : 46
- J : 0.41

Selanjutnya dilakukan perhitungan *Function point* sebagai berikut:

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times RCAF) \\
 &= 58 \times (0.65 + (0.01 \times 46)) \\
 &= 58 \times (0,65 + 0,46) \\
 &= 58 \times 1,11 \\
 &= 64,38
 \end{aligned}$$

Untuk mencari tahu *effort* dan waktu pengerjaan memerlukan nilai FP atau  $f$  (*Function point*) dan  $j$  (estimasi eksponensial oleh Jone's) dengan menggunakan persamaan seperti sebelumnya, nilai yang diperoleh adalah sebagai berikut.

$$\begin{aligned}
 \text{Effort} &= FP^{3*j} / 27 \\
 &= 64.38^{3*0.41} / 27 \\
 &= 64.38^{1,23} / 27 \\
 &= 167,79 / 27 \\
 &= 6 \text{ Orang}
 \end{aligned}$$

$$\begin{aligned}
 S &= f^j \\
 &= 64.38^{0,41} \\
 &= 5.5 \text{ bulan} \\
 &= 110 \text{ hari (20 hari kerja per-bulan)}
 \end{aligned}$$

Jika dilakukan *breakdown* maka kebutuhan akan 6 orang tersebut adalah seperti pada **Tabel 6.17**.

Tabel 6.17 Breakdown effort dan Waktu Pengerjaan Informasi Saldo

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	1	15	Rp 350.000	Rp 5.250.000
Spesification & Design	<i>System analyst</i>	1	20	Rp 350.000	Rp 7.000.000

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
Coding	<i>Programmer</i>	2	40	Rp 400.000	Rp 32.000.000
Testing	<i>Technical tester</i>	1	25	Rp 400.000	Rp 10.000.000
Acceptance and Deployment	<i>Technical Team leader</i>	1	10	Rp 750.000	Rp 7.500.000
<b>TOTAL</b>					<b>Rp 61.750.000</b>

### 6.2.1.3 Menampilkan Aktivitas Transaksi

Dalam perhitungan sebelumnya, perhitungan terkait aktivitas transaksi yang diperoleh adalah sebagai berikut.

- CFP : 54
- RCAF : 46
- J : 0.41

Selanjutnya dilakukan perhitungan *Function point* sebagai berikut:

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times RCAF) \\
 &= 54 \times (0.65 + (0.01 \times 46)) \\
 &= 54 \times (0,65 + 0,46) \\
 &= 54 \times 1,11 \\
 &= 59,94
 \end{aligned}$$

Untuk mencari tahu *effort* dan waktu pengerjaan memerlukan nilai FP atau *f (Function point)* dan *j* (estimasi eksponensial oleh Jone's) dengan menggunakan persamaan seperti sebelumnya, nilai yang diperoleh adalah sebagai berikut.

$$\begin{aligned}
 \text{Effort} &= FP^{3*j} / 27 \\
 &= 59,94^{3*0,41} / 27
 \end{aligned}$$

$$= 59,94^{1,23} / 27$$

$$= 153,67 / 27$$

$$= 6 \text{ Orang}$$

$$S = f^j$$

$$= 59,94^{0,41}$$

$$= 5,3 \text{ bulan}$$

$$= 106 \text{ hari (20 hari kerja per-bulan)}$$

Jika dilakukan *breakdown* maka kebutuhan akan 6 orang tersebut adalah seperti pada **Tabel 6.18**.

Tabel 6.18 Breakdown effort dan Waktu Pengerjaan Aktivitas Transaksi

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	1	11	Rp 350.000	Rp 5.250.000
Spesification & Design	<i>System analyst</i>	1	20	Rp 350.000	Rp 7.000.000
Coding	<i>Programmer</i>	2	40	Rp 400.000	Rp 32.000.000
Testing	<i>Technical tester</i>	1	25	Rp 400.000	Rp 10.000.000
Acceptance and Deployment	<i>Technical Team leader</i>	1	10	Rp 750.000	Rp 7.500.000
<b>TOTAL</b>					<b>Rp 60.350.000</b>

### 6.2.1.4 Menampilkan Aktivitas Pemesanan Layanan dan Jasa

Dalam perhitungan sebelumnya, perhitungan terkait menampilkan aktivitas pemesanan layanan dan jasa yang diperoleh adalah sebagai berikut.

- CFP : 91
- RCAF : 46
- J : 0.41

Selanjutnya dilakukan perhitungan *Function point* sebagai berikut:

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times RCAF) \\
 &= 91 \times (0.65 + (0.01 \times 46)) \\
 &= 91 \times (0,65 + 0,46) \\
 &= 91 \times 1,11 \\
 &= 101,01
 \end{aligned}$$

Untuk mencari tahu *effort* dan waktu pengerjaan memerlukan nilai FP atau  $f$  (*Function point*) dan  $j$  (estimasi eksponensial oleh Jone's) dengan menggunakan persamaan seperti sebelumnya, nilai yang diperoleh adalah sebagai berikut.

$$\begin{aligned}
 \text{Effort} &= FP^{3*j} / 27 \\
 &= 101.01^{3*0.41} / 27 \\
 &= 101.01^{1.23} / 27 \\
 &= 291.99 / 27 \\
 &= 11 \text{ Orang}
 \end{aligned}$$

$$\begin{aligned}
 S &= f^j \\
 &= 101.01^{0.41} \\
 &= 6,6 \text{ bulan} \\
 &= 132 \text{ hari (20 hari kerja per-bulan)}
 \end{aligned}$$

Jika dilakukan *breakdown* maka kebutuhan akan 11 orang tersebut adalah seperti pada **Tabel 6.19**.

Tabel 6.19 Breakdown effort dan Waktu Pengerjaan Aktivitas Pemesanan Layanan dan Jasa

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	2	17	Rp 350.000	Rp 11.900.000
Spesification & Design	<i>System analyst</i>	2	20	Rp 350.000	Rp 14.000.000
Coding	<i>Programmer</i>	4	45	Rp 400.000	Rp 72.000.000
Testing	<i>Technical tester</i>	2	30	Rp 400.000	Rp 24.000.000
Acceptance and Deployment	<i>Technical Team leader</i>	1	20	Rp 750.000	Rp 15.000.000
<b>TOTAL</b>					<b>Rp 136.900.000</b>

### 6.2.1.5 Menampilkan Informasi Pendamping dan Penjemputan Atlet

Dalam perhitungan sebelumnya, perhitungan terkait menampilkan informasi pendamping dan penjemput atlet sekitar yang diperoleh adalah sebagai berikut.

- CFP : 36
- RCAF : 46
- J : 0.41

Selanjutnya dilakukan perhitungan *Function point* sebagai berikut:

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times RCAF) \\
 &= 36 \times (0.65 + (0.01 \times 46)) \\
 &= 36 \times (0,65 + 0,46)
 \end{aligned}$$

$$= 36 \times 1,11$$

$$= 39,96$$

Untuk mencari tahu *effort* dan waktu pengerjaan memerlukan nilai FP atau  $f$  (*Function point*) dan  $j$  (estimasi eksponensial oleh Jone's) dengan menggunakan persamaan seperti sebelumnya, nilai yang diperoleh adalah sebagai berikut.

$$\begin{aligned} \text{Effort} &= \text{FP}^{3*j} / 27 \\ &= 39,96^{3*0,41} / 27 \\ &= 39,96^{1,23} / 27 \\ &= 93,33 / 27 \\ &= 4 \text{ Orang} \end{aligned}$$

$$\begin{aligned} S &= f^j \\ &= 39,96^{0,41} \\ &= 4.5 \text{ bulan} \\ &= 90 \text{ hari (20 hari kerja per-bulan)} \end{aligned}$$

Jika dilakukan *breakdown* maka kebutuhan akan 4 orang tersebut adalah seperti pada **Tabel 6.20**.

Tabel 6.20 Breakdown effort dan Waktu Pengerjaan Informasi Pendamping dan Penjemput Atlet

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b><i>Software Development</i></b>					
Requirement	<i>Business analyst</i>	1	15	Rp 350.000	Rp 5.250.000
Spesification & Design	<i>System analyst</i>	1	15	Rp 350.000	Rp 5.250.000
Coding	<i>Programmer</i>	1	45	Rp 400.000	Rp 18.000.000
Testing	<i>Technical tester</i>	1	15	Rp 400.000	Rp 6.000.000

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
Acceptance and Deployment	<i>Technical Team leader</i>	0	0	Rp 750.000	Rp 0
<b>TOTAL</b>					<b>Rp 34.500.000</b>

### 6.2.1.6 Menampilkan Informasi Hotel di Sekitar

Dalam perhitungan sebelumnya, perhitungan terkait menampilkan informasi hotel di sekitar yang diperoleh adalah sebagai berikut.

- CFP : 47
- RCAF : 46
- J : 0.41

Selanjutnya dilakukan perhitungan *Function point* sebagai berikut:

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times RCAF) \\
 &= 47 \times (0.65 + (0.01 \times 46)) \\
 &= 47 \times (0,65 + 0,46) \\
 &= 47 \times 1,11 \\
 &= 52,17
 \end{aligned}$$

Untuk mencari tahu *effort* dan waktu pengerjaan memerlukan nilai FP atau *f (Function point)* dan *j* (estimasi eksponensial oleh Jone's) dengan menggunakan persamaan seperti sebelumnya, nilai yang diperoleh adalah sebagai berikut.

$$\begin{aligned}
 \text{Effort} &= FP^{3*j} / 27 \\
 &= 52.17^{3*0.41} / 27 \\
 &= 52.17^{1,23} / 27 \\
 &= 129.55 / 27 \\
 &= 5 \text{ Orang}
 \end{aligned}$$

$$\begin{aligned}
 S &= f^j \\
 &= 52.17^{0,41} \\
 &= 5 \text{ bulan} \\
 &= 100 \text{ hari (20 hari kerja per-bulan)}
 \end{aligned}$$

Jika dilakukan *breakdown* maka kebutuhan akan 5 orang tersebut adalah seperti pada **Tabel 6.21**.

Tabel 6.21 Breakdown effort dan Waktu Pengerjaan Informasi Hotel

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	1	15	Rp 350.000	Rp 5.250.000
Spesification & Design	<i>System analyst</i>	1	15	Rp 350.000	Rp 5.250.000
Coding	<i>Programmer</i>	1	35	Rp 400.000	Rp 14.000.000
Testing	<i>Technical tester</i>	1	20	Rp 400.000	Rp 8.000.000
Acceptance and Deployment	<i>Technical Team leader</i>	1	15	Rp 750.000	Rp 11.250.000
<b>TOTAL</b>					<b>Rp 43.750.000</b>

### 6.2.1.7 Menampilkan Informasi Mall *Official Partner* di Sekitar

Dalam perhitungan sebelumnya, perhitungan terkait menampilkan informasi *mall official partner* yang diperoleh adalah sebagai berikut.

- CFP : 45
- RCAF : 46

- $J = 0.41$

Selanjutnya dilakukan perhitungan *Function point* sebagai berikut:

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times RCAF) \\
 &= 45 \times (0.65 + (0.01 \times 46)) \\
 &= 45 \times (0,65 + 0,46) \\
 &= 45 \times 1,11 \\
 &= 49,95
 \end{aligned}$$

Untuk mencari tahu *effort* dan waktu pengerjaan memerlukan nilai FP atau  $f$  (*Function point*) dan  $j$  (estimasi eksponensial oleh Jone's) dengan menggunakan persamaan seperti sebelumnya, nilai yang diperoleh adalah sebagai berikut.

$$\begin{aligned}
 \text{Effort} &= FP^{3*j} / 27 \\
 &= 49.95^{3*0.41} / 27 \\
 &= 49.95^{1.23} / 27 \\
 &= 122.8 / 27 \\
 &= 5 \text{ Orang}
 \end{aligned}$$

$$\begin{aligned}
 S &= f^j \\
 &= 49.95^{0.41} \\
 &= 5 \text{ bulan} \\
 &= 100 \text{ hari (20 hari kerja per-bulan)}
 \end{aligned}$$

Jika dilakukan *breakdown* maka kebutuhan akan 5 orang tersebut adalah seperti pada **Tabel 6.22**.

Tabel 6.22 Breakdown effort dan Waktu Pengerjaan Aktivitas Menampilkan *Mall Official Partner*

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	1	15	Rp 350.000	Rp 5.250.000

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
Spesification & Design	<i>System analyst</i>	1	15	Rp 350.000	Rp 5.250.000
Coding	<i>Programmer</i>	1	35	Rp 400.000	Rp 14.000.000
Testing	<i>Technical tester</i>	1	20	Rp 400.000	Rp 8.000.000
Acceptance and Deployment	<i>Technical Team leader</i>	1	15	Rp 750.000	Rp 11.250.000
<b>TOTAL</b>					<b>Rp 43.750.000</b>

### 6.2.1.8 Menampilkan Informasi Jadwal dan Lokasi Pertandingan

Dalam perhitungan sebelumnya, perhitungan terkait menampilkan informasi jadwal dan lokasi pertandingan yang diperoleh adalah sebagai berikut.

- CFP : 13
- RCAF : 46
- J : 0.41

Selanjutnya dilakukan perhitungan *Function point* sebagai berikut:

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times RCAF) \\
 &= 13 \times (0.65 + (0.01 \times 46)) \\
 &= 13 \times (0,65 + 0,46) \\
 &= 13 \times 1,11 \\
 &= 14,43
 \end{aligned}$$

Untuk mencari tahu *effort* dan waktu pengerjaan memerlukan nilai FP atau *f (Function point)* dan *j* (estimasi eksponensial oleh

Jone's) dengan menggunakan persamaan seperti sebelumnya, nilai yang diperoleh adalah sebagai berikut.

$$\begin{aligned}
 \text{Effort} &= FP^{3^j} / 27 \\
 &= 14,43^{3 \cdot 0,41} / 27 \\
 &= 14,43^{1,23} / 27 \\
 &= 26,66 / 27 \\
 &= 1 \text{ Orang}
 \end{aligned}$$

$$\begin{aligned}
 S &= f^j \\
 &= 14,43^{0,41} \\
 &= 3 \text{ bulan} \\
 &= 60 \text{ hari (20 hari kerja per-bulan)}
 \end{aligned}$$

Jika dilakukan *breakdown* maka kebutuhan akan 1 orang tersebut adalah seperti pada **Tabel 6.23**.

Tabel 6.23 Breakdown effort dan Waktu Pengerjaan Menampilkan Informasi Jadwal dan Pertandingan

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	0	0	Rp 350.000	Rp 0
Spesification & Design	<i>System analyst</i>	0	0	Rp 350.000	Rp 0
Coding	<i>Programmer</i>	1	60	Rp 400.000	Rp 24.000.000
Testing	<i>Technical tester</i>	0	0	Rp 400.000	Rp 0
Acceptance and Deployment	<i>Technical Team leader</i>	0	0	Rp 750.000	Rp 0
<b>TOTAL</b>					<b>Rp 24.000.000</b>

### 6.2.1.9 Menyampaikan keluhan

Dalam perhitungan sebelumnya, perhitungan terkait menyampaikan keluhan pertandingan yang diperoleh adalah sebagai berikut.

- CFP : 20
- RCAF : 46
- J : 0.41

Selanjutnya dilakukan perhitungan *Function point* sebagai berikut:

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times RCAF) \\
 &= 20 \times (0.65 + (0.01 \times 46)) \\
 &= 20 \times (0,65 + 0,46) \\
 &= 20 \times 1,11 \\
 &= 22,2
 \end{aligned}$$

Untuk mencari tahu *effort* dan waktu pengerjaan memerlukan nilai FP atau *f* (*Function point*) dan *j* (estimasi eksponensial oleh Jone's) dengan menggunakan persamaan seperti sebelumnya, nilai yang diperoleh adalah sebagai berikut.

$$\begin{aligned}
 \text{Effort} &= FP^{3*j} / 27 \\
 &= 22,2^{3*0,41} / 27 \\
 &= 22,2^{1,23} / 27 \\
 &= 45,29 / 27 \\
 &= 2 \text{ Orang}
 \end{aligned}$$

$$\begin{aligned}
 S &= f^j \\
 &= 22,2^{0,41} \\
 &= 3.6 \text{ bulan} \\
 &= 72 \text{ hari (20 hari kerja per-bulan)}
 \end{aligned}$$

Jika dilakukan *breakdown* maka kebutuhan akan 2 orang tersebut adalah seperti pada **Tabel 6.24**.

Tabel 6.24 Breakdown effort dan Waktu Pengerjaan Aktivitas Menyampaikan Keluhan

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	1	22	Rp 350.000	Rp 7.700.000
Spesification & Design	<i>System analyst</i>	0	0	Rp 350.000	Rp 0
Coding	<i>Programmer</i>	1	50	Rp 400.000	Rp 20.000.000
Testing	<i>Technical tester</i>	0	0	Rp 400.000	Rp 0
Acceptance and Deployment	<i>Technical Team leader</i>	0	0	Rp 750.000	Rp 0
<b>TOTAL</b>					<b>Rp 27.700.000</b>

### 6.2.1.10 Memberikan Informasi Lokasi

Dalam perhitungan sebelumnya, perhitungan terkait memberikan petunjuk lokasi yang diperoleh adalah sebagai berikut.

- CFP : 26
- RCAF : 46
- J : 0.41

Selanjutnya dilakukan perhitungan *Function point* sebagai berikut:

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times RCAF) \\
 &= 26 \times (0.65 + (0.01 \times 46)) \\
 &= 26 \times (0,65 + 0,46) \\
 &= 26 \times 1,11 \\
 &= 28,86
 \end{aligned}$$

Untuk mencari tahu *effort* dan waktu pengerjaan memerlukan nilai FP atau  $f$  (*Function point*) dan  $j$  (estimasi eksponensial oleh Jone's) dengan menggunakan persamaan seperti sebelumnya, nilai yang diperoleh adalah sebagai berikut.

$$\begin{aligned} \text{Effort} &= \text{FP}^{3*j} / 27 \\ &= 28,86^{3*0,41} / 27 \\ &= 28,86^{1,23} / 27 \\ &= 62.5 / 27 \\ &= 2 \text{ Orang} \end{aligned}$$

$$\begin{aligned} S &= f^j \\ &= 28,86^{0,41} \\ &= 4 \text{ bulan} \\ &= 80 \text{ hari (20 hari kerja per-bulan)} \end{aligned}$$

Jika dilakukan *breakdown* maka kebutuhan akan 2 orang tersebut adalah seperti pada **Tabel 6.25**.

Tabel 6.25 Breakdown effort dan Waktu Pengerjaan Aktivitas Memberi Petunjuk Lokasi

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	1	20	Rp 350.000	Rp 7.000.000
Spesification & Design	<i>System analyst</i>	0	0	Rp 350.000	Rp 0
Coding	<i>Programmer</i>	1	60	Rp 400.000	Rp 24.000.000
Testing	<i>Technical tester</i>	0	0	Rp 400.000	Rp 0
Acceptance and Deployment	<i>Technical Team leader</i>	0	0	Rp 750.000	Rp 0
<b>TOTAL</b>					<b>Rp 31.000.000</b>

### 6.2.1.11 Multiple Language

Dalam perhitungan sebelumnya, perhitungan terkait memberikan petunjuk lokasi yang diperoleh adalah sebagai berikut.

- CFP : 10
- RCAF : 46
- J : 0.41

Selanjutnya dilakukan perhitungan *Function point* sebagai berikut:

$$\begin{aligned}
 FP &= CFP \times (0.65 + 0.01 \times RCAF) \\
 &= 10 \times (0.65 + (0.01 \times 46)) \\
 &= 10 \times (0,65 + 0,46) \\
 &= 10 \times 1,11 \\
 &= 11,1
 \end{aligned}$$

Untuk mencari tahu *effort* dan waktu pengerjaan memerlukan nilai FP dengan menggunakan persamaan seperti sebelumnya, nilai yang diperoleh adalah sebagai berikut.

$$\begin{aligned}
 \text{Effort} &= FP^{3*j} / 27 \\
 &= 11,1^{3*0.41} / 27 \\
 &= 11,1^{1,23} / 27 \\
 &= 19,3 / 27 \\
 &= 1 \text{ Orang}
 \end{aligned}$$

$$\begin{aligned}
 S &= f^j \\
 &= 11,1^{0,41} \\
 &= 2,7 \text{ bulan} \\
 &= 54 \text{ hari (20 hari kerja per-bulan)}
 \end{aligned}$$

Jika dilakukan *breakdown* maka kebutuhan akan 2 orang tersebut adalah seperti pada **Tabel 6.26**.

Tabel 6.26 Breakdown effort dan Waktu Pengerjaan Aktivitas  
Memberi Petunjuk Lokasi

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	0	0	Rp 350.000	Rp 0
Spesification & Design	<i>System analyst</i>	0	0	Rp 350.000	Rp 0
Coding	<i>Programmer</i>	1	54	Rp 400.000	Rp 21.600.000
Testing	<i>Technical tester</i>	0	0	Rp 400.000	Rp 0
Acceptance and Deployment	<i>Technical Team leader</i>	0	0	Rp 750.000	Rp 0
<b>TOTAL</b>					<b>Rp 21.600.000</b>

### 6.2.2 Pemilihan Objek Penelitian

Berdasarkan hasil perhitungan biaya kebutuhan fungsional, selanjutnya penulis akan melakukan pemilihan fungsi yang akan dijadikan sebagai objek penelitian. Untuk memudahkan makahasil perhitungan yang diperoleh sebelumnya dimasukkan kedalam **Tabel 6.27.** untuk kemudian dilakukan perhitungan bobot.

Tabel 6.27 Estimasi Biaya Kebutuhan Fungsional

No	Kebutuhan Fungsional	FP	Biaya	Bobot
1.	Login	19.98	<b>Rp 25.800.000</b>	<b>5.0%</b>

No	Kebutuhan Fungsional	FP	Biaya	Bobot
2.	Aplikasi dapat menampilkannya informasi saldo	64.38	<b>Rp 61.750.000</b>	<b>12.1%</b>
3.	Aplikasi dapat menampilkan aktivitas transaksi yang telah dilakukan <i>customer</i>	59.94	<b>Rp 60.350.000</b>	<b>11.8%</b>
4.	Aplikasi dapat melakukan pemesanan transportasi online, tiket vanue tertentu dan transaksi lain seperti hotel, pembelian barang, dan` lain-lain	101.01	<b>Rp 136.900.000</b>	<b>26.8%</b>
5.	Aplikasi dapat menampilkan informasi pendamping dan penjemput atlet	39.96	<b>Rp 34.500.000</b>	<b>6.8%</b>
6.	Aplikasi dapat menampilkan informasi hotel yang berada disekitar lokasi <i>Event</i>	52.17	<b>Rp 43.750.000</b>	<b>8.6%</b>
7.	Aplikasi dapat menampilkan informasi <i>mall</i> yang menjadi <i>official partner</i> dan diskon produk tertentu	49.95	<b>Rp 43.750.000</b>	<b>8.6%</b>
8.	Aplikasi dapat menampilkan	14.43	<b>Rp 24.000.000</b>	<b>4.7%</b>

No	Kebutuhan Fungsional	FP	Biaya	Bobot
	informasi jadwal dan lokasi pertandingan			
9.	Aplikasi dapat dimanfaatkan untuk menyampaikan keluhan atau menanyakan hal terkait layanan yang tersedia pada aplikasi	22.2	Rp 27.700.000	5.4%
10.	Aplikasi dapat memberikan petunjuk lokasi suatu tempat	28.86	Rp 31.000.000	6.1%
11	Aplikasi dapat ditampilkan dalam berbagai bahasa, minima dua bahasa yakni Bahasa Inggris dan Bahasa Indonesia	11.1	Rp 21.600.000	4.2%
<b>TOTAL</b>			<b>Rp 511.100.000</b>	<b>100%</b>

Berdasarkan hasil tersebut, selanjutnya perlu mencari fungsi yang bersifat *complex*. Berdasarkan sumber yang penulis dapat, fungsi *complex* adalah fungsi dengan *range* pengerjaan 600-800 jam. Berdasarkan perhitungan FP, penulis dapat menghitung waktu pengerjaan fungsi. Hasil yang diperoleh adalah sebagai berikut :

Tabel 6. 28 Penentuan Objek Penelitian

No	Kebutuhan Fungsional	Waktu pengerjaan (Jam)	Jumlah SDM	Keterangan
1.	Login	544	3	Ditolak

No	Kebutuhan Fungsional	Waktu pengerjaan (Jam)	Jumlah SDM	Keterangan
2.	Aplikasi dapat menampilkannya informasi saldo	880	6	Diterima
3.	Aplikasi dapat menampilkan aktivitas transaksi yang telah dilakukan <i>customer</i>	848	6	Diterima
4.	Aplikasi dapat melakukan pemesanan transportasi online, tiket vanue tertetu dan transaksi lain seperti hotel, pembelian barang, dan` lain-lain	1.056	11	Diterima
5.	Aplikasi dapat menampilkan informasi pendamping dan penjemput atlet	720	4	Diterima
6.	Aplikasi dapat menampilkan informasi hotel yang berada disekitar lokasi <i>Event</i>	800	5	Diterima
7.	Aplikasi dapat menampilkan informasi <i>mall</i>	800	5	Diterima

No	Kebutuhan Fungsional	Waktu pengerjaan (Jam)	Jumlah SDM	Keterangan
	yang menjadi <i>official partner</i> dan diskon produk tertentu			
8.	Aplikasi dapat menampilkan informasi jadwal dan lokasi pertandingan	480	1	Ditolak
9.	Aplikasi dapat dimanfaatkan untuk menyampaikan keluhan atau menanyakan hal terkait layanan yang tersedia pada aplikasi	576	2	Ditolak
10.	Aplikasi dapat memberikan petunjuk lokasi suatu tempat	640	2	Diterima
11	Aplikasi dapat ditampilkan dalam berbagai bahasa, minima dua bahasa yakni Bahasa Inggris dan Bahasa Indonesia	432	1	Ditolak

Berdasarkan hasil tersebut, diperoleh 7 kebutuhan fungsional utama yang dapat dijadikan objek dalam pengerjaan Tugas Akhir ini. Pada fase ini penulis akan melakukan analisa lebih

lanjut hingga sampai pada tahapan pembangunan FAST Diagram masing-masing objek.

## 6.2.3 Pembangunan FAST Diagram

### 6.2.3.1 Menampilkan Informasi Saldo

Menampilkan informasi saldo. Fungsi ini berlaku sejak *customer* melakukan *login* hingga informasi saldo *customer* ditampilkan (**Gambar 6.7**). Secara lebih detail informasi saldo dapat dipahami dengan menjawab pertanyaan berikut ini :

1. “Bagaimana fungsi tersebut dapat diperoleh?”

Pada fungsi menampilkan saldo sebagai *higher order Function*, memerlukan pengecekan histori transaksi dengan mengakses database transaksi sesuai dengan profil masing-masing *customer*, sehingga diperlukan juga aktivitas cek transaksi.

2. “Mengapa fungsi tersebut bisa dijalankan?”

Fungsi ini bisa dijalankan karena *system* dapat melakukan cek transaksi dari *customer* yang sesuai dengan profile, kemudian melakukan pengurangan saldo sesuai dengan nilai pembelian yang dilakukan oleh *customer* serta melakukan penambahan jumlah saldo sesuai dengan pembelian top-up yang dilakukan *customer*. Hal tersebut akan dicatatkan kedalam database transaksi untuk kemudian dapat diakses untuk menampilkan informasi saldo.

Untuk mendetailkan ketercapaian fungsi, diperlukan jawaban atas pertanyaan berikut:

- a. “Kapan fungsi tersebut digunakan?”

Fungsi pengurangan jumlah saldo dapat dilakukan ketika *system* telah menampilkan pemesanan dan total nilai pemesanan kepada *customer* untuk melakukan pembayaran, yang selanjutnya *system* akan

melakukan validasi atau penyesuaian antara nilai yang dibayarkan dengan total nilai pemesanan. Selain itu, penambahan saldo dapat dilakukan ketika *system* mampu menampilkan pemesanan top-up, penyesuaian harga pembayaran dengan total nilai top-up dan melakukan validasi top up.

### 6.2.3.2 Menampilkan Aktivitas Transaksi

Menampilkan aktivitas transaksi. Fungsi ini berlaku sejak *customer* melakukan login hingga perekaman aktivitas transaksi yang dilakukan *customer*. Transaksi dalam hal ini bukan hanya berhasil atau gagal, namun memasukkan pesanan kedalam keranjang, melakukan pemesanan dan belum dibayar, dll juga termasuk kedalam transaksi (**Gambar 6.8**). Secara lebih detail informasi saldo dapat dipahami dengan menjawab pertanyaan berikut ini :

1. “Bagaimana fungsi tersebut bisa diperoleh?”

Aktivitas transaksi dapat ditampilkan jika *system* dapat mengakses database transaksi dan melakukan cek *customer*.

2. “Mengapa fungsi tersebut bisa dijalankan?”

Fungsi ini dapat dijalankan jika *system* dapat menampilkan produk, jasa dan layanan yang tersedia kepada *customer* untuk dilakukan pemesanan. Kemudian *system* dapat menampilkan hasil pemesanan *customer* untuk dibagikan kedalam beberapa kategori. Kemudian disimpan kedalam database transaksi untuk kemudian diakses oleh *system* untuk menampilkan hasil transaksi tersebut sesuai dengan kategori.

Untuk mendetailkan ketercapaian fungsi, diperlukan jawaban atas pertanyaan berikut:

- a. “Kapan fungsi tersebut digunakan?”

Fungsi ini digunakan ketika *customer* ingin mengetahui status transaksi mereka berdasarkan kategori yang ada pada *system*. Kategori tersebut meliputi transaksi yang berhasil dilakukan, transaksi yang gagal dilakukan, transaksi yang sudah dibayar dan sudah digunakan, transaksi yang sudah dibayar namun belum digunakan, transaksi yang sudah di-*booking* atau dilist namun belum dibayar dan transaksi yang sedang diproses.

### 6.2.3.3 Menampilkan Aktivitas Pemesanan Layanan dan Jasa

Menampilkan pemesanan produk layanan dan jasa. Fungsi ini adalah fungsi yang dijalankan dari mulai *customer* login hingga hingga pemesanan dikatakan gagal atau berhasil (**Gambar 6.9**). Secara lebih detail informasi saldo dapat dipahami dengan menjawab pertanyaan berikut ini :

1. “Bagaimana fungsi tersebut bisa diperoleh?”

Fungsi ini dapat diperoleh dengan pengaksesan database transaksi yang dilakukan oleh sistem untuk mengakses data penyimpanan transaksi yang berhasil dan gagal setelah pembayaran dilakukan *customer*. Selain itu, fungsi ini juga akan berjalan jika sistem mampu melakukan cek *customer*, sebab masing-masing *customer* akan menghasilkan aktivitas pemesanan yang berbeda-beda.

2. “Mengapa fungsi tersebut bisa dijalankan?”

Fungsi ini dapat dijalankan karena setelah *customer* melakukan login, sistem mampu melakukan pengecekan penyedia jasa dan layanan untuk kemudian ditampilkan kedalam satu halaman yang terdiri atas 2 tipe, yakni informasi penyedia jasa dan layanan itu sendiri dan informasi kategori produk jasa dan layanan yang tersedia.

Informasi penyedia jasa dan layanan memerlukan beberapa tahapan lain seperti sistem menampilkan seluruh produk atau jasa yang tersedia, dan menampilkan halaman katalog penyedia jasa dan layanan sesuai pilihan *customer*. Selanjutnya kedua tipe tersebut akan sama-sama menampilkan halaman detail produk yang *customer* pilih dan jika *customer* telah melakukan transaksi, maka sistem akan menampilkan halaman status transaksi, menyimpan aktivitas transaksi tersebut kedalam database transaksi sehingga dapat diakses untuk ditampilkan.

Untuk mendetailkan ketercapaian fungsi, diperlukan jawaban atas pertanyaan berikut:

- a. “Kapan fungsi tersebut digunakan?”

Fungsi halaman status dapat digunakan saat *customer* telah melakukan aktivitas pemesanan. Sistem akan memasukkan pesanan kedalam keranjang belanja, menampilkan total harga pesanan *customer*, menampilkan informasi detail pembelian seperti: cara pembayaran, nomor resi, dll. Selanjutnya sistem akan melakukan validasi pembayaran *customer* dengan menyesuaikan harga total pemesanan dengan jumlah pembayaran *customer*. Transaksi *customer* dikatakan berhasil ketika nilai total pembayaran sesuai dengan harga yang dibayarkan. Transaksi dikatakan gagal jika, nilai total pembayaran tidak sesuai dengan harga yang dibayarkan atau terjadi kesalahan jaringan saat melakukan transaksi.

#### **6.2.3.4 Menampilkan Informasi Pendamping dan Penjemputan**

Menampilkan informasi pendamping dan penjemputan merupakan aktivitas untuk menampilkan informasi kepada pengguna secara bertim. Pada aktivitas ini informasi akan ditampilkan secara keseluruhan, sehingga tidak memerlukan

pengecekan *customer* (**Gambar 6.10**). Secara lebih detail dapat dieproleh degan menjawab pertanyaan berikut :

1. “Bagaimana fungsi tersebut bisa diperoleh?”

Fungsi ini diperoleh dengan melakukan akses database *Event* olahraga dan sistem terlebih dahulu menampilkan halaman home aplikasi, yang berisi banyak fitur-fitur terkait.

2. “Mengapa fungsi tersebut bisa dijalankan?”

Fungsi ini dapat dijalankan karena sistem dapat menampilkan halaman *Event* olahraga yang sebelumnya telah dipilih oleh *customer*, kemudian halaman tersebut akan diteruskan kedalam halaman yang berisi kategori olahraga, kemudian dilakukan akses database *Event* olahraga untuk kemudian informasi pendamping dan penjemputan ditampilkan.

Untuk mendetailkan ketercapaian fungsi, diperlukan jawaban atas pertanyaan berikut:

- a. “Kapan fungsi tersebut digunakan?”

Fungsi halaman kategori akan berisi informasi jadwal pertandingan, lokasi pertandingan dan informasi pendampingan dan penjemutan untuk selanjutnya dapat dipilih oleh *customer* tergantung apa yang ingin ditampilkan. Dalam hal ini ingin menampilkan pendampingan dan penjemputan, maka *customer* harus memilih kategori tersebut.

#### **6.2.3.5 Menampilkan Informasi Hotel di Sekitar**

Menampilkan informasi hotel disekitar merupakan aktivitas yang hanya sampai pada menampilkan informasi tidak ada keterlibatan pemesanan, dll (**Gambar 6.11**). Secara lebih detail

informasi saldo dapat dipahami dengan menjawab pertanyaan berikut ini :

1. “Bagaimana fungsi tersebut bisa diperoleh?”

Fungsi ini diperoleh dengan melakukan akses peta lokasi untuk mengetahui lokasi hotel yang berada disekitar kita. Selain itu, sistem mampu menampilkan *inputan* hotel yang berisi : lokasi, waktu check in dan durasi menginap.

2. “Mengapa fungsi tersebut bisa dijalankan?”

Fungsi ini dapat dijalankan karena sistem dapat mengakses informasi penyedia layanan dan jasa yang sesuai dengan *inputan* serta menampilkan hotel yang *available* sesuai dengan informasi *inputan*. Sistem dapat menampilkan kategori pencarian dengan *filter and sort* dan *map view* untuk membawa sistem kedalam pengaksesan peta hingga memunculkan informasi hotel sekitar.

Untuk mendetailkan ketercapaian fungsi, diperlukan jawaban atas pertanyaan berikut:

- a. “Kapan fungsi tersebut digunakan?”

Fungsi *filter and sort* digunakan untuk memudahkan *customer* dalam melakukan penyaringan hotel berdasarkan *filter* lokasi sehingga lebih mudah untuk mencari hotel yang tersedia disekitar. Adapun *filter and sort* akan diisi berdasarkan harga terendah, harga tertinggi, tipe kamar, tipe tempat tidur, filter lokasi dan range harga per-kamar.

#### **6.2.3.6 Menampilkan Informasi Mall *Official Partner* di Sekitar**

Menampilkan informasi *mall official partner* merupakan aktivitas yang hanya sampai pada menampilkan informasi tidak ada keterlibatan pemesanan, dll (**Gambar 6.12**). Secara lebih

detail informasi saldo dapat dipahami dengan menjawab pertanyaan berikut ini :

1. “Bagaimana fungsi tersebut dapat dijalankan?”

Fungsi ini diperoleh dengan melakukan akses peta lokasi untuk mengetahui lokasi *mall official partner* yang berada disekitar kita. Selain itu, sistem perlu menampilkan tampilan awal informasi aplikasi terlebih dahulu.

2. “Mengapa fungsi tersebut digunakan?”

Fungsi ini dapat dijalankan karena sistem melakukan cek informasi penyedia layanan dan jasa untuk secara keseluruhan penyedia ditampilkan dalam satu halaman. Selanjutnya *customer* akan memilih kategori *mall* dan sistem akan menampilkan halaman informasi *mall* dengan kategori. *Customer* akan memilih kategori *official partner* untuk selanjutnya sistem akan menampilkan informasi *mall official partner*. Pada halaman ini akan didapati 2 kategori pilihan yang keduanya dapat membawa *customer* untuk mencari informasi *mall official partner* yang berada disekitar lokasi *Event* yakni berdasarkan *filter and sort* dan *map view* yang selanjutnya dengan mengakses peta maka sistem dapat menampilkan informasi *mall official partner* yang berada disekitar.

Untuk mendetailkan ketercapaian fungsi, diperlukan jawaban atas pertanyaan berikut:

- a. “Kapan fungsi tersebut digunakan?”

Fungsi kategori informasi *mall* digunakan untuk memudahkan *customer* untuk mencari *mall official partner* karena kategori tersebut akan dibagi 2 yakni *official partner* dan *others*, sehingga *customer* dapat lebih cepat memperoleh informasi *mall* apa saja yang menjadi *official partner*. Fungsi *filter and sort* digunakan untuk memudahkan *customer* dalam

melakukan penyaringan *mall official partner* berdasarkan *filter* lokasi sehingga lebih mudah untuk mencari *mall* yang tersedia disekitar. Filter ini akan diisi rekomendasi, jarak dari landmark dan lokasi.

### 6.2.3.7 Menampilkan Informasi Lokasi

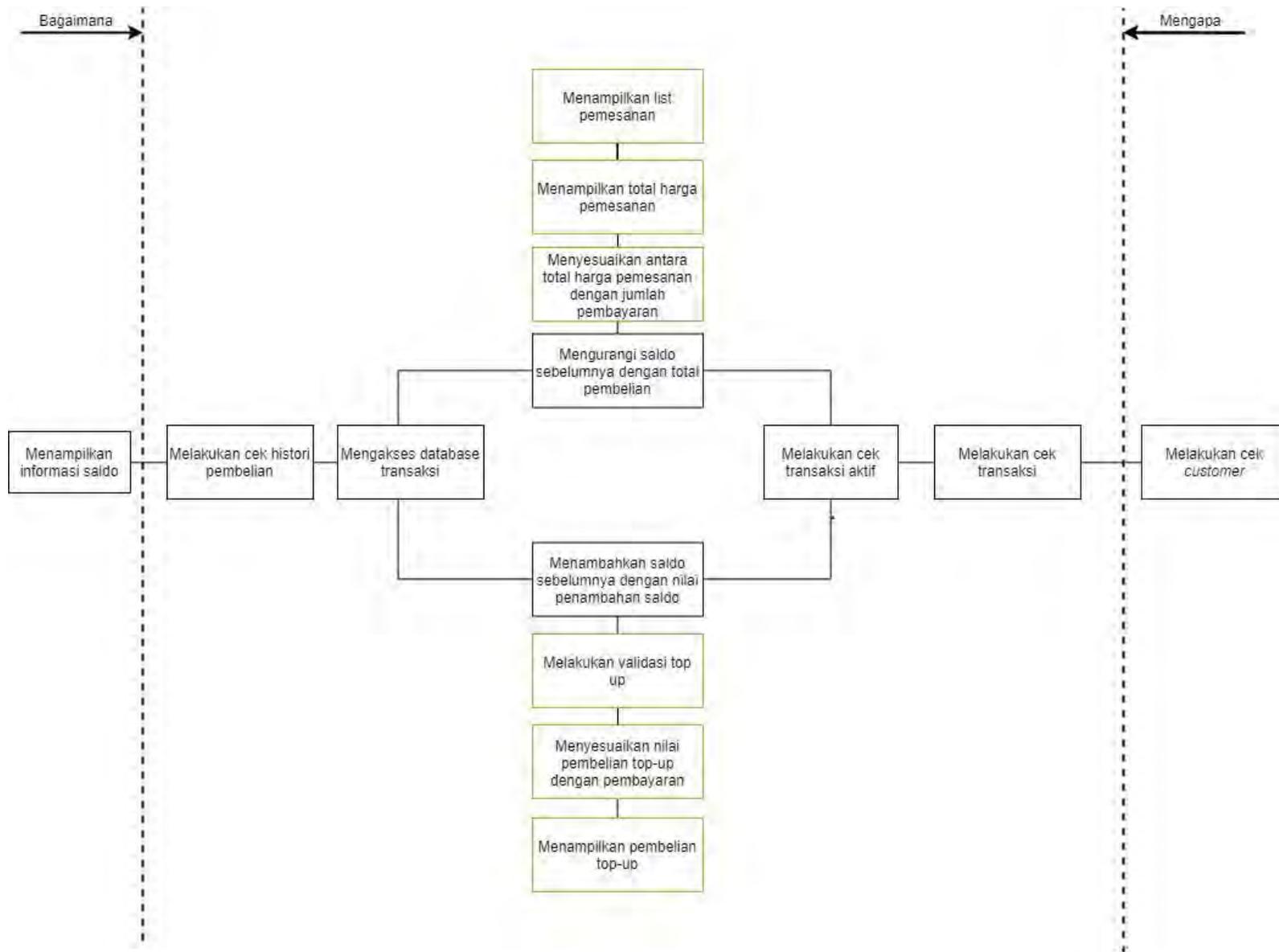
Menampilkan informasi lokasi merupakan aktivitas menampilkan informasi lokasi seperti pada *maps*, aktivitas ini bermula sejak kata kunci lokasi dimasukkan hingga peta menampilkan informasi jarak dengan garis berwarna biru (**Gambar 6.13**). Secara lebih detail informasi saldo dapat dipahami dengan menjawab pertanyaan berikut ini :

1. “Bagaimana fungsi tersebut dapat dijalankan?”  
Fungsi ini dijalankan dengan sistem menampilkan halaman *input* kata kunci lokasi. Selain itu sistem juga harus mampu mengakses peta sehingga dapat menampilkan informasi lokasi.
2. “Mengapa fungsi tersebut digunakan?”  
Fungsi ini dapat digunakan karena sistem mampu mengakses peta untuk selanjutnya menampilkan halaman dengan dua kepentingan, yakni halaman peta (*view maps*) dan halaman detail informasi terkait lokasi (jarak tempuh, waktu tempuh, deskripsi singkat, dll), kemudian sistem akan menampilkan halaman baru yang berisi pilihan rute perjalanan yang diinginkan, kemudian diakses melalui akses peta untuk ditampilkan.

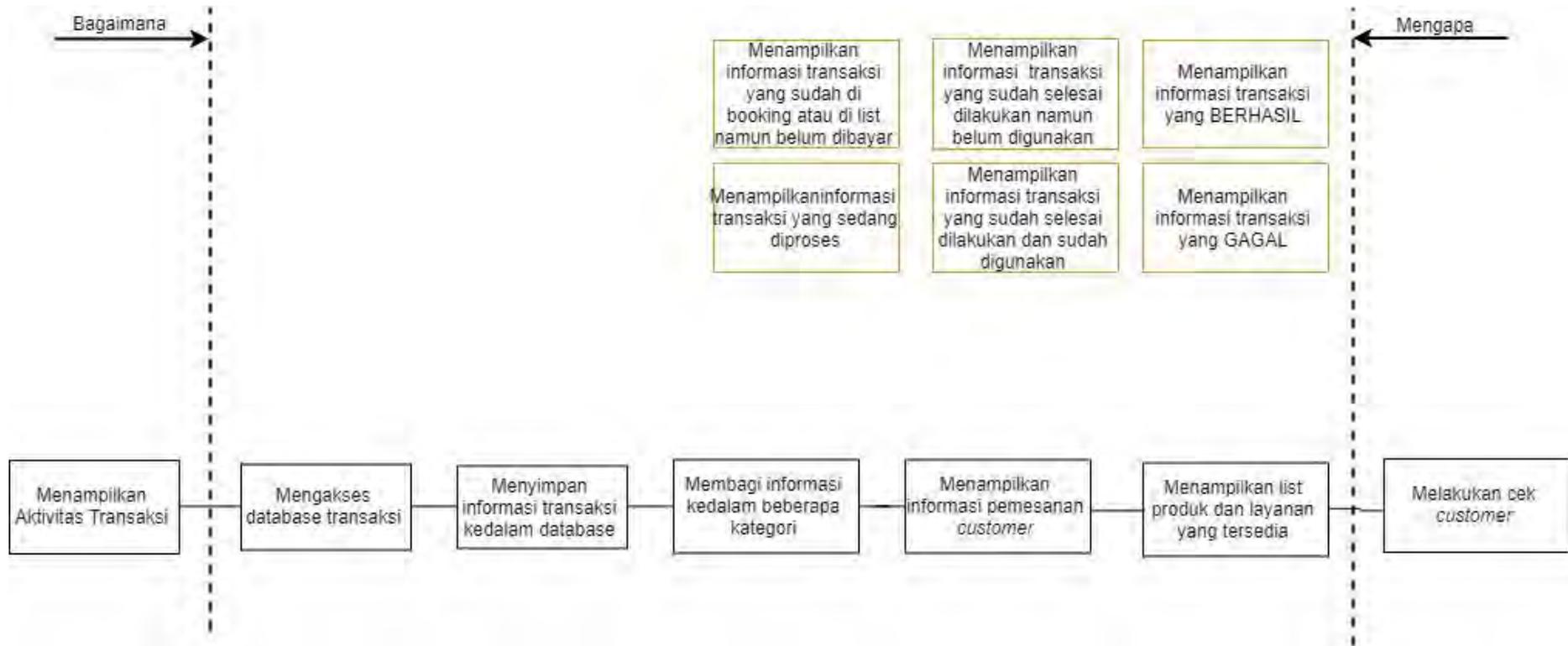
Untuk mendetailkan ketercapaian fungsi, diperlukan jawaban atas pertanyaan berikut:

- a. “Kapan fungsi tersebut digunakan?”

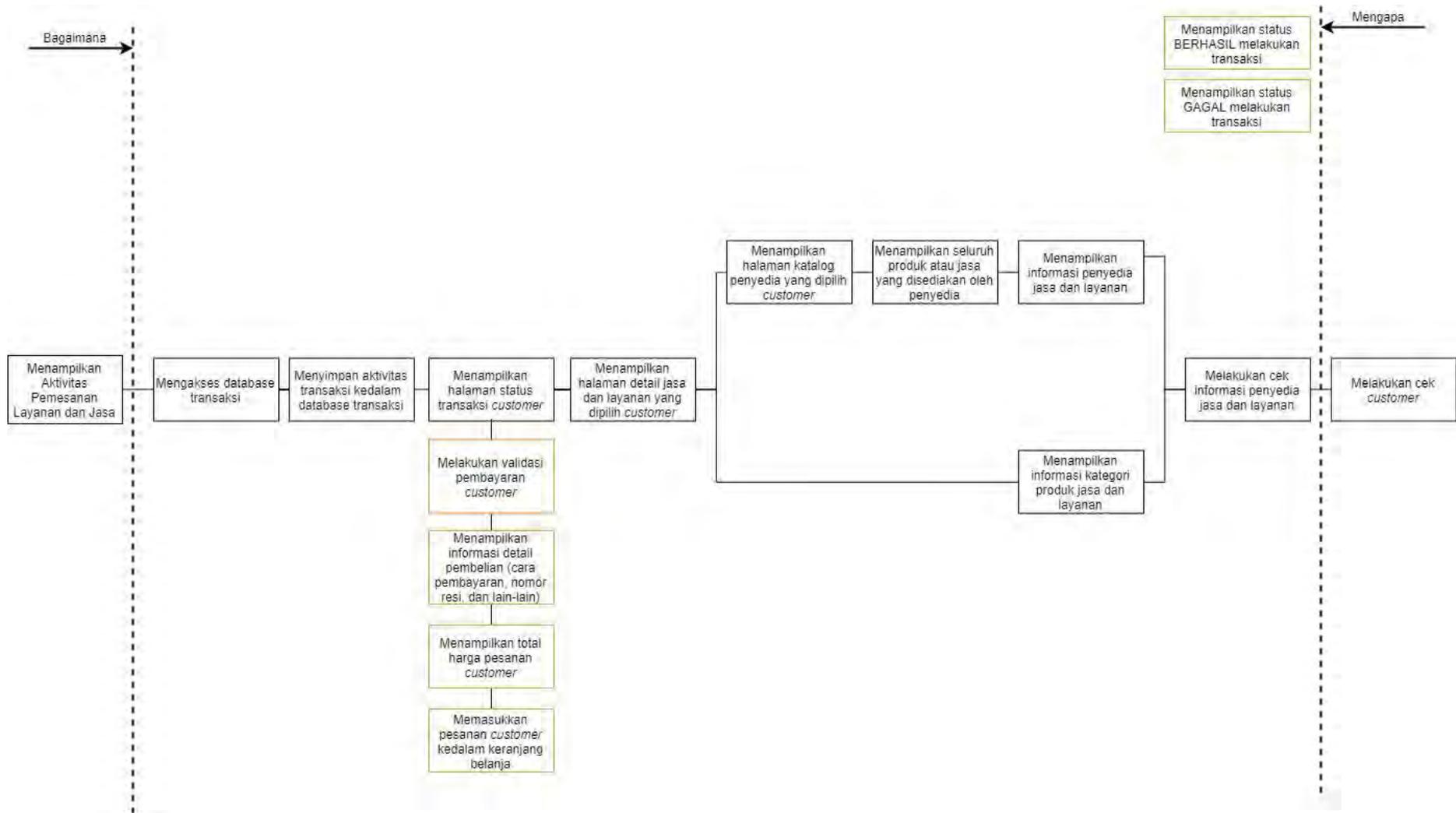
Fungsi ini digunakan ketika *customer* ingin mengetahui lokasi suatu tempat. Terlebih ketika daerah tersebut merupakan daerah yang baru pertama kali dikunjungi



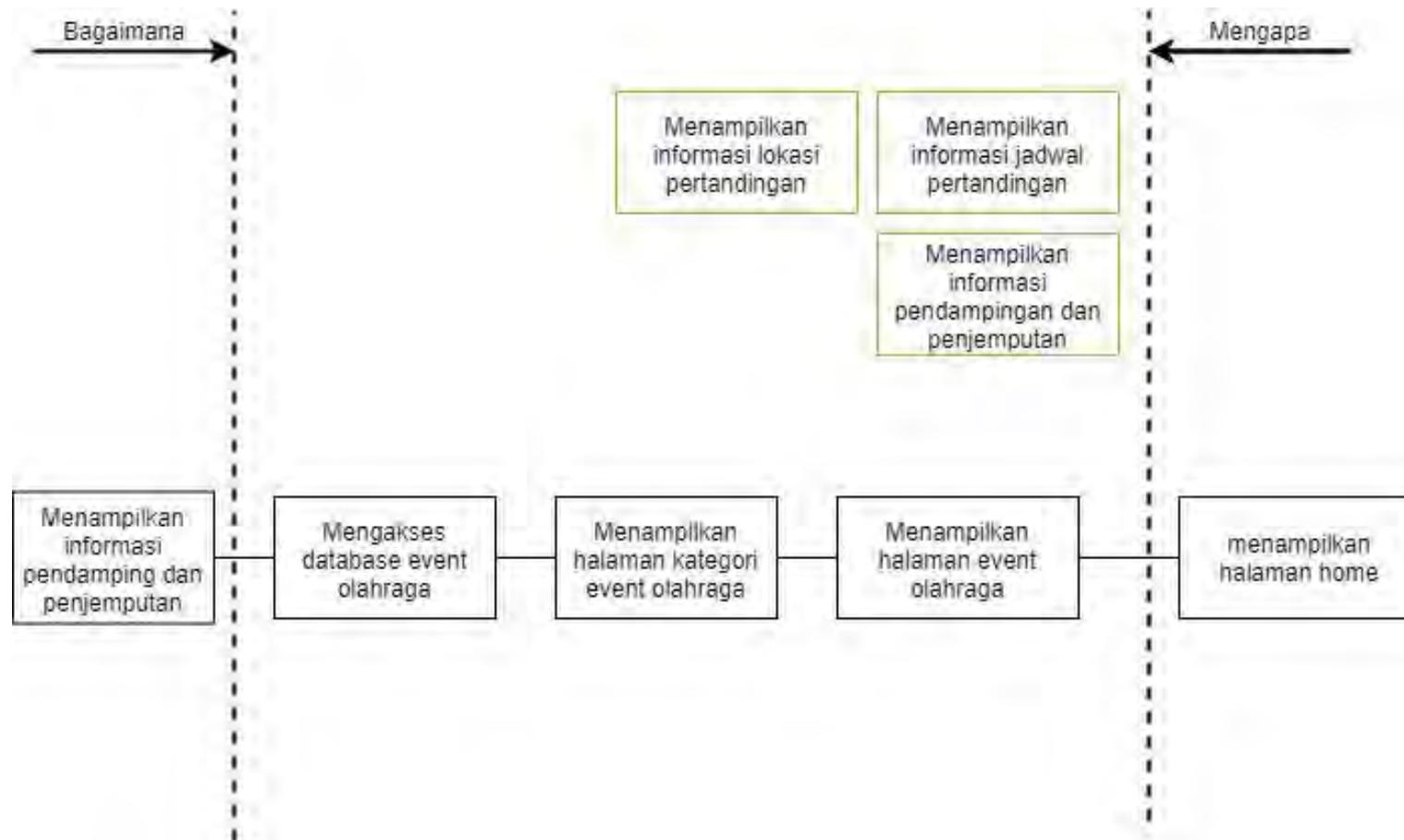
Gambar 6.7 FAST Diagram Menampilkan Informasi Saldo



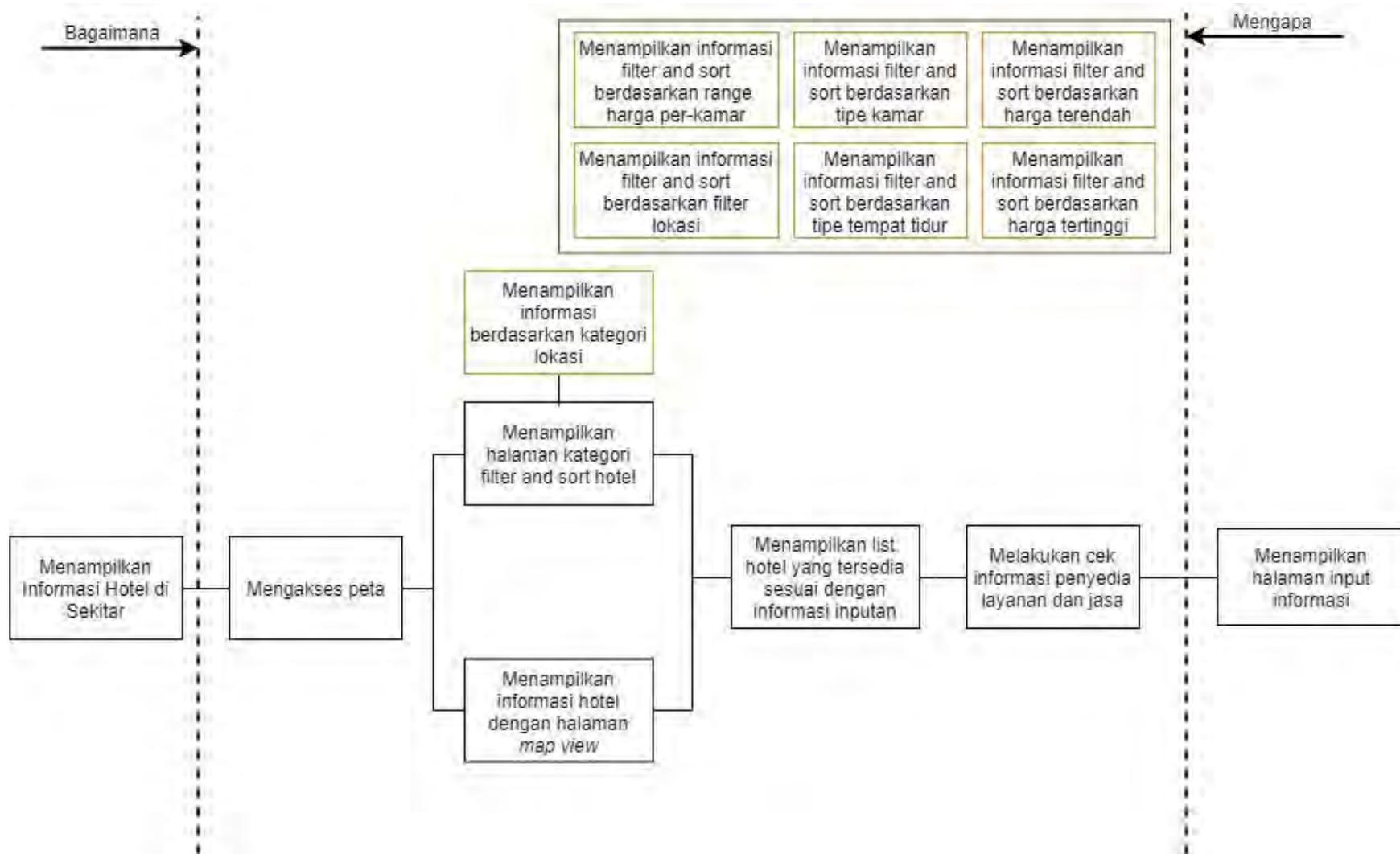
Gambar 6.8 FAST Diagram Menampilkan Aktivitas Transaksi



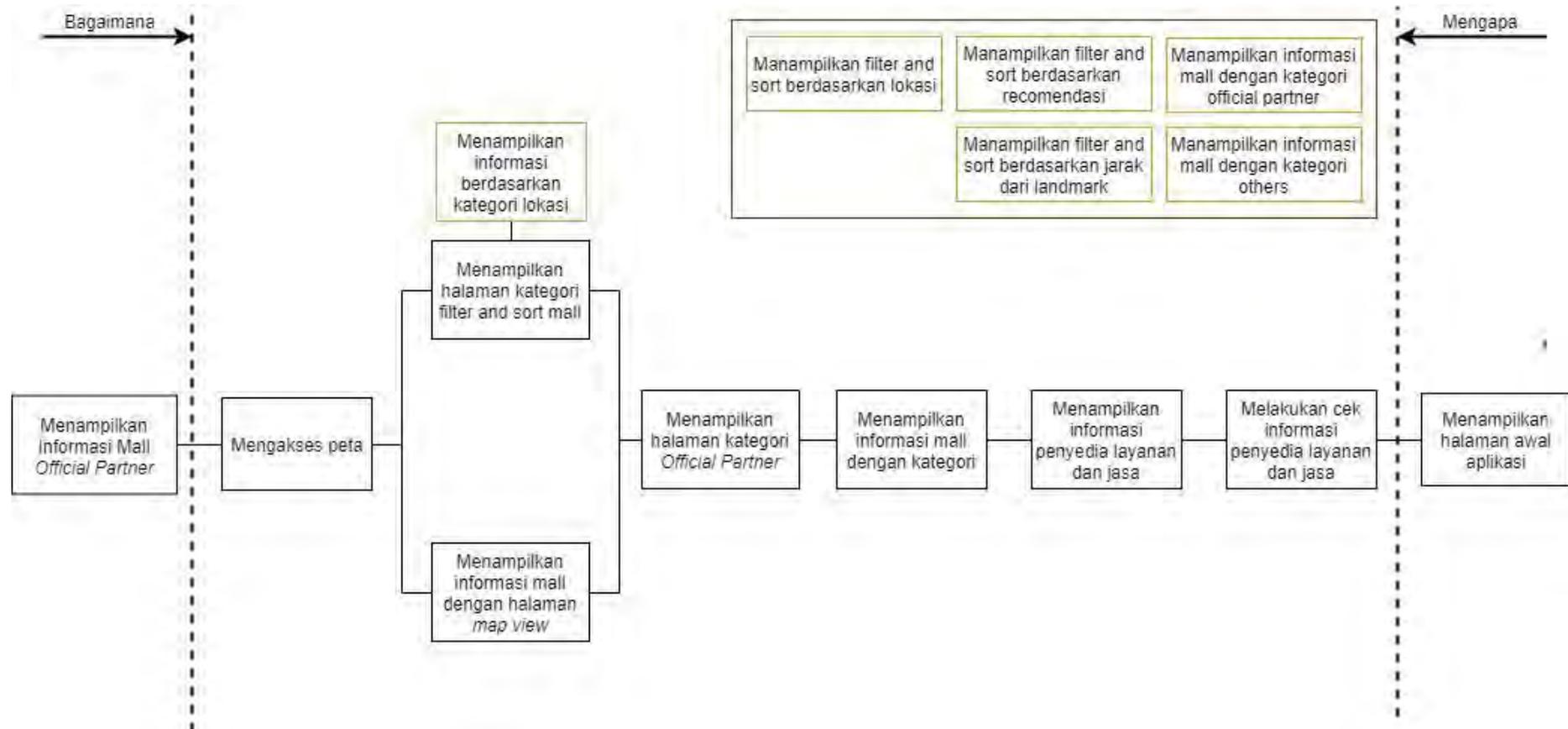
Gambar 6.9 FAST Diagram Menampilkan Aktivitas Pemesanan Layanan dan Jasa



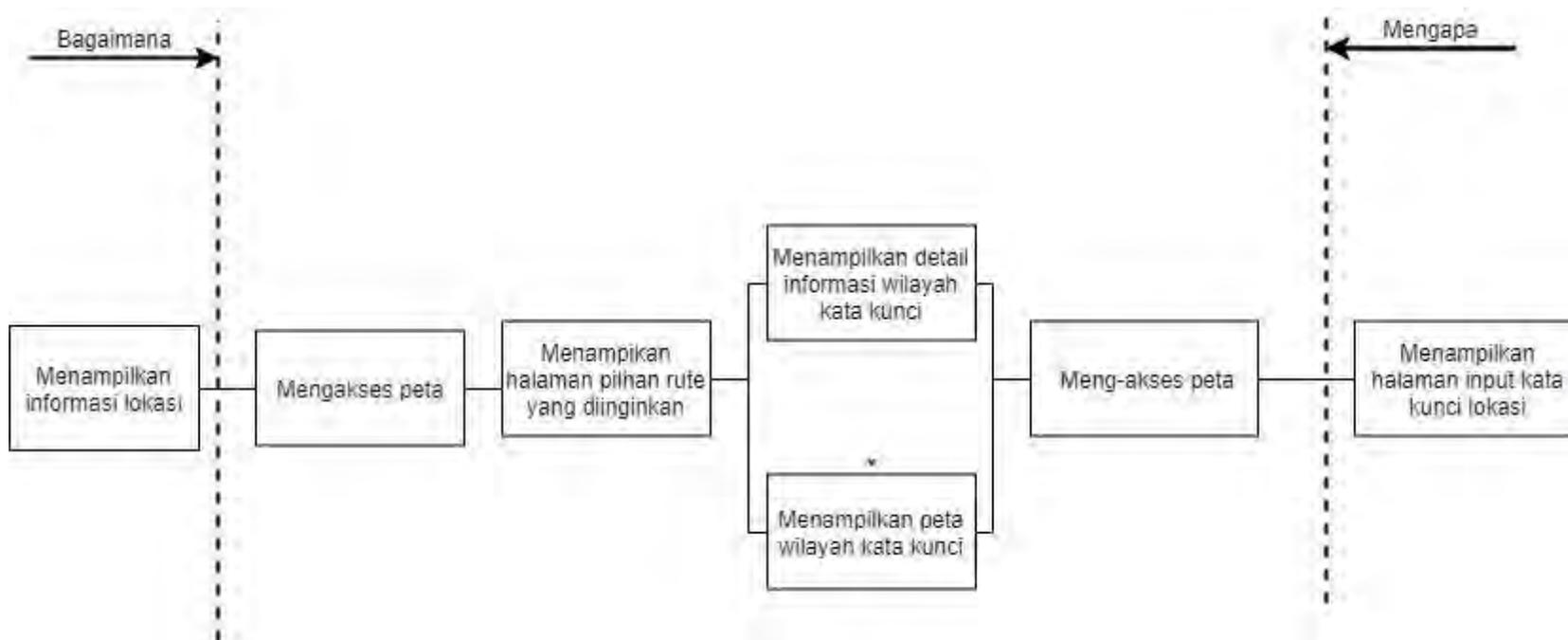
Gambar 6.10 FAST Diagram Menampilkan Informasi Pendamping dan Penjemputan



Gambar 6.11 FAST Diagram Menampilkan Informasi Hotel di Sekitar



Gambar 6.12 FAST Diagram Menampilkan Informasi *Mall Official Partner*



Gambar 6.13 FAST Diagram Menampilkan Informasi Lokasi

*Halaman ini sengaja dikosongkan*

### **6.3 Creativity Phase**

Pada tahapan pengolahan akan dilanjutkan hanya kepada 7 fungsi yang terpilih pada tahap sebelumnya. Tahapan ini akan berisi skenario terkait masing-masing fungsi. Untuk skenario pertama akan melibatkan perhitungan *Function point* dengan mencari *effort* dan waktu pengerjaan masing-masing fitur.

#### **6.3.1 Menampilkan Informasi Saldo**

Jika dilihat dari pembangunan aplikasi secara keseluruhan, jumlah waktu yang diperlukan dalam melakukan aktivitas *software development* adalah 77 hari (616 jam). Sehingga skenario ini bersifat memanfaatkan waktu lembur dan penambahan SDM.

##### **6.3.1.1 Alternatif 1. Memanfaatkan Waktu Lembur**

Dalam menghitung waktu lembur, sama seperti sebelumnya perlu dicari perubahan hari kerja biasa dari 110 hari menjadi 77 hari (616 jam), sehingga hanya akan memerlukan 33 hari waktu lembur (264 jam). Perubahan tersebut seperti pada **Tabel 6.29**.

Tabel 6.29 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Informasi Saldo)

PERUBAHAN HARI KERJA DALAM 5 BULAN										
Aktivitas	Jabatan	Jam Kerja / hari	Hari kerja 110 haro	Total jam kerja 110 hari	Total jam kerja 77 hari	Bobot	Pembulatan (jam)	Hari kerja (hari)	Pembulatan (hari)	Perubahan dalam Jam
<b>Software Development</b>										
<i>Requirement</i>	<i>Business analyst</i>	8	120	880	616	84	84	10.5	10	80
<i>Spesification &amp; Design</i>	<i>System analyst</i>	8	160	880	616	112	112	14	14	112
<i>Coding</i>	<i>Programmer</i>	8	320	880	616	224	224	28	28	224
<i>Testing</i>	<i>Technical tester</i>	8	200	880	616	140	140	17.5	18	144

<b>PERUBAHAN HARI KERJA DALAM 5 BULAN</b>										
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jam Kerja / hari</b>	<b>Hari kerja 110 haro</b>	<b>Total jam kerja 110 hari</b>	<b>Total jam kerja 77 hari</b>	<b>Bobot</b>	<b>Pembulatan (jam)</b>	<b>Hari kerja (hari)</b>	<b>Pembulatan (hari)</b>	<b>Perubahan dalam Jam</b>
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	8	80	880	616	56	56	7	7	56
<b>TOTAL</b>						<b>781,61</b>	<b>780</b>	<b>97.5</b>	<b>100</b>	<b>616</b>

Selanjutnya dilakukan perhitungan biaya pemanfaatan waktu lembur dengan hasil seperti pada **Tabel 6.30**.

Tabel 6.30 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Informasi Saldo)

Kelompok aktivitas	Jabatan	Person	Hours	Standar Gaji (per-jam)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	80	Rp 43.750	Rp 3.500.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	112	Rp 43.750	Rp 4.900.000
<i>Coding</i>	<i>Programmer</i>	2	224	Rp 50.000	Rp 22.400.000
<i>Testing</i>	<i>Technical tester</i>	1	144	Rp 50.000	Rp 7.200.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	56	Rp 93.750	Rp 5.250.000
<b>TOTAL</b>					<b>Rp 43.250.000</b>

Hasil perhitungan biaya hari kerja yang diperoleh adalah Rp 43.250.000, selanjutnya dilakukan perhitungan pemanfaatan waktu lembur. Oleh karena bobot pekerjaan antar jabatan berbeda, sehingga waktu lembur yang mereka miliki juga berbeda. Perbedaan tersebut seperti pada **Tabel 6.31**.

Tabel 6.31 Perhitungan Hari Lembur (Informasi Saldo)

<b>PERHITUNGAN HARI LEMBUR</b>		
<b>Jam kerja di 110 hari</b>	<b>Perubahan jam kerja ke 77 hari</b>	<b>Jam lembur 77 hari</b>
<b><i>Software Development</i></b>		
120	80	40
160	112	48
320	224	96
200	144	56
80	56	24

Berdasarkan **Tabel 6.31**, maka perhitungan estimasi biaya kerja lembur selama 77 hari seperti pada **Tabel 6.32**.

Tabel 6.32 Perhitungan Estimasi Biaya Lembur 77 hari (Informasi Saldo)

Kelompok aktivitas	Jabatan	Person (Orang)	Hours (Jam)	Standar Gaji (per-jam)	Upah/bulan (Rp)	Gaji lembur jam pertama (Rp)	Gaji lembur jam selanjutnya (Rp)
<b>Software Development</b>							
<i>Requirement</i>	<i>Business analyst</i>	1	40	Rp 43.750	7.000.000	Rp 60.694	Rp 3.156.069
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	48	Rp 43.750	7.000.000	Rp 60.694	Rp 3.803.468
<i>Coding</i>	<i>Programmer</i>	2	96	Rp 50.000	8.000.000	Rp 138.728	Rp 17.572.254
<i>Testing</i>	<i>Technical tester</i>	1	56	Rp 50.000	8.000.000	Rp 69.364	Rp 5.086.705
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	24	Rp 93.750	15.000.000	Rp 130.058	Rp 3.988.439
<b>TOTAL</b>						<b>Rp 459.538</b>	<b>Rp 33.606.936</b>

Berdasarkan hasil perhitungan konversi hari kerja, biaya pengerjaan fungsi ini adalah dengan menjumlahkan nilai estimasi biaya + biaya lembur hari kerja. Secara numerik sebagai berikut.

Biaya pembangunan aplikasi :

$$= \text{Rp } 43.250.000 + \text{Rp } 459.538 + \text{Rp } 33.606.936$$

$$= \text{Rp } 77.316.474^*$$

**\*Biaya lebih mahal dari hasil awal.**

### 6.3.1.2 Alternatif 2. Penambahan SDM

Dalam perhitungan sebelumnya, perhitungan terkait login yang

Skenario selanjutnya adalah dengan menambahkan SDM agar waktu pengerjaan mencapai 77 hari dari 110 hari. Hasil penambahan jumlah SDM seperti **Tabel 6.33**.

Tabel 6.33 Penambahan Jumlah SDM (Informasi Saldo)

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
Aktivitas	Jabatan	BEFORE		AFTER	
		Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	15	1	15
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	20	1	20
<i>Coding</i>	<i>Programmer</i>	2	40	4	20
<i>Testing</i>	<i>Technical tester</i>	1	25	2	12
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	10	1	10
<b>TOTAL</b>			<b>110</b>		<b>77</b>

Berdasarkan perubahan jumlah SDM tersebut, maka estimasi biaya pembangunan aplikasi menjadi seperti pada **Tabel 6.34**.

Tabel 6.34 Perhitungan Estimasi Biaya dengan Penambahan SDM

<b>PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM</b>					
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jumlah pekerja (Orang)</b>	<b>Hari Kerja (Hari)</b>	<b>Standar gaji/hari (Rp)</b>	<b>Hasil (Rp)</b>
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	15	350.000	5.250.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	20	350.000	7.000.000
<i>Coding</i>	<i>Programmer</i>	4	20	400.000	32.000.000
<i>Testing</i>	<i>Technical tester</i>	2	12	400.000	9.600.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	10	750.000	7.500.000
<b>TOTAL</b>					<b>61.350.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 61.350.000, lebih murah dari harga awal.

### **6.3.1.3 Alternatif 3. Penggunaan Visual Programming**

Skenario ketiga adalah pemanfaatan visual programming. Dengan adanya visual programming, waktu pengerjaan yang dipegang oleh *programmer* dapat dikurangi. Hasil yang diperoleh seperti pada **Tabel 6.35**.

Tabel 6.35 Penggunaan Visual Programming (Informasi Saldo)

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	15	Rp 350.000	Rp 5.250.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	20	Rp 350.000	Rp 7.000.000
<i>Coding</i>	<i>Programmer</i>	1	11	Rp 400.000	Rp 4.400.000
<i>Testing</i>	<i>Technical tester</i>	1	25	Rp 400.000	Rp 10.000.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	10	Rp 750.000	Rp 7.500.000
<b>TOTAL</b>					<b>Rp 34.150.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 34.150.000, lebih murah dari harga awal.

### 6.3.2 Menampilkan Aktivitas Transaksi

Jika dilihat dari pembangunan aplikasi secara keseluruhan, jumlah waktu yang diperlukan dalam melakukan aktivitas *software development* adalah 77 hari. Sehingga skenario ini bersifat memanfaatkan waktu lembur dan penambahan SDM.

#### 6.3.2.1 Alternatif 1. Memanfaatkan Waktu Lembur

Dalam menghitung waktu lembur, sama seperti sebelumnya perlu dicari perubahan hari kerja biasa dari 106 hari (848 jam) menjadi 77 hari (616 jam), sehingga hanya akan memerlukan 29 hari (232 jam) waktu lembur. Perubahan tersebut seperti pada **Tabel 6.36**.

Tabel 6.36 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Aktivitas Transaksi)

<b>PERUBAHAN HARI KERJA DALAM 5 BULAN</b>										
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jam Kerja / hari</b>	<b>Hari kerja 106 hari</b>	<b>Total jam kerja 106 hari</b>	<b>Total jam kerja 77 hari</b>	<b>Bobot</b>	<b>Pembulatan (jam)</b>	<b>Hari kerja (hari)</b>	<b>Pembulatan (hari)</b>	<b>Pwrubah andalam Jam</b>
<b>Software Development</b>										
<i>Requirement</i>	<i>Business analyst</i>	8	88	848	616	63.92	64	8	8	64
<i>Spesification &amp; Design</i>	<i>System analyst</i>	8	160	848	616	116.23	116	14.5	15	120
<i>Coding</i>	<i>Programmer</i>	8	320	848	616	232.45	232	29	29	232
<i>Testing</i>	<i>Technical tester</i>	8	200	848	616	145.29	145	18.13	18	144
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	8	80	848	616	58.11	58	7.25	7	56
<b>TOTAL</b>						<b>616,00</b>	<b>616</b>	<b>77</b>	<b>77</b>	<b>616</b>

Selanjutnya dilakukan perhitungan biaya pemanfaatan waktu lembur dengan hasil seperti pada **Tabel 6.37**.

Tabel 6.37 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Aktivitas Transaksi)

Kelompok aktivitas	Jabatan	Person	Hours	Standar Gaji (per-jam)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	64	Rp 43.750	Rp 2.800.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	120	Rp 43.750	Rp 5.250.000
<i>Coding</i>	<i>Programmer</i>	2	232	Rp 50.000	Rp 23.200.000
<i>Testing</i>	<i>Technical tester</i>	1	144	Rp 50.000	Rp 7.200.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	56	Rp 93.750	Rp 5.250.000
<b>TOTAL</b>					<b>Rp 43.700.000</b>

Selanjutnya dilakukan perhitungan pemanfaatan waktu lembur. Disebabkan bobot pekerjaan antar jabatan berbeda, sehingga waktu lembur yang mereka miliki juga berbeda. Pembagiannya seperti pada **Tabel 6.38**.

Tabel 6.38 Perhitungan Hari Lembur (Aktivitas Transaksi)

<b>PERHITUNGAN HARI LEMBUR</b>		
<b>Jam kerja di 106 hari</b>	<b>Perubahan jam kerja ke 77 hari</b>	<b>Jam lembur 77 hari</b>
<b>Software Development</b>		
88	64	24
160	120	40
320	232	88

<b>PERHITUNGAN HARI LEMBUR</b>		
<b>Jam kerja di 106 hari</b>	<b>Perubahan jam kerja ke 77 hari</b>	<b>Jam lembur 77 hari</b>
<b><i>Software Development</i></b>		
200	144	56
80	56	24

Berdasarkan **Tabel 6.38**, maka perhitungan estimasi biaya kerja lembur selama 77 hari seperti pada **Tabel 6.39**.

Tabel 6.39 Perhitunga Estimasi Biaya Lembur 77 hari (Aktivitas Transaksi)

Kelompok aktivitas	Jabatan	Person (Orang)	Hours (Jam)	Standar Gaji (per-jam)	Upah/bulan (Rp)	Gaji lembur jam pertama (Rp)	Gaji lembur jam selanjutnya (Rp)
<b>Software Development</b>							
<i>Requirement</i>	<i>Business analyst</i>	1	24	Rp 43.750	7.000.000	Rp 60.694	Rp 1.861.272
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	40	Rp 43.750	7.000.000	Rp 60.694	Rp 3.156.069
<i>Coding</i>	<i>Programmer</i>	2	88	Rp 50.000	8.000.000	Rp 138.728	Rp 16.092.486
<i>Testing</i>	<i>Technical tester</i>	1	56	Rp 50.000	8.000.000	Rp 69.364	Rp 5.086.705
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	24	Rp 93.750	15.000.000	Rp 130.058	Rp 3.988.439
<b>TOTAL</b>						<b>Rp 459.538</b>	<b>Rp 30.184.971</b>

Berdasarkan hasil perhitungan konversi hari kerja, biaya pengerjaan fungsi ini adalah dengan menjumlahkan nilai estimasi biaya + biaya lembur hari kerja. Secara numerik sebagai berikut.

Biaya pembangunan aplikasi :

= Rp 43.700.000+ Rp 459.538+ Rp 30.184.971

= **Rp 74.344.509\***

**\*Biaya lebih mahal dari hasil awal.**

### 6.3.2.2 Alternatif 2. Penambahan SDM

Skenario selanjutnya adalah dengan menambahkan SDM agar waktu pengerjaan mencapai 77 hari dari 110 hari. Hasil penambahan jumlah SDM seperti **Tabel 6.40**.

Tabel 6.40 Penambahan Jumlah SDM (Aktivitas Transaksi)

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
Aktivitas	Jabatan	BEFORE		AFTER	
		Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	11	2	5
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	20	2	10
<i>Coding</i>	<i>Programmer</i>	2	40	2	40
<i>Testing</i>	<i>Technical tester</i>	1	25	2	12
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	10	1	10
<b>TOTAL</b>		<b>110</b>		<b>77</b>	

Berdasarkan perubahan jumlah SDM tersebut, maka estimasi biaya pembangunan aplikasi menjadi seperti pada **Tabel 6.41**.

Tabel 6.41 Perhitungan Estimasi Biaya dengan Penambahan SDM

<b>PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM</b>					
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jumlah pekerja (Orang)</b>	<b>Hari Kerja (Hari)</b>	<b>Standar gaji/hari</b>	<b>Hasil</b>
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	2	5	Rp. 350.000	Rp. 3.500.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	2	10	Rp. 350.000	Rp. 7.000.000
<i>Coding</i>	<i>Programmer</i>	2	40	Rp. 400.000	Rp. 32.000.000
<i>Testing</i>	<i>Technical tester</i>	2	12	Rp. 400.000	Rp. 9.600.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	10	Rp. 750.000	Rp. 7.500.000
<b>TOTAL</b>					<b>Rp. 59.600.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 59.600.000, lebih murah dari harga awal.

### **6.3.2.3 Alternatif 3. Pemanfaatan Visual Programming**

Skenario ketiga adalah pemanfaatan visual programming. Dengan adanya visual programming, waktu pengerjaan yang dipegang oleh *programmer* dapat dikurangi. Hasil yang diperoleh seperti pada **Tabel 6.42**.

Tabel 6.42 Penggunaan Visual Programming (Aktivitas Transaksi)

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	11	Rp 350.000	Rp 5.250.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	20	Rp 350.000	Rp 7.000.000
<i>Coding</i>	<i>Programmer</i>	1	11	Rp 400.000	Rp 4.400.000
<i>Testing</i>	<i>Technical tester</i>	1	25	Rp 400.000	Rp 10.000.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	10	Rp 750.000	Rp 7.500.000
<b>TOTAL</b>					<b>Rp 32.750.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 32.750.000, lebih murah dari harga awal.

### 6.3.3 Menampilkan Aktivitas Pemesanan Layanan dan Jasa

Jika dilihat dari pembangunan aplikasi secara keseluruhan, jumlah waktu yang diperlukan dalam melakukan aktivitas *software development* adalah 77 hari.

#### 6.3.3.1 Alternatif 1. Memanfaatkan Waktu Lembur

Dalam menghitung waktu lembur, sama seperti sebelumnya perlu dicari perubahan hari kerja biasa dari 132 hari (1.056 jam) menjadi 77 hari (616 jam), sehingga memerlukan 55 hari (440 jam) waktu lembur. Perubahan tersebut seperti pada **Tabel 6.43.**

Tabel 6.43 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Pemesanan Layanan dan Jasa)

<b>PERUBAHAN HARI KERJA DALAM 5 BULAN</b>										
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jam Kerja / hari</b>	<b>Hari kerja 132 hari</b>	<b>Total jam kerja 132 hari</b>	<b>Total jam kerja 77 hari</b>	<b>Bobot</b>	<b>Pembulatan (jam)</b>	<b>Hari kerja (hari)</b>	<b>Pembulatan (hari)</b>	<b>Pwrubah andalam Jam</b>
<b>Software Development</b>										
<i>Requirement</i>	<i>Business analyst</i>	8	136	1056	616	79.33	79	9.88	10	80
<i>Spesification &amp; Design</i>	<i>System analyst</i>	8	160	1056	616	93.33	93	11.63	12	96
<i>Coding</i>	<i>Programmer</i>	8	360	1056	616	210	210	26.25	26	208
<i>Testing</i>	<i>Technical tester</i>	8	240	1056	616	140	140	17.5	18	144
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	8	160	1056	616	93.33	93	11.63	12	96
<b>TOTAL</b>						<b>616,00</b>	<b>616</b>	<b>77</b>	<b>77</b>	<b>616</b>

Selanjutnya dilakukan perhitungan biaya pemanfaatan waktu lembur dengan hasil seperti pada **Tabel 6.44**.

Tabel 6.44 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Pemesanan Layanan dan Jasa)

Kelompok aktivitas	Jabatan	Person	Hours	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	2	80	Rp 43.750	Rp 7.000.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	2	96	Rp 43.750	Rp 8.400.000
<i>Coding</i>	<i>Programmer</i>	4	208	Rp 50.000	Rp 41.600.000
<i>Testing</i>	<i>Technical tester</i>	2	144	Rp 50.000	Rp 13.600.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	96	Rp 93.750	Rp 9.000.000
<b>TOTAL</b>					<b>Rp 79.600.000</b>

Selanjutnya dilakukan perhitungan pemanfaatan waktu lembur. Disebabkan bobot pekerjaan antar jabatan berbeda, sehingga waktu lembur yang mereka miliki juga berbeda. Pembagiannya seperti pada **Tabel 6.45**.

Tabel 6.45 Perhitungan Hari Lembur (Pemesanan Layanan dan Jasa)

<b>PERHITUNGAN HARI LEMBUR</b>		
Jam kerja di 132 hari	Perubahan jam kerja ke 77 hari	Jam lembur 77 hari
<b>Software Development</b>		
160	96	56
360	208	64

<b>PERHITUNGAN HARI LEMBUR</b>		
<b>Jam kerja di 132 hari</b>	<b>Perubahan jam kerja ke 77 hari</b>	<b>Jam lembur 77 hari</b>
<b><i>Software Development</i></b>		
240	144	152
160	96	104
160	96	64

Berdasarkan **Tabel 6.45**, maka perhitungan estimasi biaya kerja lembur selama 77 hari seperti pada **Tabel 6.46**.

Tabel 6.46 Perhitungan Estimasi Biaya Lembur 77 hari (Pemesanan Layanan dan Jasa)

Kelompok aktivitas	Jabatan	Person (Orang)	Hours (Jam)	Standar Gaji (per-jam)	Upah/bulan (Rp)	Gaji lembur jam pertama (Rp)	Gaji lembur jam selanjutnya (Rp)
<b>Software Development</b>							
<i>Requirement</i>	<i>Business analyst</i>	2	56	Rp 43750	7.000.000	Rp 121.387	Rp 8.901.734
<i>Spesification &amp; Design</i>	<i>System analyst</i>	2	64	Rp 43750	7.000.000	Rp 121.387	Rp 10.196532
<i>Coding</i>	<i>Programmer</i>	4	152	Rp 50000	8.000.000	Rp 277.457	Rp 55.861.272
<i>Testing</i>	<i>Technical tester</i>	2	104	Rp 50000	8.000.000	Rp 138.728	Rp 19.05.023
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	64	Rp 93750	15.000.000	Rp 130.058	Rp 10.924.855
<b>TOTAL</b>						<b>Rp 789.017</b>	<b>Rp 104.936.416</b>

Berdasarkan hasil perhitungan konversi hari kerja, biaya pengerjaan fungsi ini adalah dengan menjumlahkan nilai estimasi biaya + biaya lembur hari kerja. Secara numerik sebagai berikut.

Biaya pembangunan aplikasi :

$$= \text{Rp } 79.600.000 + \text{Rp } 789.017 + \text{Rp } 104.936.416$$

$$= \text{Rp } 185.325.433^*$$

**\*Biaya lebih mahal dari hasil awal.**

### 6.3.3.2 Alternatif 2. Penambahan SDM versi 1

Berdasarkan perhitungan skenario sebelumnya, nilai dengan memanfaatkan waktu lembur akan lebih tinggi dari nilai awal, sehingga skenario ini bersifat memanfaatkan penambahan SDM. Hasil penambahan jumlah SDM seperti **Tabel 6.47**.

Tabel 6.47 Penambahan Jumlah SDM (Pemesanan Layanan dan Jasa)

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
		BEFORE		AFTER	
Aktivitas	Jabatan	Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	2	17	4	8
<i>Spesification &amp; Design</i>	<i>System analyst</i>	2	20	5	8
<i>Coding</i>	<i>Programmer</i>	4	45	5	36
<i>Testing</i>	<i>Technical tester</i>	2	30	4	15
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	20	2	10
<b>TOTAL</b>		<b>110</b>		<b>77</b>	

Berdasarkan perubahan jumlah SDM tersebut, maka estimasi biaya pembangunan aplikasi menjadi seperti pada **Tabel 6.48**.

Tabel 6.48 Perhitungan Estimasi Biaya dengan Penambahan SDM

<b>PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM</b>					
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jumlah pekerja (Orang)</b>	<b>Hari Kerja (Hari)</b>	<b>Standar gaji/hari (Rp)</b>	<b>Hasil (Rp)</b>
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	4	8	350.000	11.200.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	5	8	350.000	14.000.000
<i>Coding</i>	<i>Programmer</i>	5	36	400.000	72.000.000
<i>Testing</i>	<i>Technical tester</i>	4	15	400.000	24.000.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	2	10	750.000	15.000.000
<b>TOTAL</b>					<b>136.200.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 136.200.000, lebih murah dari harga awal.

### 6.3.3.3 Alternatif 3. Penambahan SDM versi 2

Hasil penambahan jumlah SDM berikutnya seperti **Tabel 6.49**.

Tabel 6.49 Penambahan Jumlah SDM (Pemesanan Layanan dan Jasa)

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
Aktivitas	Jabatan	BEFORE		AFTER	
		Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	2	17	6	5
<i>Spesification &amp; Design</i>	<i>System analyst</i>	2	20	6	7
<i>Coding</i>	<i>Programmer</i>	4	45	7	25
<i>Testing</i>	<i>Technical tester</i>	2	30	2	30
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	20	2	10
<b>TOTAL</b>		<b>110</b>		<b>77</b>	

Berdasarkan perubahan jumlah SDM tersebut, maka estimasi biaya pembangunan aplikasi menjadi seperti pada **Tabel 6.50**.

Tabel 6.50 Perhitungan Estimasi Biaya dengan Penambahan SDM

PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM					
Aktivitas	Jabatan	Jumlah pekerja (Orang)	Hari Kerja (Hari)	Standar gaji/hari (Rp)	Hasil (Rp)
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	6	5	350.000	10.500.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	6	7	350.000	14.700.000
<i>Coding</i>	<i>Programmer</i>	7	25	400.000	70.000.000

PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM					
Aktivitas	Jabatan	Jumlah pekerja (Orang)	Hari Kerja (Hari)	Standar gaji/hari (Rp)	Hasil (Rp)
<i>Testing</i>	<i>Technical tester</i>	2	30	400.000	24.000.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	2	10	750.000	15.000.000
<b>TOTAL</b>					<b>134.200.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 134.200.000, lebih murah dari harga awal.

#### 6.3.3.4 Alternatif 4. Pemanfaatan Visual Programming

Skenario selanjutnya adalah pemanfaatan visual programming. Dengan adanya visual programming, waktu pengerjaan yang dipegang oleh *programmer* dapat dikurangi. Hasil yang diperoleh seperti pada **Tabel 6.51**.

Tabel 6.51 Penggunaan Visual Programming (Pemesanan Layanan dan Jasa)

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	2	17	Rp 350.000	Rp 11.900.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	2	20	Rp 350.000	Rp 14.000.000
<i>Coding</i>	<i>Programmer</i>	1	11	Rp 400.000	Rp 4.400.000
<i>Testing</i>	<i>Technical tester</i>	2	30	Rp 400.000	Rp 24.000.000

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	20	Rp 750.000	Rp 15.000.000
<b>TOTAL</b>					<b>Rp 69.300.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 69.300.000, lebih murah dari harga awal.

### 6.3.4 Menampilkan Informasi Pendamping

Jika dilihat dari pembangunan aplikasi secara keseluruhan, jumlah waktu yang diperlukan dalam melakukan aktivitas *software development* adalah 77 hari.

#### 6.3.4.1 Alternatif 1. Memanfaatkan Waktu Lembur

Dalam menghitung waktu lembur, sama seperti sebelumnya perlu dicari perubahan hari kerja biasa dari 90 hari (720 jam) menjadi 77 hari, sehingga memerlukan 13 hari (104 jam) waktu lembur. Perubahan tersebut seperti pada **Tabel 6.52**.

Tabel 6.52 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Informasi Pendamping)

<b>PERUBAHAN HARI KERJA DALAM 5 BULAN</b>										
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jam Kerja / hari</b>	<b>Hari kerja 90 hari</b>	<b>Total jam kerja 90 hari</b>	<b>Total jam kerja 77 hari</b>	<b>Bobot</b>	<b>Pembulatan (jam)</b>	<b>Hari kerja (hari)</b>	<b>Pembulatan (hari)</b>	<b>Pwrubah andalam Jam</b>
<b>Software Development</b>										
<i>Requirement</i>	<i>Business analyst</i>	8	120	720	616	102.67	103	12.88	13	104
<i>Spesification &amp; Design</i>	<i>System analyst</i>	8	120	720	616	102.67	103	12.88	13	104
<i>Coding</i>	<i>Programmer</i>	8	360	720	616	308.00	308	38.5	38	304
<i>Testing</i>	<i>Technical tester</i>	8	120	720	616	102.67	103	12.88	13	104
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	8	0	720	616	0	0	0	0	0
<b>TOTAL</b>						<b>616,00</b>	<b>617</b>	<b>77</b>	<b>77</b>	<b>616</b>

Selanjutnya dilakukan perhitungan biaya pemanfaatan waktu lembur dengan hasil seperti pada **Tabel 6.53**.

Tabel 6.53 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Informasi Pendamping)

Kelompok aktivitas	Jabatan	Person	Hours	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	104	Rp 43.750	Rp 4.550.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	104	Rp 43.750	Rp 4.550.000
<i>Coding</i>	<i>Programmer</i>	1	304	Rp 50.000	Rp 15.200.000
<i>Testing</i>	<i>Technical tester</i>	1	104	Rp 50.000	Rp 4.550.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	0	Rp 93.750	Rp 0
<b>TOTAL</b>					<b>Rp 29.500.000</b>

Selanjutnya dilakukan perhitungan pemanfaatan waktu lembur. Disebabkan bobot pekerjaan antar jabatan berbeda, sehingga waktu lembur yang mereka miliki juga berbeda. Pembagiannya seperti pada **Tabel 6.54**.

Tabel 6.54 Perhitungan Hari Lembur (Informasi Pendamping)

<b>PERHITUNGAN HARI LEMBUR</b>		
<b>Jam kerja di 90 hari</b>	<b>Perubahan jam kerja ke 77 hari</b>	<b>Jam lembur 77 hari</b>
<b>Software Development</b>		
120	104	16
120	104	16
360	304	56

<b>PERHITUNGAN HARI LEMBUR</b>		
<b>Jam kerja di 90 hari</b>	<b>Perubahan jam kerja ke 77 hari</b>	<b>Jam lembur 77 hari</b>
<b><i>Software Development</i></b>		
120	104	16
0	0	0

Berdasarkan **Tabel 6.54.** maka perhitungan estimasi biaya kerja lembur selama 77 hari seperti pada **Tabel 6.55.**

Tabel 6.55 Perhitunga Estimasi Biaya Lembur 77 hari (Informasi Hotel di Sekitar)

Kelompok aktivitas	Jabatan	Person (Orang)	Hours (Jam)	Standar Gaji (per-jam)	Upah/bulan (Rp)	Gaji lembur jam pertama (Rp)	Gaji lembur jam selanjutnya (Rp)
<b>Software Development</b>							
<i>Requirement</i>	<i>Business analyst</i>	1	16	Rp 43750	7.000.000	Rp 60.694	Rp 1.213.873
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	16	Rp 43750	7.000.000	Rp 60.694	Rp 1.213.873
<i>Coding</i>	<i>Programmer</i>	1	56	Rp 50000	8.000.000	Rp 69.364	Rp 5.086.705
<i>Testing</i>	<i>Technical tester</i>	1	16	Rp 50000	8.000.000	Rp 69.364	Rp 1.387.283
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	0	Rp 93750	15.000.000	Rp 0	
<b>TOTAL</b>						<b>Rp 260.116</b>	<b>Rp 8.901.734</b>

Berdasarkan hasil perhitungan konversi hari kerja, biaya pengerjaan fungsi ini adalah dengan menjumlahkan nilai estimasi biaya + biaya lembur hari kerja. Secara numerik sebagai berikut.

Biaya pembangunan aplikasi :

$$= \text{Rp } 29.500.000 + \text{Rp } 260.116 + \text{Rp } 8.901.734$$

$$= \text{Rp } 38.661.850^*$$

**\*Biaya lebih mahal dari hasil awal.**

### 6.3.4.2 Alternatif 2. Penambahan SDM

Berdasarkan perhitungan skenario sebelumnya, nilai dengan memanfaatkan waktu lembur akan lebih tinggi dari nilai awal, sehingga skenario ini bersifat memanfaatkan penambahan SDM. Hasil penambahan jumlah SDM seperti **Tabel 6.56**.

Tabel 6.56 Penambahan Jumlah SDM (Informasi Pendamping)

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
Aktivitas	Jabatan	BEFORE		AFTER	
		Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	15	3	5
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	15	1	12
<i>Coding</i>	<i>Programmer</i>	1	45	1	45
<i>Testing</i>	<i>Technical tester</i>	1	15	1	15
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	0	0	0
<b>TOTAL</b>			<b>90</b>	<b>77</b>	

Berdasarkan perubahan jumlah SDM tersebut, maka estimasi biaya pembangunan aplikasi menjadi seperti pada **Tabel 6.57**.

Tabel 6.57 Perhitungan Estimasi Biaya dengan Penambahan SDM

<b>PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM</b>					
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jumlah pekerja (Orang)</b>	<b>Hari Kerja (Hari)</b>	<b>Standar gaji/hari (Rp)</b>	<b>Hasil (Rp)</b>
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	3	5	350.000	5.250.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	12	350.000	4.200.000
<i>Coding</i>	<i>Programmer</i>	1	45	400.000	18.000.000
<i>Testing</i>	<i>Technical tester</i>	1	15	400.000	6.000.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	0	750.000	0
<b>TOTAL</b>					<b>33.450.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 33.450.000, lebih murah dari harga awal.

### **6.3.4.3 Alternatif 3. Pemanfaatan Visual Programming**

Skenario ketiga adalah pemanfaatan visual programming. Dengan adanya visual programming, waktu pengerjaan yang dipegang oleh *programmer* dapat dikurangi. Hasil yang diperoleh seperti pada **Tabel 6.58**.

Tabel 6.58 Penggunaan Visual Programming (Informasi Pendamping)

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	15	Rp 350.000	Rp 5.250.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	15	Rp 350.000	Rp 5.250.000
<i>Coding</i>	<i>Programmer</i>	1	11	Rp 400.000	Rp 4.400.000
<i>Testing</i>	<i>Technical tester</i>	1	15	Rp 400.000	Rp 6.000.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	0	Rp 750.000	Rp 0
<b>TOTAL</b>					<b>Rp 20.900.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 20.900.000, lebih murah dari harga awal.

### 6.3.5 Menampilkan Informasi Hotel di Sekitar

Jika dilihat dari pembangunan aplikasi secara keseluruhan, jumlah waktu yang diperlukan dalam melakukan aktivitas *software development* adalah 77 hari.

#### 6.3.5.1 Alternatif 1. Memanfaatkan Waktu Lembur

Dalam menghitung waktu lembur, sama seperti sebelumnya perlu dicari perubahan hari kerja biasa dari 100 hari (800 jam) menjadi 77 hari, sehingga memerlukan 23 hari (184 jam) waktu lembur. Perubahan tersebut seperti pada **Tabel 6.59**.

Tabel 6. 59 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Informasi Hotel di Sekitar)

<b>PERUBAHAN HARI KERJA DALAM 5 BULAN</b>										
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jam Kerja / hari</b>	<b>Hari kerja 100 hari</b>	<b>Total jam kerja 100 hari</b>	<b>Total jam kerja 77 hari</b>	<b>Bobot</b>	<b>Pembulatan (jam)</b>	<b>Hari kerja (hari)</b>	<b>Pembulatan (hari)</b>	<b>Pwrubah andalam Jam</b>
<b>Software Development</b>										
<i>Requirement</i>	<i>Business analyst</i>	8	120	800	616	92.40	92	11.5	12	96
<i>Spesification &amp; Design</i>	<i>System analyst</i>	8	120	800	616	92.40	92	11.5	12	96
<i>Coding</i>	<i>Programmer</i>	8	280	800	616	215.60	216	27	27	216
<i>Testing</i>	<i>Technical tester</i>	8	160	800	616	123.40	123	15.38	15	120
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	8	120	800	616	92.40	93	11.63	11	88
<b>TOTAL</b>						<b>616,00</b>	<b>616</b>	<b>77</b>	<b>77</b>	<b>616</b>

Selanjutnya dilakukan perhitungan biaya pemanfaatan waktu lembur dengan hasil seperti pada **Tabel 6.60**.

Tabel 6.60 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Informasi Hotel di Sekitar)

Kelompok aktivitas	Jabatan	Person	Hours	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	96	Rp 43.750	Rp 4.200.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	96	Rp 43.750	Rp 4.200.000
<i>Coding</i>	<i>Programmer</i>	1	216	Rp 50.000	Rp 10.800.000
<i>Testing</i>	<i>Technical tester</i>	1	120	Rp 50.000	Rp 6.000.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	88	Rp 93.750	Rp 8.250.000
<b>TOTAL</b>					<b>Rp 33.450.000</b>

Selanjutnya dilakukan perhitungan pemanfaatan waktu lembur. Disebabkan bobot pekerjaan antar jabatan berbeda, sehingga waktu lembur yang mereka miliki juga berbeda. Pembagiannya seperti pada **Tabel 6.61**.

Tabel 6.61 Perhitungan Hari Lembur (Informasi Hotel di Sekitar)

<b>PERHITUNGAN HARI LEMBUR</b>		
<b>Jam kerja di 100 hari</b>	<b>Perubahan jam kerja ke 77 hari</b>	<b>Jam lembur 77 hari</b>
<b>Software Development</b>		
120	96	24
120	96	24

<b>PERHITUNGAN HARI LEMBUR</b>		
<b>Jam kerja di 100 hari</b>	<b>Perubahan jam kerja ke 77 hari</b>	<b>Jam lembur 77 hari</b>
<b><i>Software Development</i></b>		
280	216	64
160	120	40
120	88	32

Berdasarkan **Tabel 6.61**, maka perhitungan estimasi biaya kerja lembur selama 77 hari seperti pada **Tabel 6.62**.

Tabel 6.62 Perhitungan Estimasi Biaya Lembur 77 hari (Informasi Hotel di Sekitar)

Kelompok aktivitas	Jabatan	Person (Orang)	Hours (Jam)	Standar Gaji (per-jam)	Upah/bulan (Rp)	Gaji lembur jam pertama (Rp)	Gaji lembur jam selanjutnya (Rp)
<b>Software Development</b>							
<i>Requirement</i>	<i>Business analyst</i>	1	24	Rp 43750	7.000.000	Rp 60.694	Rp 1.861.272
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	24	Rp 43750	7.000.000	Rp 60.694	Rp 1.861.272
<i>Coding</i>	<i>Programmer</i>	1	64	Rp 50000	8.000.000	Rp 69.364	Rp 5.826.590
<i>Testing</i>	<i>Technical tester</i>	1	40	Rp 50000	8.000.000	Rp 69.364	Rp 3.606.936
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	32	Rp 93750	15.000.000	Rp 130.057	Rp 5.375.723
<b>TOTAL</b>						<b>Rp 390.173</b>	<b>Rp 18.531.791</b>

Berdasarkan hasil perhitungan konversi hari kerja, biaya pengerjaan fungsi ini adalah dengan menjumlahkan nilai estimasi biaya + biaya lembur hari kerja. Secara numerik sebagai berikut.

Biaya pembangunan aplikasi :

$$= \text{Rp } 33.450.000 + \text{Rp } 390.173 + \text{Rp } 18.531.791$$

$$= \text{Rp } 52.371.964^*$$

**\*Biaya lebih mahal dari hasil awal.**

### 6.3.5.2 Alternatif 2. Penambahan SDM

Berdasarkan perhitungan skenario sebelumnya, nilai dengan memanfaatkan waktu lembur akan lebih tinggi dari nilai awal, sehingga skenario ini bersifat memanfaatkan penambahan SDM. Hasil penambahan jumlah SDM seperti **Tabel 6.63**.

Tabel 6.63 Penambahan Jumlah SDM (Menampilkan informasi hotel di sekitar)

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
Aktivitas	Jabatan	BEFORE		AFTER	
		Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	1	15	1	15
Spesification & Design	<i>System analyst</i>	1	15	1	15
Coding	<i>Programmer</i>	1	35	2	14
Testing	<i>Technical tester</i>	1	20	1	20
Acceptance and Deployment	<i>Technical Team leader</i>	1	15	1	15
<b>TOTAL</b>		<b>110</b>		<b>77</b>	

Berdasarkan perubahan jumlah SDM tersebut, maka estimasi biaya pembangunan aplikasi menjadi seperti pada **Tabel 6.64**.

Tabel 6.64 Perhitungan Estimasi Biaya dengan Penambahan SDM

<b>PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM</b>					
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jumlah pekerja (Orang)</b>	<b>Hari Kerja (Hari)</b>	<b>Standar gaji/hari (Rp)</b>	<b>Hasil (Rp)</b>
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	1	14	350.000	4.900.000
Spesification & Design	<i>System analyst</i>	1	14	350.000	4.900.000
Coding	<i>Programmer</i>	2	14	400.000	11.200.000
Testing	<i>Technical tester</i>	1	20	400.000	8.000.000
Acceptance and Deployment	<i>Technical Team leader</i>	1	15	750.000	11.250.000
<b>TOTAL</b>					<b>40.250.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 40.250.000, lebih murah dari harga awal.

### 6.3.5.3 Alternatif 3. Memanfaatkan Visual Programming

Skenario ketiga adalah pemanfaatan visual programming. Dengan adanya visual programming, waktu pengerjaan yang dipegang oleh *programmer* dapat dikurangi. Hasil yang diperoleh seperti pada **Tabel 6.65**.

Tabel 6.65 Penggunaan Visual Programming (Informasi Hotel di Sekitar)

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
Requirement	<i>Business analyst</i>	1	15	Rp 350.000	Rp 5.250.000
Spesification & Design	<i>System analyst</i>	1	15	Rp 350.000	Rp 5.250.000
Coding	<i>Programmer</i>	1	11	Rp 400.000	Rp 4.400.000
Testing	<i>Technical tester</i>	1	20	Rp 400.000	Rp 8.000.000
Acceptance and Deployment	<i>Technical Team leader</i>	1	15	Rp 750.000	Rp 11.250.000
<b>TOTAL</b>					<b>Rp 34.150.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 34.150.000, lebih murah dari harga awal.

### 6.3.6 Menampilkan Informasi Mall *Official Partner* di Sekitar

Jika dilihat dari pembangunan aplikasi secara keseluruhan, jumlah waktu yang diperlukan dalam melakukan aktivitas *software development* adalah 77 hari.

#### 6.3.6.1 Alternatif 1. Memanfaatkan Waktu Lembur

Dalam menghitung waktu lembur, sama seperti sebelumnya perlu dicari perubahan hari kerja biasa dari 100 hari (800 jam) menjadi 77 hari, sehingga memerlukan 23 hari (184 jam) waktu lembur. Perubahan tersebut seperti pada **Tabel 6.66**.

Tabel 6.66 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Informasi *Mall Official Partner* di Sekitar)

PERUBAHAN HARI KERJA DALAM 5 BULAN										
Aktivitas	Jabatan	Jam Kerja / hari	Hari kerja 100 hari	Total jam kerja 100 hari	Total jam kerja 77 hari	Bobot	Pembulatan (jam)	Hari kerja (hari)	Pembulatan (hari)	Pwrubah andalam Jam
<b>Software Development</b>										
<i>Requirement</i>	<i>Business analyst</i>	8	120	800	616	92.40	92	11.5	12	96
<i>Spesification &amp; Design</i>	<i>System analyst</i>	8	120	800	616	92.40	92	11.5	12	96
<i>Coding</i>	<i>Programmer</i>	8	280	800	616	215.60	216	27	27	216
<i>Testing</i>	<i>Technical tester</i>	8	160	800	616	123.40	123	15.38	15	120
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	8	120	800	616	92.40	93	11.63	11	88
<b>TOTAL</b>						<b>616,00</b>	<b>616</b>	<b>77</b>	<b>77</b>	<b>616</b>

Selanjutnya dilakukan perhitungan biaya pemanfaatan waktu lembur dengan hasil seperti pada **Tabel 6.67**.

Tabel 6.67 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Informasi *Mall Official Partner* di Sekitar)

Kelompok aktivitas	Jabatan	Person	Hours	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	96	Rp 43.750	Rp 4.200.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	96	Rp 43.750	Rp 4.200.000
<i>Coding</i>	<i>Programmer</i>	1	216	Rp 50.000	Rp 10.800.000
<i>Testing</i>	<i>Technical tester</i>	1	120	Rp 50.000	Rp 6.000.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	88	Rp 93.750	Rp 8.250.000
<b>TOTAL</b>					<b>Rp 33.450.000</b>

Selanjutnya dilakukan perhitungan pemanfaatan waktu lembur. Disebabkan bobot pekerjaan antar jabatan berbeda, sehingga waktu lembur yang mereka miliki juga berbeda. Pembagiannya seperti pada **Tabel 6.68**.

Tabel 6.68 Perhitungan Hari Lembur (Informasi *Mall Official Partner* di Sekitar)

<b>PERHITUNGAN HARI LEMBUR</b>		
Jam kerja di 100 hari	Perubahan jam kerja ke 77 hari	Jam lembur 77 hari
<b>Software Development</b>		
120	96	24

<b>PERHITUNGAN HARI LEMBUR</b>		
<b>Jam kerja di 100 hari</b>	<b>Perubahan jam kerja ke 77 hari</b>	<b>Jam lembur 77 hari</b>
<b><i>Software Development</i></b>		
120	96	24
280	216	64
160	120	40
120	88	32

Berdasarkan **Tabel 6.68**, maka perhitungan estimasi biaya kerja lembur selama 77 hari seperti pada **Tabel 6.69**.

Tabel 6.69 Perhitunga Estimasi Biaya Lembur 77 hari (Informasi *Mall Official Partner* di Sekitar)

Kelompok aktivitas	Jabatan	Person (Orang)	Hours (Jam)	Standar Gaji (per-jam)	Upah/bulan (Rp)	Gaji lembur hari pertama (Rp)	Gaji lembur hari selanjutnya (Rp)
<b><i>Software Development</i></b>							
<i>Requirement</i>	<i>Business analyst</i>	1	24	Rp 43750	7.000.000	Rp 60.694	Rp 1.861.272
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	24	Rp 43750	7.000.000	Rp 60.694	Rp 1.861.272
<i>Coding</i>	<i>Programmer</i>	1	64	Rp 50000	8.000.000	Rp 69.364	Rp 5.826.590
<i>Testing</i>	<i>Technical tester</i>	1	40	Rp 50000	8.000.000	Rp 69.364	Rp 3.606.936
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	32	Rp 93750	15.000.000	Rp 130.057	Rp 5.375.723
<b>TOTAL</b>						<b>Rp 390.173</b>	<b>Rp 18.531.791</b>

Berdasarkan hasil perhitungan konversi hari kerja, biaya pengerjaan fungsi ini adalah dengan menjumlahkan nilai estimasi biaya + biaya lembur hari kerja. Secara numerik sebagai berikut.

Biaya pembangunan aplikasi :

$$= \text{Rp } 33.450.000 + \text{Rp } 390.173 + \text{Rp } 18.531.791$$

$$= \text{Rp } 52.371.964^*$$

**\*Biaya lebih mahal dari hasil awal.**

### 6.3.6.2 Alternatif 2. Penambahan SDM

Berdasarkan perhitungan skenario sebelumnya, nilai dengan memanfaatkan waktu lembur akan lebih tinggi dari nilai awal, sehingga skenario ini bersifat memanfaatkan penambahan SDM. Hasil penambahan jumlah SDM seperti **Tabel 6.70**.

Tabel 6.70 Penambahan Jumlah SDM (Informasi *Mall Official Partner* di Sekitar)

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
Aktivitas	Jabatan	BEFORE		AFTER	
		Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	15	1	15
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	15	1	15
<i>Coding</i>	<i>Programmer</i>	1	35	2	14
<i>Testing</i>	<i>Technical tester</i>	1	20	1	20
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	15	1	15
<b>TOTAL</b>		<b>110</b>		<b>77</b>	

Berdasarkan perubahan jumlah SDM tersebut, maka estimasi biaya pembangunan aplikasi menjadi seperti pada **Tabel 6.71**.

Tabel 6.71 Perhitungan Estimasi Biaya dengan Penambahan SDM

<b>PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM</b>					
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jumlah pekerja (Orang)</b>	<b>Hari Kerja (Hari)</b>	<b>Standar gaji/hari (Rp)</b>	<b>Hasil (Rp)</b>
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	14	350.000	4.900.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	14	350.000	4.900.000
<i>Coding</i>	<i>Programmer</i>	2	14	400.000	11.200.000
<i>Testing</i>	<i>Technical tester</i>	1	20	400.000	8.000.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	15	750.000	11.250.000
<b>TOTAL</b>					<b>40.250.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 40.250.000, lebih murah dari harga awal.

### **6.3.6.3 Alternatif 3. Memanfaatkan Visual Programming**

Skenario ketiga adalah pemanfaatan visual programming. Dengan adanya visual programming, waktu pengerjaan yang dipegang oleh *programmer* dapat dikurangi. Hasil yang diperoleh seperti pada **Tabel 6.72**.

Tabel 6.72 Penggunaan Visual Programming (Informasi *Mall Official Partner* di Sekitar)

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	15	Rp 350.000	Rp 5.250.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	1	15	Rp 350.000	Rp 5.250.000
<i>Coding</i>	<i>Programmer</i>	1	11	Rp 400.000	Rp 4.400.000
<i>Testing</i>	<i>Technical tester</i>	1	20	Rp 400.000	Rp 8.000.000
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	1	15	Rp 750.000	Rp 11.250.000
<b>TOTAL</b>					<b>Rp 34.150.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 34.150.000, lebih murah dari harga awal.

### 6.3.7 Menampilkan Informasi Lokasi

Jika dilihat dari pembangunan aplikasi secara keseluruhan, jumlah waktu yang diperlukan dalam melakukan aktivitas *software development* adalah 77 hari.

#### 6.3.7.1 Alternatif 1. Memanfaatkan Waktu Lembur

Dalam menghitung waktu lembur, sama seperti sebelumnya perlu dicari perubahan hari kerja biasa dari 80 hari (640 jam) menjadi 77 hari, sehingga memerlukan 3 hari (24 jam) waktu lembur. Perubahan tersebut seperti pada **Tabel 6.73**.

Tabel 6.73 Perhitungan Hari Kerja Biasa Pembangunan dalam 77 hari (Informasi Lokasi)

<b>PERUBAHAN HARI KERJA DALAM 5 BULAN</b>										
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jam Kerja / hari</b>	<b>Hari kerja 80 hari</b>	<b>Total jam kerja 80 hari</b>	<b>Total jam kerja 77 hari</b>	<b>Bobot</b>	<b>Pembulatan (jam)</b>	<b>Hari kerja (hari)</b>	<b>Pembulatan (hari)</b>	<b>Pwrubah andalam Jam</b>
<b>Software Development</b>										
<i>Requirement</i>	<i>Business analyst</i>	8	160	640	616	154.00	154	19.25	15	120
<i>Spesification &amp; Design</i>	<i>System analyst</i>	8	0	0	616	0	0	0	0	0
<i>Coding</i>	<i>Programmer</i>	8	480	640	616	462.00	462	57.75	58	464
<i>Testing</i>	<i>Technical tester</i>	8	0	0	616	0	0	0	0	0
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	8	0	0	616	0	0	0	0	0
<b>TOTAL</b>						<b>616,00</b>	<b>616</b>	<b>77</b>	<b>77</b>	<b>616</b>

Selanjutnya dilakukan perhitungan biaya pemanfaatan waktu lembur dengan hasil seperti pada **Tabel 6.74**.

Tabel 6.74 Perhitungan Estimasi Biaya Kerja Biasa setelah Perubahan selama 77 hari (Informasi Lokasi)

Kelompok aktivitas	Jabatan	Person	Hours	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	120	Rp 43.750	Rp 6.650.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	0	0	Rp 43.750	Rp 0
<i>Coding</i>	<i>Programmer</i>	1	464	Rp 50.000	Rp 23.200.000
<i>Testing</i>	<i>Technical tester</i>	0	0	Rp 50.000	Rp 0
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	0	Rp 93.750	Rp 0
<b>TOTAL</b>					<b>Rp 29.850.000</b>

Selanjutnya dilakukan perhitungan pemanfaatan waktu lembur. Disebabkan bobot pekerjaan antar jabatan berbeda, sehingga waktu lembur yang mereka miliki juga berbeda. Pembagiannya seperti pada **Tabel 6.75**.

Tabel 6.75 Perhitungan Hari Lembur (Informasi Lokasi)

<b>PERHITUNGAN HARI LEMBUR</b>		
Jam kerja di 80 hari	Perubahan jam kerja ke 77 hari	Jam lembur 77 hari
<b>Software Development</b>		
160	120	8
0	0	0
480	464	16

<b>PERHITUNGAN HARI LEMBUR</b>		
<b>Jam kerja di 80 hari</b>	<b>Perubahan jam kerja ke 77 hari</b>	<b>Jam lembur 77 hari</b>
<b><i>Software Development</i></b>		
0	0	0
0	0	0

Berdasarkan **Tabel 6.75.** maka perhitungan estimasi biaya kerja lembur selama 77 hari seperti pada **Tabel 6.76.**

Tabel 6.76 Perhitunga Estimasi Biaya Lembur 77 hari (Informasi Lokasi)

Kelompok aktivitas	Jabatan	Person (Orang)	Hours (Jam)	Standar Gaji (per-jam)	Upah/bulan (Rp)	Gaji lembur hari pertama (Rp)	Gaji lembur hari selanjutnya (Rp)
<b>Software Development</b>							
<i>Requirement</i>	<i>Business analyst</i>	1	24	Rp 43750	7.000.000	Rp 60.694	Rp 566.474
<i>Spesification &amp; Design</i>	<i>System analyst</i>	0	24	Rp 43750	7.000.000	Rp 0	Rp 0
<i>Coding</i>	<i>Programmer</i>	1	64	Rp 50000	8.000.000	Rp 69.364	Rp 1.387.283
<i>Testing</i>	<i>Technical tester</i>	0	40	Rp 50000	8.000.000	Rp 0	Rp 0
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	32	Rp 93750	15.000.000	Rp 0	Rp 0
<b>TOTAL</b>						<b>Rp 130.078</b>	<b>Rp 1.953.757</b>

Berdasarkan hasil perhitungan konversi hari kerja, biaya pengerjaan fungsi ini adalah dengan menjumlahkan nilai estimasi biaya + biaya lembur hari kerja. Secara numerik sebagai berikut.

Biaya pembangunan aplikasi :

$$= \text{Rp } 29.850.000 + \text{Rp } 130.058 + \text{Rp } 1.953.757$$

$$= \text{Rp } 31.933.815^*$$

**\*Biaya lebih mahal dari hasil awal.**

### 6.3.7.2 Alternatif 2. Penambahan SDM versi 1

Berdasarkan perhitungan skenario sebelumnya, nilai dengan memanfaatkan waktu lembur akan lebih tinggi dari nilai awal, sehingga skenario ini bersifat memanfaatkan penambahan SDM. Hasil penambahan jumlah SDM seperti **Tabel 6.77**.

Tabel 6.77 Penambahan Jumlah SDM (Informasi Lokasi)

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
Aktivitas	Jabatan	BEFORE		AFTER	
		Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	20	1	17
<i>Spesification &amp; Design</i>	<i>System analyst</i>	0	0	0	0
<i>Coding</i>	<i>Programmer</i>	1	60	1	60
<i>Testing</i>	<i>Technical tester</i>	0	0	0	0
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	0	0	0
<b>TOTAL</b>		<b>80</b>		<b>77</b>	

Berdasarkan perubahan jumlah SDM tersebut, maka estimasi biaya pembangunan aplikasi menjadi seperti pada **Tabel 6.78**.

Tabel 6.78 Perhitungan Estimasi Biaya dengan Penambahan SDM

<b>PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM</b>					
<b>Aktivitas</b>	<b>Jabatan</b>	<b>Jumlah pekerja (Orang)</b>	<b>Hari Kerja (Hari)</b>	<b>Standar gaji/hari (Rp)</b>	<b>Hasil (Rp)</b>
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	17	350.000	5.950.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	0	0	350.000	0
<i>Coding</i>	<i>Programmer</i>	1	60	400.000	24.000.000
<i>Testing</i>	<i>Technical tester</i>	0	0	400.000	0
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	0	750.000	0
<b>TOTAL</b>					<b>29.950.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 29.950.000, lebih murah dari harga awal.

### 6.3.7.3 Alternatif 3. Penambahan SDM versi 2

Berdasarkan perhitungan skenario sebelumnya, nilai dengan memanfaatkan waktu lembur akan lebih tinggi dari nilai awal, sehingga skenario ini bersifat memanfaatkan penambahan SDM. Hasil penambahan jumlah SDM seperti **Tabel 6.79**.

Tabel 6.79 Penambahan Jumlah SDM (Informasi Lokasi)

PERHITUNGAN PENAMBAHAN JUMLAH SDM					
		BEFORE		AFTER	
Aktivitas	Jabatan	Jumlah pekerja	Hari Kerja	Jumlah pekerja	Hari Kerja
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	20	1	20
<i>Spesification &amp; Design</i>	<i>System analyst</i>	0	0	0	0
<i>Coding</i>	<i>Programmer</i>	1	60	1	57
<i>Testing</i>	<i>Technical tester</i>	0	0	0	0
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	0	0	0
<b>TOTAL</b>		<b>80</b>		<b>77</b>	

Berdasarkan perubahan jumlah SDM tersebut, maka estimasi biaya pembangunan aplikasi menjadi seperti pada **Tabel 6.80**.

Tabel 6.80 Perhitungan Estimasi Biaya dengan Penambahan SDM

PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM					
Aktivitas	Jabatan	Jumlah pekerja (Orang)	Hari Kerja (Hari)	Standar gaji/hari (Rp)	Hasil (Rp)
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	20	350.000	7.000.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	0	0	350.000	0
<i>Coding</i>	<i>Programmer</i>	1	57	400.000	22.800.000
<i>Testing</i>	<i>Technical tester</i>	0	0	400.000	0

PERHITUNGAN BIAYA DENGAN PENAMBAHAN SDM					
Aktivitas	Jabatan	Jumlah pekerja (Orang)	Hari Kerja (Hari)	Standar gaji/hari (Rp)	Hasil (Rp)
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	0	750.000	0
<b>TOTAL</b>					<b>29.800.000</b>

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 29.950.000, lebih murah dari harga awal.

#### 6.3.7.4 Alternatif 4. Memanfaatkan Visual Programming

Skenario ketiga adalah pemanfaatan visual programming. Dengan adanya visual programming, waktu pengerjaan *programmer* dapat dikurangi. Hasil yang diperoleh seperti pada **Tabel 6.81**.

Tabel 6.81 Penggunaan Visual Programming (Informasi Lokasi)

Kelompok aktivitas	Jabatan	Person	Days	Standar Gaji (per-hari)	Total
<b>Software Development</b>					
<i>Requirement</i>	<i>Business analyst</i>	1	20	Rp 350.000	Rp 7.000.000
<i>Spesification &amp; Design</i>	<i>System analyst</i>	0	0	Rp 350.000	Rp 0
<i>Coding</i>	<i>Programmer</i>	1	11	Rp 400.000	Rp 4.400.000
<i>Testing</i>	<i>Technical tester</i>	0	0	Rp 400.000	Rp 0
<i>Acceptance and Deployment</i>	<i>Technical Team leader</i>	0	0	Rp 750.000	Rp 0

<b>TOTAL</b>	<b>Rp 11.400.000</b>
--------------	--------------------------

Berdasarkan hasil perhitungan penambahan SDM, biaya pengerjaan fungsi ini adalah Rp 11.400.000, lebih murah dari harga awal.

Untuk mempermudah mencari tahu alternatif apa saja yang penulis berhasil temukan, maka seluruh alternative akan dirangkum kedalam **Tabel 6.82**.

Tabel 6.82 Hasil Tahapan Creative

No.	Kebutuhan Fungsional	Alternatif Kebutuhan Fungsional	Kode Skenario
1.	Melihat Informasi Saldo	Pemanfaatan waktu lembur	SK01
		Penambahan SDM	SK02
		Penggunaan Visual Programming	SK03
2.	Melihat Aktivitas Transaksi	Pemanfaatan waktu lembur	SK04
		Penambahan SDM	SK05
		Penggunaan Visual Programming	SK06
3.	Pemesanan Layanan dan Jasa	Pemanfaatan waktu lembur	SK07
		Penambahan SDM versi 1	SK08
		Penambahan SDM versi 2	SK09
		Penggunaan Visual Programming	SK10
4.	Menampilkan informasi penjemputan	Pemanfaatan waktu lembur	SK11
		Penambahan SDM	SK12
		Penggunaan Visual Programming	SK13
5.	Melihat Informasi Hotel di Sekitar	Pemanfaatan waktu lembur	SK14
		Penambahan SDM	SK15
		Penggunaan Visual Programming	SK16
6.		Pemanfaatan waktu lembur	SK17
		Penambahan SDM	SK18

No.	Kebutuhan Fungsional	Alternatif Kebutuhan Fungsional	Kode Skenario
	Melihat Informasi <i>Mall Official Partner</i>	Penggunaan Visual Programming	SK19
7.	Menampilkan informasi lokasi	Pemanfaatan waktu lembur	SK20
		Penambahan SDM versi 1	SK21
		Penambahan SDM versi 2	SK22
		Penggunaan Visual Programming	SK23

#### 6.4 Evaluation Phase

Tahapan ini adalah tahapan terakhir dalam pengerjaan Tugas Akhir ini. Berupa tahapan evaluasi yang akan melibatkan 2 teknik analisis. Hal pertama yang penulis lakukan adalah merangkum semua alternatif yang telah penulis peroleh untuk melihat perubahan biaya yang didapatkan.

Tabel 6.83 Hasil Perubahan Biaya berdasarkan Skenario Pemanfaatan Waktu Lembur

No.	Kebutuhan Fungsional	Biaya Awal	Biaya Akhir
1.	Login	Rp 25.800.000	Rp 25.800.000
2.	Melihat Informasi Saldo	Rp 61.750.000	Rp 77.316.474
3.	Melihat Aktivitas Transaksi	Rp 60.350.000	Rp 74.344.509
4.	Pemesanan Layanan dan Jasa	Rp 136.900.000	RP 185.325.433
5.	Menampilkan informasi penjemputan	Rp 34.500.000	Rp 38.661.850
6.	Melihat Informasi	Rp 43.750.000	Rp 52.371.964

No.	Kebutuhan Fungsional	Biaya Awal	Biaya Akhir
	Hotel di Sekitar		
7.	Melihat Informasi <i>Mall Official Partner</i>	Rp 43.750.000	Rp 52.371.964
8.	Menampilkan informasi jadwal dan lokasi pertandingan	Rp 24.000.000	Rp 24.000.000
9.	Menyampaikan keluhan	Rp 27.700.000	Rp 27.700.000
10.	Menampilkan informasi lokasi	Rp 31.000.000	Rp 31.933.815
11.	Menampilkan app dalam berbagai bahasa	Rp 21.600.000	Rp 21.600.000
<b>TOTAL</b>		Rp 511.100.000	Rp 611.426.009

Tabel 6.84 Hasil Perubahan Biaya berdasarkan Skenario Penambahan SDM versi 1

No.	Kebutuhan Fungsional	Biaya Awal	Biaya Akhir
1.	Login	Rp 25.800.000	Rp 25.800.000
2.	Melihat Informasi Saldo	Rp 61.750.000	Rp 61.350.000
3.	Melihat Aktivitas Transaksi	Rp 60.350.000	Rp 59.600.000

No.	Kebutuhan Fungsional	Biaya Awal	Biaya Akhir
4.	Pemesanan Layanan dan Jasa	Rp 136.900.000	Rp 136.200.000
5.	Menampilkan informasi penjemputan	Rp 34.500.000	Rp 33.450.000
6.	Melihat Informasi Hotel di Sekitar	Rp 43.750.000	Rp 40.250.000
7.	Melihat Informasi <i>Mall Official Partner</i>	Rp 43.750.000	Rp 40.250.000
8.	Menampilkan informasi jadwal dan lokasi pertandingan	Rp 24.000.000	Rp 24.000.000
9.	Menyampaikan keluhan	Rp 27.700.000	Rp 27.700.000
10.	Menampilkan informasi lokasi	Rp 31.000.000	Rp 29.950.000
11.	Menampilkan app dalam berbagai bahasa	Rp 21.600.000	Rp 21.600.000
<b>TOTAL</b>		Rp 511.100.000	Rp 500.150.000

Tabel 6.85 Hasil Perubahan Biaya berdasarkan Skenario Penambahan SDM versi 2

No.	Kebutuhan Fungsional	Biaya Awal	Biaya Akhir
1.	Login	Rp 25.800.000	Rp 25.800.000
2.	Melihat Informasi Saldo	Rp 61.750.000	Rp 61.350.000
3.	Melihat Aktivitas Transaksi	Rp 60.350.000	Rp 59.600.000
4.	Pemesanan Layanan dan Jasa	Rp 136.900.000	Rp 134.200.000
5.	Menampilkan informasi penjemputan	Rp 34.500.000	Rp 33.450.000
6.	Melihat Informasi Hotel di Sekitar	Rp 43.750.000	Rp 40.250.000
7.	Melihat Informasi <i>Mall Official Partner</i>	Rp 43.750.000	Rp 40.250.000
8.	Menampilkan informasi jadwal dan lokasi pertandingan	Rp 24.000.000	Rp 24.000.000
9.	Menyampaikan keluhan	Rp 27.700.000	Rp 27.700.000
10.	Menampilkan informasi lokasi	Rp 31.000.000	Rp 29.800.000
11.	Menampilkan app dalam	Rp 21.600.000	Rp 21.600.000

No.	Kebutuhan Fungsional	Biaya Awal	Biaya Akhir
	berbagai bahasa		
<b>TOTAL</b>		Rp 511.100.000	Rp 498.000.000

Tabel 6.86 Hasil Perubahan Biaya berdasarkan Skenario Penggunaan Visual Programming

No.	Kebutuhan Fungsional	Biaya Awal	Biaya Akhir
1.	Login	Rp 25.800.000	Rp 25.800.000
2.	Melihat Informasi Saldo	Rp 61.750.000	Rp 34.150.000
3.	Melihat Aktivitas Transaksi	Rp 60.350.000	Rp 32.750.000
4.	Pemesanan Layanan dan Jasa	Rp 136.900.000	Rp 69.300.000
5.	Menampilkan informasi penjemputan	Rp 34.500.000	Rp 20.900.000
6.	Melihat Informasi Hotel di Sekitar	Rp 43.750.000	Rp 34.150.000
7.	Melihat Informasi <i>Mall Official Partner</i>	Rp 43.750.000	Rp 34.150.000
8.	Menampilkan informasi jadwal dan lokasi pertandingan	Rp 24.000.000	Rp 24.000.000

No.	Kebutuhan Fungsional	Biaya Awal	Biaya Akhir
9.	Menyampaikan keluhan	Rp 27.700.000	Rp 27.700.000
10.	Menampilkan informasi lokasi	Rp 31.000.000	Rp 11.400.000
11.	Menampilkan app dalam berbagai bahasa	Rp 21.600.000	Rp 21.600.000
<b>TOTAL</b>		Rp 511.100.000	Rp 335.900.000

Berdasarkan alternatif yang penulis peroleh, alternatif dengan memanfaatkan waktu lembur justru menambah biaya pembangunan aplikasi, sedangkan biaya penambahan SDM dan Pemanfaatan Visual Programming, menurunkan biaya pembangunan aplikasi. Alternatif dengan biaya penghematan terbesar adalah alternatif Pemanfaatan Visual Programming yakni 34 % atau sebesar Rp 175.200.000.

#### 6.4.1 Pencarian Indeks Nilai

Untuk menganalisis apakah alternatif tersebut layak dipertimbangkan atau tidak, penulis melakukan pencarian indeks nilai dengan menggunakan persamaan yang sudah dijelaskan pada bab sebelumnya. Adapun hasil yang diperoleh akan dimasukkan kedalam **Tabel 6.87**.

Tabel 6.87 Hasil Pencarian Indeks Nilai

No	Alternatif Kebutuhan Fungsional	Hasil Pencarian Indeks	Keterangan
1.	Pemanfaatan waktu lembur	-19.63%	Tidak Layak Dipertimbangkan
2.	Penambahan SDM versi 1	2.14%	Layak Dipertimbangkan

No	Alternatif Kebutuhan Fungsional	Hasil Pencarian Indeks	Keterangan
3.	Penambahan SDM versi 2	2.56%	Layak Dipertimbangkan
4.	Pemanfaatan Visual Programming	34.27%	Layak Dipertimbangkan

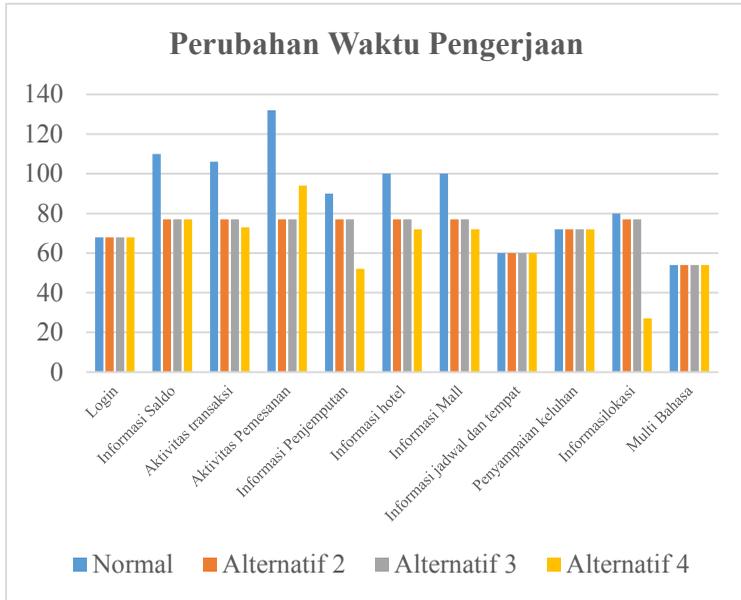
Berdasarkan hasil perhitungan indeks nilai, terdapat 3 alternatif yang layak dipertimbangkan, yakni penambahan SDM versi 1 dan 2 serta Pemanfaatan Visual Programming.

#### 6.4.2 Pemilihan Rekomendasi

Dengan menggunakan prinsip pengerjaan proyek yang optimal, penulis memilih rekomendasi dengan waktu dan biaya pengerjaan paling murah dengan memanfaatkan grafik hubungan waktu dan biaya. Alternatif yang diperhitungkan adalah alternatif yang layak dipertimbangkan berdasarkan hasil pencarian indeks.

##### ➤ Perubahan Waktu Pengerjaan

Untuk mengetahui waktu pengerjaan proyek paling cepat, maka perlu dicarikan perubahan waktu pengerjaan proyek, yang pada kasus ini perubahan pada masing-masing pengerjaan kebutuhan fungsional aplikasi. Perubahan waktu pengerjaan yang dicarikan adalah waktu normal, alternatif 2, 3 dan 4.



**Grafik 6.1 Perubahan Waktu Pengerjaan Aplikasi**

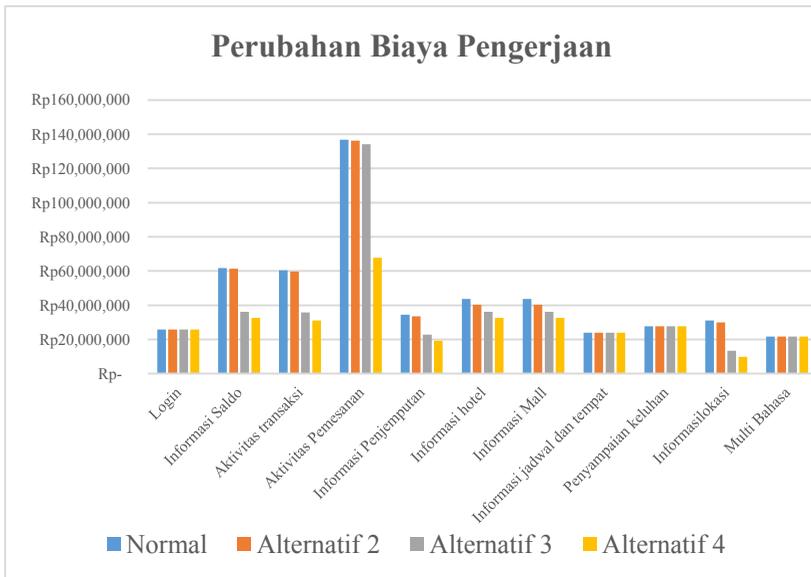
Berdasarkan grafik tersebut, dapat diketahui :

- Terdapat perubahan waktu pengerjaan yang sangat tinggi pada beberapa fungsional seperti informasi saldo, aktivitas transaksi, dan aktivitas pemesanan.
- Fungsi yang tidak dijadikan sebagai objek penelitian (diskenariokan) tidak memiliki perubahan waktu pengerjaan
- Waktu pengerjaan normal adalah 972 hari.
- Waktu pengerjaan alternative 2 dan 3 adalah sama yakni 793 atau berkurang 18% dari waktu normal.
- Waktu pengerjaan alternatif 4 adalah 721 atau berkurang 25,8% dari waktu normal.

Sehingga dapat disimpulkan bahwa alternatif dengan waktu paling cepat adalah alternatif 4.

### ➤ Perubahan Biaya Pengerjaan

Untuk mengetahui biaya pengerjaan proyek paling cepat, maka perlu dicarikan perubahan biaya pengerjaan proyek, yang pada kasus ini perubahan pada masing-masing pengerjaan kebutuhan fungsional aplikasi. Perubahan biaya pengerjaan yang dicarikan adalah biaya normal, alternatif 2, 3 dan 4.



Grafik 6.2 Perubahan Biaya Pembangunan Aplikasi

Berdasarkan grafik tersebut, dapat diketahui :

- Terdapat perubahan biaya pengerjaan aplikasi yang sangat tinggi dari harga normal ke alternatif 4 pada beberapa fungsi seperti saldo, aktivitas transaksi, dan aktivitas pemesanan.
- Fungsi yang tidak dijadikan sebagai objek penelitian (diskernariokan) tidak memiliki perubahan biayapengerjaan

- Biaya pengerjaan normal adalah Rp 511.100.000
- Biaya pengerjaan alternatif 2 adalah Rp 500.150.000 atau berkurang 2,14% dari harga normal.
- Biaya pengerjaan alternatif 3 adalah Rp 498.000.000 atau berkurang 2,56% dari harga normal.
- Biaya pengerjaan alternatif 4 adalah Rp 335.9000.000 atau berkurang 34,27% dari harga normal.

Sehingga dapat disimpulkan bahwa skenario dengan biaya paling cepat adalah alternatif 4.

Berdasarkan hasil evaluasi tersebut, maka dapat disimpulkan bahwa alternatif 4 yakni pemanfaatan visual programming menjadi alternatif yang memiliki waktu paling cepat dan biaya paling murah, sehingga menjadi rekomendasi penulis bagi implementasi proyek pembangunan aplikasi ini. Secara grafik hubungan biaya dan waktu pembangunan aplikasi adalah sebagai berikut.



Grafik 6.3 Hubungan Waktu dan Biaya

Grafik dengan pengerjaan paling optimum yakni grafik yang memiliki hubungan antara waktu dan biaya paling mendekati sumbu 0, yakni alternatif 4.

#### **6.4.3 Perbandingan Estimasi Biaya**

Berdasarkan hasil akhir yang penulis dapatkan, berikut adalah perbandingan hasil beserta justifikasinya.

Tabel 6.88 Perbandingan Harga Aplikasi

<b>Metode</b>	Pengalaman Pihak Ketiga		<i>Function point</i>		<i>Value engineering</i>	
<b>Biaya</b>	Rp 439.300.000		Rp 201.900.000		Rp 335.900.000	
<b>Keterangan</b>	Harga berdasarkan Kerangka Acuan Kerja (KAK)		Menghitung harga pengalaman pihak ketiga dengan <i>Function point</i>		Menerapkan skenario memanfaatkan visual programming	
<b>Justifikasi</b>	<b>Kelebihan</b>	<b>Kelemahan</b>	<b>Kelebihan</b>	<b>Kelemahan</b>	<b>Kelebihan</b>	<b>Kelemahan</b>
	<ul style="list-style-type: none"> <li>Waktu kerja masing-masing jabatan sudah ditentukan.</li> </ul>	<ul style="list-style-type: none"> <li>Harga didasarkan pada <i>manhours</i> bukan pada aktivitas dan kebutuhan fungsional yang akan dilakukan saat pembangunana aplikasi.</li> <li>Standar gaji yang digunakan tidak diketahui.</li> </ul>	<ul style="list-style-type: none"> <li>Menggunakan standart gaji dari <i>Sallary Guide 2018</i> oleh Kelly.</li> <li>Pembagian aktivitas masing-masing jabatan sudah jelas.</li> <li>Pembagian pembangunan aplikasi</li> </ul>	<ul style="list-style-type: none"> <li>Estimasi harga masih didasarkan pada <i>manhours</i>, bukan pada kebutuhan fungsional meskipun KAK sudah memberikan kebutuhan fungsinya.</li> </ul>	<ul style="list-style-type: none"> <li>Menggunakan standart gaji dari <i>Sallary Guide 2018</i> oleh Kelly.</li> <li>Pembagian aktivitas masing-masing jabatan sudah jelas.</li> <li>Pembagian pembangunan aplikasi</li> </ul>	<ul style="list-style-type: none"> <li>Perlu membeli lisensi visual programming sebesar \$ 2,940.00 atau senilai Rp 41.421.807</li> <li>Harga ini dihitung dengan mengibaratkan pembangunan dilakukan per-fungsi,</li> </ul>

		<ul style="list-style-type: none"> <li>• Pengerjaan aplikasi terbagi atas <i>product</i> dan <i>deployment</i>, namun tidak dijelaskan aktivitas apa yang dilaksanakan pada masing-masing tahap. Pembagian aktivitas masing-masing jabatan tidak dijelaskan.</li> </ul>	terbagi 3 yakni yakni <i>software development</i> , <i>ongoing</i> , dan <i>testing</i> , yang seluruhnya dilengkapi aktivitas masing-masing.		terbagi 3 yakni yakni <i>software development</i> , <i>ongoing</i> , dan <i>testing</i> , yang seluruhnya dilengkapi aktivitas masing-masing <ul style="list-style-type: none"> <li>• Estimasi harga berdasarkan kebutuhan fungsional aplikasi</li> </ul>	sehingga harga memiliki perhitungan yang berulang (misal : harga pembangunan database transaksi, diperlukan pada informasi transaksi dan aktivitas transaksi, dalam perhitungan ini, harga database transaksi dihitung dua kali).
--	--	---	---	--	---	---

#### **6.4.4 *Value Engineering Change Proposal***

Berdasarkan hasil akhir yang penulis dapatkan, maka penulis membuat *Value Engineering Change Proposal* (VECP) (Lampiran B) sebagai bentuk rangkuman hasil penelitian penulis untuk diserahkan kepada pihak ketiga, sehingga pihak ketiga dapat mengetahui dan menyetujui rekomendasi tersebut untuk kemudian diimplementasikan dalam pengajuan biaya baru pembangunan aplikasi XYZ.

*Halaman ini sengaja dikosongkan*

## **BAB VII**

### **KESIMPULAN DAN SARAN**

Bab ini akan menjelaskan hasil analisis dan proses yang telah dilakukan selama penelitian Tugas Akhir ini dilakukan.

#### **7.1 Kesimpulan**

Kesimpulan yang didapatkan dari penelitian ini adalah :

1. Hasil dari perhitungan harga pembangunan masing-masing fungsi pada aplikasi ini, disesuaikan dengan syarat pemilihan objek penelitian yakni fungsi yang memiliki waktu pengerjaan lebih dari 600 jam dan dikerjakan oleh minimal 2 orang, maka terdapat 7 fungsi yang dipilih menjadi objek penelitian sekaligus objek yang akan dibuat skenario untuk dilakukan penghematan. Objek tersebut adalah :
  - Menampilkan informasi saldo
  - Menampilkan informasi transaksi
  - Menampilkan aktivitas pemesanan layanan dan jasa
  - Menampilkan informasi hotel di sekitar
  - Menampilkan informasi *mall official partner* di sekitar
  - Menampilkan informasi pendamping
  - Menampilkan informasi lokasi

Temuan penulis terhadap fungsi adalah sebagai berikut :

- Berdasarkan 7 fungsi yang terpilih, terdapat 6 fungsi untuk menampilkan informasi dan hanya 1 fungsi menampilkan aktivitas.
- Fungsi menampilkan aktivitas adalah fungsi yang membutuhkan biaya paling besar yakni Rp 136.900.000 atau memakan biaya sebesar 26,78%, sedangkan fungsi paling murah adalah menampilkan informasi lokasi yakni Rp 21.000.000 atau 4,1%.

2. Hasil pemilihan objek penelitian tersebut, diolah dengan memberikan beberapa skenario untuk melakukan penghematan biaya. Terdapat 4 jenis alternatif yang penulis dapatkan. Temuan penulis terhadap alternatif adalah sebagai berikut :
  - Terdapat total 23 skenario dari 4 alternatif yang ada.
  - Alternatif pemanfaatan waktu lembur tidak layak dipertimbangkan karena membuat biaya pembangunan menjadi lebih mahal
  - Fungsi menampilkan informasi hotel dan *mall official partner* di sekitar memiliki kesamaan, baik dalam jumlah SDM maupun waktu pengerjaan, sehingga hasil skenario cenderung sama.

Berdasarkan hasil yang diperoleh, alternatif yang memberikan penghematan biaya yang paling besar adalah alternative jenis 4 yakni pemanfaatan *visual programming*.

3. Hasil perhitungan alternatif 4, menunjukkan perubahan yang besar yakni dari Rp 511.100.000 menjadi Rp 335.900.000 atau 63,73%, artinya alternatif ini mampu menghemat biaya pengerjaan aplikasi sebesar 34.27% atau Rp 175.200.000. Penghematan ini diperoleh karena masing-masing fungsi membutuhkan *programmer*, yang waktu pengerjaannya berkurang dengan adanya *visual programming*.

## 7.2 Saran

Saran yang dapat diambil dari hasil pengerjaan Tugas Akhir dengan studi kasus Proyek Pembangunan Aplikasi XYZ adalah :

1. Pada KAK (Kerangka Acuan Kerja) kebutuhan fungsional aplikasi telah diberikan sehingga penulis dapat langsung mengolah informasi tersebut. Jika penelitian berikutnya tidak terdapat KAK yang menyertakan kebutuhan

fungsional aplikasi, maka kebutuhan fungsional dapat diketahui dengan melakukan *Benchmarking* dengan penyedia jasa yang membuat aplikasi sejenis.

2. Tujuan penelitian ini adalah penghematan biaya. Hal ini disebabkan oleh cakupan penelitian yang terbatas pada tahap perencanaan. Sehingga tidak banyak nilai yang bisa dibandingkan. Jika penelitian berikutnya sampai kepada tahap implementasi, maka untuk memperoleh nilai-nilai lain, penulis bisa menggunakan metode *David Silk*, untuk mengetahui keuntungan-keuntungan lain yang bisa diperoleh dengan adanya aplikasi tersebut. Adapun implementasi metode David Silk [33].
3. Penelitian ini sangat bergantung pada standar gaji karyawan dalam perhitungan *Function point*. Hal ini karena penulis mengacu pada KAK (Kerangka Acuan Kerja) dari pihak ketiga yang telah menyertakan gaji (*manhours*) karyawan yang terlibat. Sehingga setelah nilai *Function point* (FP) diperoleh, penulis perlu melakukan *breakdown* lagi SDM dan waktu pengerjaan pada masing-masing aktivitas. Namun, jika pada penelitian berikutnya, tidak terdapat acuan gaji masing-masing karyawan, maka estimasi biaya dapat dihitung dengan menggunakan konsep pengali. Konsep ini tetap menggunakan *Function point* namun tidak perlu melakukan *breakdown* setelah FP diperoleh, hanya mengalikannya dengan standart pengali sesuai negara masing-masing [34].

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] Tassinari, R. (1985). *Le rapport qualité/ prix. Paris: Les éditions d'organisation.*
- [2] Standish. 2014. *Chaos Report – Project Smart, pp. 2-4, (Online),* Available :  
([www.projectsmart.co.uk/docs/chaos-report.pdf](http://www.projectsmart.co.uk/docs/chaos-report.pdf))  
[Diakses : 17 Februari 2018].
- [3] Indriani, Farida. 2006. *Studi Mengenai Orientasi Inovasi, Pengembangan Produk dan Efektifitas Promosi Sebagai Sebuah Strategi untuk Meningkatkan Kinerja Produk. Jurnal Studi Manajemen & Organisasi.* Universitas Diponegoro. Semarang. Available:  
<https://ejournal.undip.ac.id/index.php/smo/article/download/4191/3812>. [Diakses: 21 Februari 2018].
- [4] Publications paper. UIN. Riau. Available :  
<http://repository.uin-suska.ac.id/4380/3/10.%20BAB%20II%281%29.pdf>.  
[Diakses : 7 Juli 2018].
- [5] Sievert, Richard W. 1991. *A Review of Value engineering as an Effective System for Planning Building Project.* Project Management Journal, p.22,31-38.
- [6] Thorson, K.R. and R. Snidar. 2011. *Improving Profits Through Prime/Subcontractor Value engineering.* USA.
- [7] Unpublished Document. *Kerangka Acuan Kerja Asian Games, 2018.* Indonesia.
- [8] Project Management Institute, Inc., 2013. *A guide to project management body of knowledge,* Fifth Edition, Newton Square, p.3-4.
- [9] Pinheiro, Jilvan. 2011. *Software Development Life Cycle (SDLC) Phases.* Available :  
<https://medium.com/@jilvanpinheiro/software-development-life-cycle-sdlc-Phases-40d46afbe384>.  
[Diakses: 9 Juli 2018].

- [10] Snodgrass, T., & M. Kasi. (1986). *Function Analysis—The Stepping Stones to Good Value*. Madison, WI: Board of Regents, University of Wisconsin System.
- [11] Wikimedia. 2018. *Value engineering*. Available: [https://caradaftar-terbaru.blogspot.co.id/2016/12/panduan-cara-menulis-daftar-pustaka\\_4.html](https://caradaftar-terbaru.blogspot.co.id/2016/12/panduan-cara-menulis-daftar-pustaka_4.html). [Diakses 14 Maret 2018].
- [12] Savaliya, Yagnik. 2014. *Work Design and Measurement FAST Diagram*. Slideshare. Available : <https://www.slideshare.net/yyfece/fast-diagram> [Diakses 08 Juni 2018]
- [13] SAVE International. (2007). *Value Methodology Standard and Body of Knowledge*. SAVE International, Dayton, OH. P.8. Gummesson, E. (1988). *Qualitative methods in management research*. Lund: Norway: Studentlitteratur, Chartwell-Bratt.
- [14] Thiry, Michel. 2013. *A Framework for Value Management Practice*. Project Management Institute: ISBN 978-1-62825-018-3.
- [15] Sabrang, H. (1998), *Value engineering*, Universitas Atma Jaya Yogyakarta.
- [16] Merla, E. 2010. *Engineering Success Project by Project: Value engineering for Project managers*. Paper Presented at PMI Global Congress 2010 - EMEA, Milan, Italy. Newtown Square, PA: Project Management Institute.
- [17] Publications Catalog. GAO. 2017. “*Information Technology Investment Management. A Framework for Assessing and Improving Process Maturity*”. Available : <http://www.gao.gov/new.items/d04394g.pdf> [Diakses: 19 Maret 2018].
- [18] Thompson, H. 2016. *Medical Design & Outsourcing. A Mass Device Resource*. WTWH Media, LLC. Available : <https://www.medicalpubdesignandoutsourcing.com/harnessing-power-Value-engineering/> [Diakses: 14 Maret 2018]

- [19] Ahuja, H.N. 1994. *Project Management, Techniques in Planning and Controlling Construction Project*. USA: John Wiley & Sons Inc.
- [20] Publications Paper. 2016. *New York State Department of Transportation Contract Administration Manual*. Available: [https://www.dot.ny.gov/main/business-center/contractors/construction-division/construction-repository/CAM\\_Sect104-10.pdf](https://www.dot.ny.gov/main/business-center/contractors/construction-division/construction-repository/CAM_Sect104-10.pdf) [Diakses: 6 Maret 2018].
- [21] Publications Catalog. E.W. Arthur. 1995. *Value engineering, Departement of the Army*. US Army Corps of Engineers. USA. Available: [http://www.publications.usace.army.mil/Portals/76/Publications/EngineerPamphlets/EP\\_11-1-4.pdf?ver=2013-08-22-090137-173](http://www.publications.usace.army.mil/Portals/76/Publications/EngineerPamphlets/EP_11-1-4.pdf?ver=2013-08-22-090137-173) [Diakses: 6 Maret 2018].
- [22] Publication Journal. Fauzannim, M. *Value Engineering Change Proposal*. Available: <http://www.digilib.itb.ac.id/files/disk1/608/jbptitbpp-gdl-mfauzannim-30371-3-2008ts-2.pdf> [Diakses : 21 Maret 2018]
- [23] Hian, S. C. 2012. “*Value engineering and Value Management – Is There Any Difference?*”. Publication of KPK Research.
- [24] Saputra, S.N. (2012), *Arti Benchmarking*. Available: <https://www.slideshare.net/sofyanardisaputra/arti-Benchmarking> [Diakses: 30 Maret 2018]
- [25] Unpublished Document. A.Fandhi. 2018. *Rancang Bangun Aplikasi Function point untk Estimasi Harga Perangkat Lunak*. Institut Teknologi Sepuluh Nopember. Surabaya.
- [26] *Value engineering Canada, “Function Analysis System Technique,”* 2015, Available: <http://www.ValueAnalysis.ca/fast.php> [Diakses: 02 Februari 2018].

- [27] Publication Module. Makarim, C. A. 2007. “*Value engineering E-Learning 2007 Module*”. GLAD Batch 3. Jakarta.
- [28] Priyo, M dan Hermawan, D.H. 2010. *Aplikasi Value engineering pada Proyek Konstruksi*. Jurnal Ilmiah Semesta Tekika, thirteenth edition, 2, 116-119.
- [29] Osborn, A.F. (1963). *Applied imagination: Principles and procedures of creative problem solving* (Third Revised Edition). New York, NY: Charles Scribner’s Sons
- [30] Lifehacker. 2015. “*This Graphic Explains How Much Time and Money It Takes to Develop a Mobile App*”. Available: <https://lifehacker.com/this-graphic-explains-how-much-time-and-money-it-takes-1735164869> [Diakses: 06 Juni 2018]
- [31] ESDS. 2013. *Introduction to Software Development Life Cycle (SDLC) Phases & Models*. Available: <https://www.esds.co.in/blog/introduction-to-software-development-life-cycle-sdlc-Phases-models/#sthash.vKB49Oq7.dpbs>. [Diakses: 08 Juni 2018].
- [32] Wiki. 2013. Rational Rose. Available : <http://wiki.c2.com/?RationalRose> [Diakses 06 Juni 2018]
- [33] Unpublished Document. Ranishia, Nina. 2018. Analisis Manfaat Intangible Persepsi Penyedia Layanan Teknologi Informasi dengan Pendekatan Silk (Studi Kasus : Service Desk).
- [34] Publications Paper. 2016. Garg. Nikita, dkk. *Cost Estimation of Functional Requirement of Insitute Examination System*. Available : <https://research.ijcaonline.org/rtfem2016/number1/rtfem45115.pdf> [Diakses : 20 Juni 2018].

## BIODATA PENULIS



Penulis bernama lengkap Febe Sofia Loren Panjaitan, dilahirkan di Kota Pematangsiantar, 6 Februari 1997 dan merupakan anak keempat dan empat bersaudara. Pada tahun 2014 penulis diterima di Jurusan Sistem Informasi yang tahun 2017 berubah menjadi Departemen Sistem Informasi Institut Teknologi Sepuluh Nopember (ITS) melalui jalur SNMPTN.

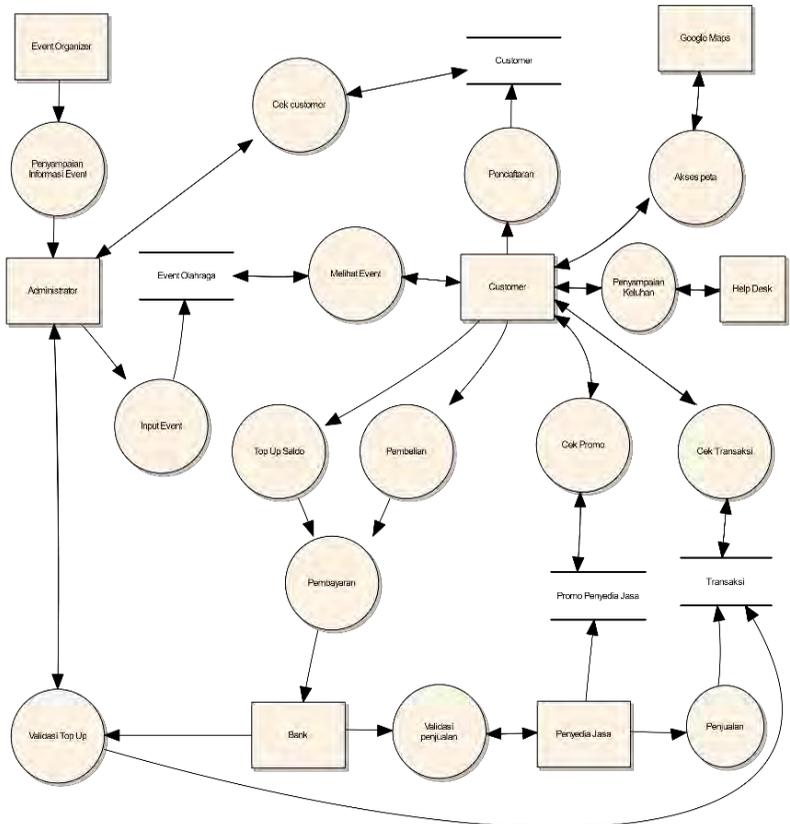
Selama kuliah penulis bergabung dalam beberapa organisasi kemahasiswaan, didalam maupun luar ITS , beberapa kepanitian diantaranya Panitia Validasi SBMPTN 2018 dan beberapa kali bergabung dalam tim proyek diantaranya Pembuatan Masterplan IT kota Probolinggo sebagai surveyor dan analis. Terakhir penulis pernah melakukan kerja praktik di Direktorat Jenderal Bea dan Cukai (DJBC) selama 2 bulan di Tahun 2017.

Pada pengerjaan Tugas Akhir ini, penulis mengambil bidang minat Manajemen Sistem Informasi (MSI) dengan topik *Value engineering* pada lingkup perencanaan TI. Topik diambil dari jurusan Teknik Sipil dan menjadi topik baru di Jurusan Sistem Informasi. Untuk keperluan penelitian, dapat menghubungi penulis melalui e-mail : febepanjaitan@gmail.com.

*Halaman ini sengaja dikosongkan*

# LAMPIRAN A

Lampiran ini merupakan lampiran DFP (Diagram Flow Diagram). Diagram ini dibuat untuk memperjelas kaitan antar masing-masing komponen didalam sistem untuk menjalankan fungsi-fungsi pada aplikasi.



*Halaman ini sengaja dikosongkan*

## **LAMPIRAN B**

Lampiran ini merupakan hasil akhir dari implementasi *Value engineering* dalam proyek, disebut *Value Engineering Change Proposal*. Lampiran ini akan dibawakan sebagai rangkuman kepada pihak perusahaan pemberi proyek untuk direview dan disetujui.

*Halaman ini sengaja dikosongkan*

## VALUE ENGINEERING CHANGE PROPOSAL SUMMARY

**VECP SUBMISSION DATE :**

**VECP SUBMISSION (Check one) :**  CONCEPTUAL \_\_\_ FORMAL \_\_\_ DOESN'T CONCEPTUAL & FORMAL

**PROJECT INFORMATION :**

COMPANY Do : -	LETTING DATE : /06/2018
DESCRIPTION : Replacement of programming tools with visual programming.	ORIGINAL : Rp. 439.300.000
COMPANY : PT.XYZ	PROJECT COSTS : Indonesia
	COUNTRY : Indonesia

**VECP INFORMATION :**

**SHORT DESCRIPTION OF VECP :** Replacement of programming tools with visual programming for savings cost.

(A) TYPE OF VECP? (Check one):  COST SAVINGS \_\_\_ TIME SAVINGS ONLY

(B) IS THERE A DATE BY WHICH THE VECP WORK MUST BE AUTHORIZED ? :  YES  NO DATE? :-

(C) ANY NEW OR EXISTING PAY ITEMS REQUIRING AGREED PRICES? :  YES  NO HOW MANY? : 1

(D) ANY PAY ITEMS WITH LONG LEAD TIMES THAT REQUIRE APPROVAL? :  YES  NO HOW MANY? : 1

(Note : Describe items A. though D in further detail in >COMMENTS= as appropriate)

(E) PROJECT COST ORIGINAL	Rp. 439.300.000	(Note :This is the original costs estimation by PT.XYZ)
(F) PROJECT COST W/O VECP	Rp. 511.100.000	(Note : This costs are totally count from 11 functions which wants to build on this app)
(G) PROJECT COST W VECP	Rp. 335.900.000	(Note : Count by implementing the fourth scenario – using visual programming)
(H) VECP PROJECT SAVINGS	Rp. 175.200.000	(Note : Equals item F. minus item G.)
(I) VECP DESIGN COST	Rp. 0	
(J) DIRECT COST SAVINGS	Rp. 175.200.000	(Note : Equals item H. minus item I.)

**COMMENT :**

1. The letting date was blanked by the writer, because in this case, the writer only communicated with the third, and the date would be determined by the third with the company of the project, and also the date of authorize.
2. The original costs of the project was determined by PT.XYZ as the third. This kind of estimation was pointed for application development. In the other hand, for the project without VECP was estimated by the writer. This estimated was based on valuation of the price of applications's feature, that's why the calculation's could be more expensive because it has passed the process of applicarion breakdown in the functional needs.
3. The cost project with VECP was estimated by the writer. This estimation came from the script that utilized the visual programming from decreased the work time of programmer.Example : From 45 to 7 days. The price calculation was done with multiply the amount of human resources, work time,and salary of the worker everyday. Then with decreased the work time,it could be decreased the cost. In this case, programmer was required in the each function of development, then more decreasing work time to do,there more cost could be saved
4. VECP Projects savings are the equals items F minus G. It is the costs that third party can savings by implementing this scenario.
5. No design cost would be applicable to this VECP.



## VECP – ITEMIZED PROJECT SAVINGS

VECP DESCRIPTION : VISUAL PROGRAMMING SCENARIO

NO.	FUNGSIONAL REQUIREMENT	INITIAL COST (s)	FINAL COST (s)	CHANGE IN FUND (S)	
				DECREASE	INCREASE
1.	<b>Login</b>	Rp 25.800.000	Rp 25.800.000	-	-
2.	<b>Saldo Information</b>	Rp 61.750.000	Rp 34.150.000	Rp 27.600.000	-
3.	<b>Transaction Activities Information</b>	Rp 60.350.000	Rp 32.750.000	Rp 27.600.000	-
4.	<b>Ordered activities</b>	Rp 136.900.000	Rp 69.300.000	Rp 67.600.000	-
5.	<b>Pick up Information</b>	Rp 34.500.000	Rp 20.900.000	Rp 13.600.000	-
6.	<b>Arrounded Hotel Information</b>	Rp 43.750.000	Rp 34.150.000	Rp 9.600.000	-
7.	<b>Arroundedn Official Partner's mall information</b>	Rp 43.750.000	Rp 34.150.000	Rp 9.600.000	-
8.	Time and Location Information	Rp 24.000.000	Rp 24.000.000	-	-
9.	Help-desk	Rp 27.700.000	Rp 27.700.000	-	-
10.	<b>Location Information</b>	Rp 31.000.000	Rp 11.400.000	Rp 19.600.000	-
11.	Multi-languange	Rp 21.600.000	Rp 21.600.000	-	-
<b>TOTAL</b>		Rp 511.100.000	Rp 335.900.000	Rp 175.200.000	-



