

TUGAS AKHIR - IS184853

IMPLEMENTASI PENDEKATAN *VALUE ENGINEERING* UNTUK MENGHITUNG PENGHEMATAN BIAYA PADA TAHAP PERENCANAAN PROYEK PEMBANGUNAN APLIKASI *E-PERFORMANCE*

IMPLEMENTATION OF VALUE ENGINEERING APPROACH TO CALCULATE SAVINGS COST IN THE E-PERFORMANCE APPLICATION DEVELOPMENT PROJECT PLANNING

Safira Lady Al Islami
0521 15 4000 0035

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 2019

TUGAS AKHIR – IS184853

IMPLEMENTASI PENDEKATAN *VALUE ENGINEERING* UNTUK MENGHITUNG PENGHEMATAN BIAYA PADA TAHAP PERENCANAAN PROYEK PEMBANGUNAN APLIKASI *E-PERFORMANCE*

Safira Lady Al Islami
0521 15 4000 0035

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 2019



FINAL PROJECT - IS184853

***IMPLEMENTATION OF VALUE ENGINEERING
APPROACH TO CALCULATE SAVINGS COST IN
THE E-PERFORMANCE APPLICATION DEVELOPMENT
PROJECT PLANNING***

Safira Lady Al Islami
0521 15 4000 0035

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 2019

LEMBAR PENGESAHAN
IMPLEMENTASI PENDEKATAN VALUE
ENGINEERING UNTUK MENGHITUNG
PENGHEMATAN BIAYA PADA TAHAP
PERENCANAAN PROYEK PEMBANGUNAN
APLIKASI E-PERFORMANCE

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:

SAFIRA LADY AL ISLAMI
0521 15 4000 0035

Surabaya, 2019

KEPALA

DEPARTEMEN SISTEM INFORMASI



Mahendrawati ER, ST, M.Sc, P.hd
NIP 197610112 006042 001

LEMBAR PERSETUJUAN

IMPLEMENTASI PENDEKATAN VALUE ENGINEERING UNTUK MENGHITUNG PENGHEMATAN BIAYA PADA TAHAP PERENCANAAN PROYEK PEMBANGUNAN APLIKASI E-PERFORMANCE

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:


SAFIRA LADY AL ISLAMI
0521154000035

Disetujui Tim Penguji: Tanggal Ujian : 08 Juli 2019
Periode Wisuda : September 2019

Ir. Khakim Ghozali, M.MT


(Pembimbing 1)

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

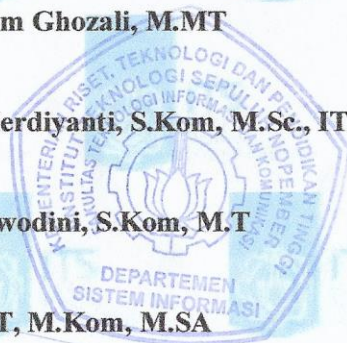

(Pembimbing 2)

Febby Artwodini, S.Kom, M.T


(Penguji 1)

Sholiq, ST, M.Kom, M.SA


(Penguji 2)



**IMPLEMENTASI PENDEKATAN VALUE ENGI-
NEERING UNTUK MENGHITUNG PENGHEMA-
TAN BIAYA PADA TAHAP PERENCANAAN
PROYEK PEMBANGUNAN APLIKASI E-
PERFORMANCE**

Nama Mahasiswa : SAFIRA LADY AL ISLAMI
NRP : 0521 15 4000 0035
Departemen : Sistem Informasi FTIK-ITS
Pembimbing I : Ir. Khakim Ghozali, M.MT
Pembimbing II : Anisah Herdiyanti, S.Kom, M.Sc.,
ITIL.

ABSTRAK

Kurangnya pengetahuan user terhadap produk TI yang mereka buat mengakibatkan tingkat overrun pada pelaksanaan proyek TI meningkat, studi terakhir menunjukkan bahwa dalam kurun waktu 2010-2014 terdapat 52,7% proyek TI yang dilaporkan terlaksana namun melebihi waktu yang telah ditentukan. Hal tersebut sangat berdampak pada biaya yang dikeluarkan. Dari permasalahan ini terdapat sebuah peluang untuk mengoptimalkan penggunaan biaya dan meningkatkan nilai dari sebuah proyek TI.

Penerapan value engineering sebagai sebuah framework untuk meningkatkan keefektifitasan proyek dapat dilakukan, mengingat studi tersebut masih sangat jarang digunakan dibidang TI. Value Engineering ini dimanfaatkan untuk meningkatkan penghematan biaya pada proyek TI yang ada di pemerintah daerah. Proyek TI tersebut berupa pembangunan aplikasi e-Performance untuk kabupaten di daerah XYZ. Dengan berbekal Kerangka Acuan Kerja dari tim pengembang, penelitian ini berfokus untuk mengetahui apakah Kerangka Acuan Kerja yang diajukan tim pengembang dapat diminimalkan dari segi biaya tanpa mengurangi nilai dari sebuah aplikasi.

Proses value engineering yang dimanfaatkan sebatas hanya pada tahap perencanaan proyek. Hasil dari penelitian ini

berupa dokumen rekomendasi perubahan (VECP) yang dapat dijadikan sebagai pertimbangan oleh tim pengembang dalam melaksanakan proyek.

Kata kunci: *Value Engineering, e-Performance, penghematan biaya, Proyek TI*

**IMPLEMENTATION OF VALUE ENGINEERING
APPROACH TO CALCULATE SAVINGS COST IN
THE E-PERFORMANCE APPLIATION DEVEL-
OPMENT PROJECT PLANNING**

Sudent Name : SAFIRA LADY AL ISLAMI
Student Number : 0521 15 4000 0035
Departement : Sistem Informasi FTIK-ITS
Supervisor I : Ir. Khakim Ghozali, M.MT
Supervisor II : Anisah Herdiyanti, S.Kom, M.Sc.,
ITIL.

ABSTRACT

Lack of user knowledge of IT products results in an increase in the level of overrun in IT project implementation, a recent study shows that in 2010-2014 there were 52.7% of IT projects reported to have been carried out but exceeded the deadline. This can result in costs incurred. Furthermore, these problems produce opportunities to optimize the use of costs and increase the value of IT projects .The application of value engineering as a framework to improve project effectiveness is carried out, considering that this study is still rarely used in the IT field. Value Engineering is used to increase cost savings on existing IT projects in local governments. The IT project is in the form of building an e-Performance application for patents in the XYZ area. Armed with the project activity document from the development team, this study focuses on finding out whether the project activity document submitted by the development team can be minimized from the cost side without removing the value of the application.

The value engineering process used is limited only to the project planning stage. The results of this study are in the form of a change recommendation document (VECP) which can be used as a consideration by the development team in implementing the project.

Keywords: *Value Engineering, e-Performance, IT Project, cost saving*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Segala puji dan syukur terpanjatkan kepada Allah SWT, Tuhan Yang Maha Esa atas izin-Nya peneliti dapat menyelesaikan sebagian penelitian dan penuntasan Tugas Akhir ini dengan judul

“IMPLEMENTASI PENDEKATAN *VALUE ENGINEERING* UNTUK MENGHITUNG PENGHEMATAN BIAYA PADA TAHAP PERENCANAAN PROYEK PEMBANGUNAN APLIKASI *E-PERFORMANCE*”

dalam menempuh langkah untuk memperoleh gelar Sarjana Komputer, Departemen Sistem Informasi – Institut Teknologi Sepuluh Nopember Surabaya.

Selama melakukan penelitian dan menyelesaikan buku tugas akhir, peneliti tidak sendiri karena telah dibantu dan didukung orang terdekat. Oleh sebab itu, peneliti menyampaikan permintaan permohonan maaf, dan Terimakasih sebanyak banyaknya atas bimbingan, arahan, saran, dukungan dan terutama doa untuk menyelesaikan tugas akhir ini, kepada :

1. Ibu, Ayah dan adik serta keluarga yang selalu tidak bosan – bosannya menyebut nama peneliti didoa mereka disepertiga malam, dan dukungan tiada habisnya kepada peneliti.
2. Bapak Ir. Khakim Ghozali, M.MT. dan Ibu Anisah Herdiyanti, S.Kom, M.Sc., ITIL selaku dosen pembimbing dan tim pengembang pihak ketiga yang selalu sabar dan mencurahkan banyak waktu dan pikiran dalam mengarahkan penyelesaian tugas akhir peneliti.
3. Bapak Sholiq, ST, M.Kom, M.SA, selaku dosen dan Kepala Lab. MSI yang memberikan masukan terhadap metodologi yang peneliti ambil dalam penyelesaian tugas akhir ini.
4. Mas, mbak, senior Febe L Panjaitan, Abdul Aziz dan senior OSIRIS lainnya yang rela membagi waktunya dalam memberikan saran dalam penyelesaian tugas akhir ini.

5. Sholihatur Rahmah (Almh), Prasasti, Andira *senpai*, Fano *senpai*, Almira, Ahsanul dan teman - teman seperguruan “Bimbingan Abah” yang mampu menenangkan dan memberi semangat kepada peneliti.
6. Teman – teman LANNISTER, Lab. MSI, dan *Expanding Brain* yang menjadi tempat penghiburan dan tukar pikiran ketika peneliti membutuhkan ide.
7. Teman - teman seperguruan “Indahnya SI”, yang selalu saling menyemangati, sebagai tempat bertukar pikiran dan diskusi peneliti dalam menyelesaikan buku ini.
8. Terimakasih dan mohon maaf pada Ulfania A, Darul H, dan Farza N selaku tim Peda yang telah memberikan banyak inspirasi kepada peneliti untuk menyelesaikan permasalahan penyelesaian buku tugas akhir.
9. Civitas akademika Departemen Sistem - Informasi Institut Teknologi Sepuluh Nopember dan pihak lain, yang selalu membagi informasi selama peneliti menempuh studi.

Penyusunan penelitian dan tugas akhir ini jauh dari kata sempurna, untuk itu peneliti sangat menerima adanya kritik dan saran demi perbaikan dimasa mendatang. Serta, semoga buku ini dapat menghadirkan kontribusi dan manfaat bagi ilmu pengetahuan dan pembaca.

Surabaya, Juni 2019

Peneliti

DAFTAR ISI

| | |
|---|------|
| ABSTRAK | i |
| ABSTRACT | iii |
| KATA PENGANTAR | v |
| DAFTAR ISI | vii |
| DAFTAR GAMBAR | xi |
| DAFTAR GRAFIK | xiii |
| DAFTAR TABEL | xv |
| DAFTAR ISITILAH | xxi |
| 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 | 5 |
| 1.5.1. Bagi peneliti | 5 |
| 1.5.2. Bagi Tim Pengembang | 5 |
| 1.6. Relevansi | 5 |
| 1.7. Target Luaran | 6 |
| BAB II TINJAUAN PUSTAKA | 7 |
| 2.1. Penelitian Sebelumnya | 7 |
| 2.2. Dasar Teori | 9 |
| 2.2.1. Aplikasi | 9 |
| 2.2.2. <i>E-Performance</i> | 9 |
| 2.2.3. Fungsi | 10 |
| 2.2.4. Nilai | 11 |
| 2.2.5. Proyek | 13 |
| 2.2.6. Teknik Terkait | 16 |
| 2.2.7. <i>Value Engineering</i> | 32 |
| 2.2.8. <i>Value Engineering Change Proposal</i> | 38 |
| BAB III METODOLOGI | 41 |
| 3.1. Diagram Metodologi | 41 |
| 3.2. Uraian Metodologi | 42 |

| | | |
|--------|--|-----|
| 3.2.1. | <i>Value – Analysis</i> | 42 |
| BAB IV | PERANCANGAN | 51 |
| 4.1. | Tahap Perencanaan Proyek | 51 |
| 4.2. | Penjelasan Aplikasi | 52 |
| 4.3. | Value Analysis | 55 |
| 4.3.1. | Information Phase..... | 55 |
| 4.3.2. | Function Analysis Phase | 72 |
| 4.3.3. | Creative Analysis Phase | 82 |
| 4.3.4. | Evaluation Phase | 86 |
| BAB V | IMPLEMENTASI | 89 |
| 5.1. | <i>Information Phase</i> | 89 |
| 5.1.1. | Pembahasan Data Eksisting..... | 89 |
| 5.1.2. | Pembahasan Data Koesioner Penilaian COCOMO II..... | 94 |
| 5.2. | <i>Function Phase Analysis</i> | 95 |
| 5.3. | Hambatan penelitian..... | 96 |
| BAB VI | HASIL DAN PEMBAHASAN..... | 99 |
| 6.1. | <i>Information Phase</i> | 99 |
| 6.1.1. | Pengkajian Data Masa Lalu..... | 99 |
| 6.1.2. | Perhitungan Estimasi Biaya dengan Constructive Cost Model (COCOMO II) | 102 |
| 6.1.3. | Perbandingan Estimasi Biaya dengan Penyedia Jasa | 112 |
| 6.2. | <i>Function Phase Analysis</i> | 115 |
| 6.2.1. | Perhitungan Biaya Kebutuhan Fungsional | 115 |
| 6.2.2. | Pemilihan Objek Penelitian | 149 |
| 6.2.3. | Pembuatan FAST Diagram | 151 |
| 6.3. | <i>Creativity Phase</i> | 172 |
| 6.3.1. | Melakukan Update Data Pegawai | 174 |
| 6.3.2. | <i>Input</i> Aktivitas Pegawai | 178 |
| 6.3.3. | Mensetujui Aktivitas Pegawai..... | 181 |
| 6.3.4. | Melakukan pembatalan Aktivitas Pegawai ... | 184 |
| 6.3.5. | Menampilkan Infrormasi Penugasan Pegawai | 187 |
| 6.3.6. | Menampilkan Infrormasi Sasaran Program... | 190 |
| 6.3.7. | Melakukan Penilaian | 193 |

| | | |
|------------------------------------|--|-----|
| 6.3.8. | Melakukan generate rapor..... | 196 |
| 6.3.9. | Melakukan Pencarian berdasarkan Filter | 200 |
| 6.4. | <i>Evaluation Phase</i> | 204 |
| 6.4.1. | Pencarian Indeks Nilai | 207 |
| 6.4.2. | Pemilihan Rekomendasi Alternatif | 208 |
| 6.4.3. | Rekap perbandingan Estimasi Biaya..... | 213 |
| 6.4.4. | Pembuatan <i>Value Engineering Change Proposal</i> | 216 |
| BAB VII KESIMPULAN DAN SARAN | | 217 |
| 7.1. | Kesimpulan | 217 |
| 7.2. | Saran | 218 |
| DAFTAR PUSTAKA | | 221 |
| BIODATA PENELITI | | 227 |
| LAMPIRAN A | | A-1 |
| LAMPIRAN B | | B-1 |
| LAMPIRAN C | | C-1 |

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

| | |
|---|-----|
| Gambar 1.1 Road Map Laboratorium MSI | 6 |
| Gambar 2.1 Siklus SDLC pada Proyek TI | 14 |
| Gambar 2.2 Penggunaan F.A.S.T Diagram..... | 30 |
| Gambar 2.3 Alur Proses Value Engineering | 35 |
| Gambar 4.1 Technical Oriented FAST[5]..... | 79 |
| Gambar 4.2 Customer (TASK) Oriented FAST[5]..... | 80 |
| Gambar 4.3 Contoh penggunaan F.A.S.T Diagram pada pembuatan aplikasi [5] | 81 |
| Gambar 6.1 Tampilan antar muka pembuka aplikasi e-Performance | 101 |
| Gambar 6.2 Tampilan antar muka yang diharapkan untuk halaman rekap aktivitas e-Performance | 102 |
| Gambar 6.3 Tampilan antar muka yang diharapkan pada halaman Pengesahaan aplikasi e-Performance..... | 102 |
| Gambar 6.4 Diagram FAST untuk fungsi Update Data Pegawai | 153 |
| Gambar 6.5 Diagram FAST untuk fungsi Input Aktivitas ... | 155 |
| Gambar 6.6 Diagram FAST untuk fungsi Mengesahkan Aktivitas | 157 |
| Gambar 6.7 Diagram FAST untuk fungsi Membatalkan Aktivitas | 159 |
| Gambar 6.8 Diagram FAST untuk Menampilkan Penugasan | 162 |
| Gambar 6.9 Diagram FAST untuk fitur menampilkan informasi Sasaran Program | 165 |
| Gambar 6.10 Diagram FAST untuk fungsi Melakukan Penilaian pegawai..... | 167 |
| Gambar 6.11 Diagram FAST untuk fungsi Filter pada Pencarian..... | 169 |
| Gambar 6.12 Diagram FAST untuk fungsi Generate Rapor | 171 |
| | |
| Gambar A. 1 Konseptual Data Model untuk Aplikasi e-Performance | A-1 |
| Gambar A. 2 Data Flow Diagram Context Level aplikasi e-Performance | A-2 |
| Gambar A. 3 Data Flow Diagram Level 1 e-Performance... | A-4 |

Halaman ini sengaja dikosongkan

DAFTAR GRAFIK

| | |
|---|-----|
| Grafik 6.1 Perubahan hari pengembangan setiap fitur | 209 |
| Grafik 6.2 Perubahan biaya pengembangan setiap fitur..... | 211 |
| Grafik 6.3 Hubungan waktu dan biaya..... | 212 |

Halaman ini sengaja dikosongkan

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Penelitian Terdahulu | 7 |
| Tabel 2.2 Penilitan terdahulu (2)..... | 8 |
| Tabel 2.3 Penelitian terdahulu (3)..... | 8 |
| Tabel 2.4 Perbandingan Metode Estimasi Biaya..... | 16 |
| Tabel 2.5 Produktivitas application point..... | 21 |
| Tabel 2.6 Nilai yang ada pada <i>Scale Factor</i> | 27 |
| Tabel 3.2 Input, output, dan proses yang ada pada information phase..... | 44 |
| Tabel 3.3 Input, ouput dan proses pada Function Analysis Phase | 45 |
| Tabel 3.4 Input, output dan proses Creativity Phase..... | 47 |
| Tabel 3.5 Input, outpu dan proses yang ada pada Evaluation Phase | 49 |
| Tabel 4.1 Pemetaan kegiatan perencanaan proyek | 51 |
| Tabel 4.2 Kebutuhan fungsional aplikasi | 53 |
| Tabel 4.3 Template tabel hasil pemetaan kategori input..... | 58 |
| Tabel 4.4 Pemetaan kompleksitas masukan dalam perhitungan UFP | 59 |
| Tabel 4.5 Kompleksitas kategori input data element | 60 |
| Tabel 4.6 Template Perhitungan UFP..... | 61 |
| Tabel 4.7 Tabel konversi UFP menjadi SLOC..... | 63 |
| Tabel 4.8 Effort Multiplier dari Early Design dan Post Architecture COCOMO II..... | 65 |
| Tabel 4.9 Prosentase Distribusi usaha pada aktivitas proyek. | 67 |
| Tabel 4.10 Template hasil pendistribusian aktivitas | 67 |
| Tabel 4.11 Standar Gaji yang digunakan dalam proses perhitungan biaya total..... | 68 |
| Tabel 4.12 Template Hasil perhitungan Estimasi Biaya Total | 69 |
| Tabel 4.13 Fungsi Perbandingan..... | 70 |
| Tabel 4.14 Template perbandingan dengan aplikasi dari penyedia lain | 71 |
| Tabel 4.15 Template Detail Pembagian effort dan waktu pengerjaan | 72 |
| Tabel 4.16 Template Estimasi biaya kebutuhan Fungsional dan hasil Bobot tiap Fungsi..... | 73 |

| | |
|---|------------|
| Tabel 4.17 Kategori Ukuran Aplikasi berdasarkan waktu pengembangan..... | 74 |
| Tabel 4.18 Template Penentuan Objek Penelitian..... | 75 |
| Tabel 4.19 Template tabel skenario Effort Multiplier..... | 83 |
| Tabel 4.20 Template tabel hasil skenario Scale Factor | 84 |
| Tabel 4.21 Template Hasil Alternatif Kebutuhan Fungsi..... | 85 |
| Tabel 4.22 Template Hasil Perubahan Biaya berdasarkan Skenario..... | 86 |
| Tabel 4.23 Tenplate Hasil Pencarian Indeks | 87 |
| Tabel 5.1 Kebutuhan Fungsional Aplikasi | 89 |
| Tabel 5.2 Pihak yang terlibat dalam Proyek..... | 90 |
| Tabel 5.3 Tahapan pengerjaan aplikasi | 91 |
| Tabel 5.4 Detail pembagian workload perbulan untuk tiap jabatan | 93 |
| Tabel 5.5 Nilai Scale Factor untuk aplikasi e-Performance ... | 94 |
| Tabel 5.6 Nilai Skala cost driver | 95 |
| Tabel 5.7 Hasil perhitungan Effort Multiplier | 95 |
| Tabel 6.1 Hasil rekap tipe masukan setiap fitur | 105 |
| Tabel 6.2 Hasil penghitungan elemen data dan record data setiap fitur..... | 106 |
| Tabel 6.3 Hasil perhitungan bobot kompleksitas aplikasi | 107 |
| <i>Tabel 6.4 Distribusi prosentase aktivitas</i> | <i>110</i> |
| Tabel 6.5 Hasil perhitungan biaya total pengerjaan proyek pengembangan aplikasi e-Performance | 111 |
| Tabel 6.6 Fungsi pembandingan yang ditawarkan pihak lain .. | 112 |
| Tabel 6.7 Hasil perbandingan fitur dan estimasi biaya terhadap penyedia jasa lain | 115 |
| Tabel 6.8 Hasil perhitungan UFP untuk setiap kebutuhan fungsional | 120 |
| Tabel 6.9 Hasil perhitungan biaya untuk kebutuhan fungsional Update Pegawai..... | 122 |
| Tabel 6.10 Hasil perhitungan biaya untuk kebutuhan fungsional Input Aktivitas | 124 |
| Tabel 6.11 Hasil perhitungan biaya untuk kebutuhan fungsional Melakukan persetujuan aktivitas | 126 |
| Tabel 6.12 Hasil perhitungan biaya untuk kebutuhan fungsional Pembatalan Aktivitas..... | 128 |

| | |
|--|-----|
| Tabel 6.13 Hasil perhitungan biaya untuk kebutuhan fungsional Menampilkan Capaian Aktivitas | 130 |
| Tabel 6.14 Hasil perhitungan biaya untuk kebutuhan fungsional Informasi Perangkat Daerah | 132 |
| Tabel 6.15 Hasil perhitungan biaya untuk kebutuhan fungsional Capaian Kegiatan | 133 |
| Tabel 6.16 Hasil perhitungan biaya untuk kebutuhan fungsional Menampilkan Indikator | 135 |
| Tabel 6.17 Hasil perhitungan biaya untuk kebutuhan fungsional Menampilkan Informasi Pemberian Tugas | 137 |
| Tabel 6.18 Hasil perhitungan biaya untuk kebutuhan fungsional Menampilkan Sasaran Program..... | 139 |
| Tabel 6.19 Hasil perhitungan biaya untuk kebutuhan fungsional Penilaian | 141 |
| Tabel 6.20 Hasil perhitungan biaya untuk kebutuhan fungsional Generate Rapor..... | 143 |
| Tabel 6.21 Hasil perhitungan biaya untuk kebutuhan fungsional Pencarian dengan Filter | 145 |
| Tabel 6.22 Hasil perhitungan biaya untuk kebutuhan fungsional Login | 146 |
| Tabel 6.23 Hasil perhitungan biaya untuk kebutuhan fungsional Ganti Password..... | 148 |
| Tabel 6.24 Hasil perhitungan Bobot tiap kebutuhan fungsional berdasarkan estimasi biaya yang diperoleh | 149 |
| Tabel 6.25 Hasil perubahan Effort Multiplier untuk alternatif 1 | 172 |
| Tabel 6.26 Hasil perubahan perhitungan Scale Factor untuk alternatif 2 | 174 |
| Tabel 6.27 Hasil perhitungan biaya dari perubahan effort alternatif 1 | 176 |
| Tabel 6.28 Hasil perhitungan biaya dari perubahan effort alternatif 1 | 177 |
| Tabel 6.29 Hasil perhitungan biaya dari perubahan effort alternatif 1 | 179 |
| Tabel 6.30 Hasil perhitungan biaya dari perubahan effort alternatif 2 | 180 |

| | |
|--|-----|
| Tabel 6.31 Hasil perhitungan biaya dari perubahan effort alternatif 1..... | 182 |
| Tabel 6.32 Hasil perhitungan biaya dari perubahan effort alternatif 2..... | 183 |
| Tabel 6.33 Hasil perhitungan biaya dari perubahan effort alternatif 1..... | 185 |
| Tabel 6.34 Hasil perhitungan biaya dari perubahan effort alternatif 2..... | 186 |
| Tabel 6.35 Hasil perhitungan biaya dari perubahan effort alternatif 1..... | 188 |
| Tabel 6.36 Hasil perhitungan biaya dari perubahan effort alternatif 2..... | 189 |
| Tabel 6.37 Hasil perhitungan biaya dari perubahan effort alternatif 1..... | 191 |
| Tabel 6.38 Hasil perhitungan biaya dari perubahan effort alternatif 2..... | 193 |
| Tabel 6.39 Hasil perhitungan biaya dari perubahan effort alternatif 1..... | 194 |
| Tabel 6.40 Hasil perhitungan biaya dari perubahan effort alternatif 2..... | 196 |
| Tabel 6.41 Hasil perhitungan biaya dari perubahan effort alternatif 1..... | 197 |
| Tabel 6.42 Hasil perhitungan biaya dari perubahan effort alternatif 2..... | 199 |
| Tabel 6.43 Hasil perhitungan biaya dari perubahan effort alternatif 1..... | 201 |
| Tabel 6.44 Hasil perhitungan biaya dari perubahan effort alternatif 2..... | 202 |
| Tabel 6.45 Hasil rekam skenario Creativity Phase..... | 203 |
| Tabel 6.46 Ringkasan perbandingan biaya dari skenario 1 .. | 205 |
| Tabel 6.47 Ringkasan perbandingan biaya dari skenario 2 .. | 206 |
| Tabel 6.48 Hasil alternatif kebutuhan fungsional..... | 207 |
| Tabel 6.49 Perbandingan biaya yang dihasilkan tiap metode estimasi dan skenario..... | 214 |

| | |
|---|-----|
| Tabel A. 1 Daftar data flow input dan output pada DFD level konteks e-Performance | A-2 |
|---|-----|

| | | |
|----------------|---|-----|
| Tabel C. 1 | Daftar pertanyaan penilaian Cost Driver | |
| COCOMO II..... | | C-2 |

Halaman ini sengaja dikosongkan

DAFTAR ISITILAH

Bagian ini menjelaskan mengenai pengertian, istilah dan singkatan yang digunakan pada perhitungan COCOMO II *Early Design Model*.

- ACAP : *Analyst Capability Cost Driver*. Faktor biaya yang berasal dari kemampuan seorang analis yang terlibat pada sebuah proyek. Faktor yang dinilai terdiri dari kemampuan analisis dan desain, efisiensi dan ketelitian serta faktor komunikasi dan kerja sama seorang analis
- APEX : *Application Experience*. Faktor yang bergantung dari tingkat pengalaman pengguna aplikasi dari tim pengembang pada tim proyek.
- CPLX : *Product Complexity Cost Driver* adalah salah satu faktor biaya disebabkan oleh tingkat kerumitan sebuah produk / sistem.
- DATA : *Database Size Cost Driver* faktor penyumbang biaya yang berasal dari ukuran *database* yang dibutuhkan dalam pengembangan sebuah sistem.
- DOCU : *Documentation Match to Lifecycle Needs Cost Driver* faktor penyumbang biaya yang berasal dari kesesuaian dokumentasi proyek berdasarkan siklus pengembangan yang sedang digunakan dalam pengembangan sebuah fitur
- FCIL : *Facilites* faktor biaya gabungan dari dua model (*Early design* dan *Post design*)
- LTEX : *Language and Tool Experience Cost Driver* faktor biaya yang diukur dari tingkat kerumitan penggunaan bahasa pemrograman

- dan pengalaman dalam menggunakan *software tool* yang digunakan oleh tim proyek.
- PCAP : *Programmer Capability Cost Driver*. Faktor biaya yang berasal dari kemampuan seorang programmer
- PCON : *Personnel Continuity Cost Driver*. Faktor yang mempengaruhi biaya berasal dari tingkat pergantian anggota tim proyek tiap tahun (*annual personnel turnover*).
- PDIF : *Platform Difficulty: composite Cost Driver for Early Design model* kombinasi faktor biaya yang berasal dari model *Post design* dan *Early Design* (TIME, STOR, PVOL) .
- PREX : *Personnel Experience: composite Cost Driver for Early Design model* kombinasi faktor biaya berasal dari model *Post design* dan *Early Design* (APEX, PLEX, LTEX)
- PVOL : *Platform Volatility Cost Driver* faktor biaya yang disebabkan dari kerumitan *hardware* dan *software* untuk menjalankan sistem yang akan dibangun.
- RCPX : *Product Reliability and Complexity: composite Cost Driver for Early Design model* kombinasi faktor biaya yang berasal dari model *Post design* dan *Early Design* (RELY, DATA, CPLX)
- RELY : *Required Software Reliability Cost Driver* faktor biaya yang diukur dari kemampuan sistem yang akan dibuat berdasarkan periode waktu tertentu
- RUSE : *Developed for Reusability Cost Driver* faktor biaya berasal dari fitur/ komponen/ baris kode yang digunakan kembali pada

pembangunan sistem baru.

Scale Factor : Sebuah karakteristik khusus dari pengembangan perangkat lunak yang memiliki efek eksponensial pengembangan seperti *precedents* (proyek terdahulu), kematangan proses sebuah sistem.

STOR : *Main Storage Constraint Cost Driver* faktor biaya yang dinilai dari tingkat besarnya penyimpanan yang digunakan untuk sistem yang akan dibangun

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

Penerapan teknologi menjadi salah satu masalah yang sering terjadi pada organisasi. Dalam pencapaian tujuan dan keunggulan kompetitif, perusahaan harus memberi perhatian khusus terhadap perubahan teknologi yang telah menjadi hal lumrah pada saat ini. Perubahan yang terjadi di lapangan membuat perusahaan harus mencari solusi berupa strategi yang tepat dalam membantu meminimalkan sumber daya kebutuhan, waktu dan usaha[1]. Salah satu solusi tersebut adalah mengatasi manajemen pengaturan teknologi informasi yang baik. Perangsangan perusahaan dalam penggunaan serta pembiasaan pemanfaatan teknologi baru pada proses bisnis perusahaan juga dapat dijadikan sebagai solusi dari permasalahan tersebut[2]. Usaha lainnya adalah dengan membuat berbagai pembuatan proyek yang terkait dengan sistem/teknologi informasi yang dapat memberikan nilai pada setiap masalah yang ada di perusahaan mulai dari tingkat operasional hingga tingkat strategis.

Pembuatan teknologi pada masa ini tidak sepenuhnya berjalan mulus dibanding dari proyek proyek lain yang biasa dikerjakan oleh perusahaan. Seperti halnya pembuatan proyek teknologi yang terhenti atau terbengkalai. Hasil penelitian yang dilakukan oleh Standish Group menunjukkan dalam kurun waktu 2010-2014 terdapat sebanyak 31,1 % proyek TI yang dibatalkan sebelum proyek tersebut rampung pengerjaannya. Sebanyak 52,7% proyek juga dilaporkan telah menghabiskan biaya lebih dari anggaran yang telah ditetapkan. Dilihat dari sisi kesuksesannya, sebesar 16,2% saja software TI yang diselesaikan sesuai waktu dan biaya yang telah disepakati[3]. Salah satu penyebab dari permasalahan ini adalah

sedikitnya pengetahuan pengguna (dalam hal ini adalah pelanggan) terhadap nilai yang dihasilkan dari software aplikasi TI yang dibuat. Pengetahuan yang dimaksud adalah pengguna atau user belum mengetahui betul perbedaan antara keinginan dan kebutuhan yang harus ada pada aplikasi yang sedang dikembangkan. Kebutuhan terkait fungsi aplikasi sering bertambah dan berubah akibat keinginan pelanggan pada nilai (value) yang terjadi selama pembuatan aplikasi dilaksanakan[4].

Pengertian nilai yang dimaksud adalah perbandingan kualitas dan biaya. Nilai tersebut dapat berupa otomatisasi yang berdampak pada pengurangan biaya tenaga kerja, reduksi waktu pengerjaan yang jika dikonversi mampu menghasilkan penghematan, serta waktu pengambilan keputusan yang lebih cepat[5].

Hal ini kemudian dipandang sebagai sebuah peluang untuk meningkatkan nilai dari sebuah proyek yang berfokus pada pengurangan biaya beserta pemaksimalan kinerja proyek. Dengan memanfaatkan *Value Engineering* ini diharapkan menjadi terobosan baru untuk mewujudkan peluang tersebut. *Value Engineering* sendiri berupa kerangka kerja yang dapat dimanfaatkan untuk melakukan kegiatan peningkatan keefektifitasan kinerja sebuah proyek. Namun penerapannya pada proyek TI masih terbilang sangat jarang. Dikarenakan penentuan harga dari sebuah aplikasi sangat kompleks, perlu digunakan teknik tertentu yang dipadukan pada setiap proses yang ada di dalam *Value Engineering*.

Penelitian ini adalah penelitian lanjutan mengenai pendekatan *Value Engineering* pada optimalisasi biaya dengan metode estimasi *Function Point Analysis*. Namun metode tersebut ditemukan adanya beberapa kelemahan seperti kebutuhan data yang harus terlebih dahulu ditentukan dan harus sudah ditentukan di awal sebelum realisasi proyek[5]. Hal ini membuat penggunaan metode estimasi biaya *Function Point Analysis* diganti dengan metode *COCOMO II*, sesuai pertimbangan bahwa metode ini dapat diterapkan karena tingkat akurasi perhitungan dari teknik ini lebih tinggi, dikarenakan banyak-

nya penggunaan variable selama proses perhitungan[6]. Pertimbangan lainnya adalah COCOMO II mampu beradaptasi pada ukuran proyek, dan ukuran jenis aplikasi yang dikembangkan dibanding metode estimasi biaya yang diterapkan sebelumnya[7].

Dari pemaparan latar belakang yang telah dijelaskan peneliti mencoba untuk menggunakan pendekatan *Value Engineering* yang mengambil studi kasus pada tahap perencanaan proyek pembuatan aplikasi *e-Performance*. Aplikasi *e-Performance* yang dimaksud berupa aplikasi sistem informasi manajemen yang digunakan untuk menilai kinerja pegawai daerah setempat secara transparan objektif dan terukur. Secara umum aplikasi ini terdiri dari beberapa fungsi. Setiap fungsi memiliki batasan berupa hak akses yang diklasifikasi berdasarkan tingkat jabatan[8]. Aplikasi ini nantinya akan dikembangkan oleh pihak ketiga. Pembuatan aplikasi tersebut dilakukan mulai dari identifikasi kebutuhan aplikasi hingga proses *maintenance* setelah pengimplementasian aplikasi.

Dengan memanfaatkan pendekatan *Value Engineering* bertujuan untuk mencari mencari hasil yang mendekati optimal pada fungsi terbaik antara biaya, keandalan dan kinerja sebuah proyek dengan berfokus pada redesign biaya[9]. Luaran yang dihasilkan pada pendekatan ini nantinya adalah menyediakan solusi yang berisi skenario penghematan dan pengoptimalan durasi proyek dengan detail lain terkait gaji karyawan, waktu lembur dan sumber daya yang saling berkaitan dan disusun menjadi dokumen usulan perubahan berupa Value Engineering Change Proposal (VECP) [10].

1.2. Perumusan Masalah

Rumusan masalah untuk Tugas Akhir ini adalah hasil analisis Value Engineering pada pembuatan aplikasi *e-Performance*. Adapun hasil analisis tersebut dapat menjawab pertanyaan tersebut :

1. Seberapa besar estimasi biaya perancangan proyek pembuatan aplikasi *e-Performance* menggunakan

metode COCOMO II pada Tahap *Information Phase*?

2. Bagaimana F.A.S.T Diagram yang dibuat pada Tahap *Function Analysis Phase*?
3. Bagaimana hasil skenario yang dibuat pada Tahap *Creativity Phase*?
4. Berapa besar penghematan biaya yang dapat diperoleh pada implementasi tahap *Evaluation Phase*?

1.3. Batasan Masalah

Dari permasalahan yang telah dijabarkan, batasan masalah diperlukan untuk mengetahui ruang lingkup masalah yang terlalu luas sehingga penelitian lebih fokus untuk dilakukan. Berikut adalah batasan masalah yang digunakan :

1. Penelitian ini terbatas pada tahapan perencanaan proyek.
2. Hasil atau luaran terbatas pada usulan perubahan proposal, tidak sampai pada implementasi usulan keberlangsungan pembuatan aplikasi *e-Performance*.
3. Untuk semua estimasi biaya yang digunakan pada pendekatan *Value Engineering* menggunakan metode COCOMO II
4. Data masukan awal terbatas pada KAK (Kerangka Acuan Kerja) dari tim pengembang

1.4. Tujuan Tugas Akhir

Tujuan utama dari pembuatan tugas akhir tentang pembuatan aplikasi ini adalah sebagai berikut:

1. Solusi (skenario) dari usulan perubahan kinerja sesuai dengan, biaya, sumber daya dan waktu pengerjaan yang dihitung dengan pendekatan *Value Engineering*.
2. Dokumen VECP (*Value Engineering Change Proposal*) dari solusi (skenario) yang terpilih.

1.5. Manfaat Penelitian

Berikut manfaat yang diperoleh, dengan melihat dari dua sudut pandang, yaitu sudut pandang peneliti dan pihak pengembang aplikasi :

1.5.1. Bagi peneliti

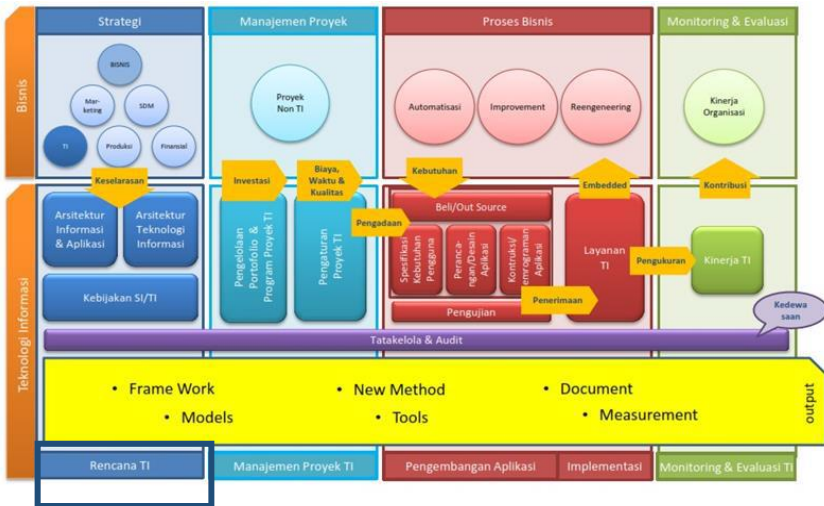
1. Menambah wawasan pengalaman mengenai implementasi penelitian terkait dengan pendekatan *Value Engineering* pada Proyek Teknologi Informasi

1.5.2. Bagi Tim Pengembang

1. Dapat menambah solusi – solusi skenario penghematan biaya pada usulan biaya pembuatan proyek TI pada aplikasi yang berkaitan dengan tata kelola pemerintah baik daerah maupun pusat.

1.6. Relevansi

Tugas Akhir ini memiliki relevansi terhadap beberapa matakuliah yang ada pada Departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi (FTIK), ITS. Usulan Tugas Akhir ini berkaitan dengan matakuliah Statistika, Manajemen Investasi TI, Manajemen Proyek TI. Tugas Akhir ini masuk kedalam lingkup Laboratorium Manajemen Sistem Informasi (MSI) dalam ranah Rencana TI (Departemen Sistem Informasi, 2017).



Gambar 1.1 Road Map Laboratorium MSI

1.7. Target Luaran

Target luaran dari pengerjaan tugas akhir ini adalah :

1. Dokumen perubahan *Value Engineering Change Proposal* yang digunakan untuk dokumentasi perubahan biaya/ waktu pada sebuah pelaksanaan proyek.

BAB II TINJAUAN PUSTAKA

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

2.1. Penelitian Sebelumnya

Terdapat beberapa penelitian yang memiliki topik sejenis dengan penelitian ini diantaranya :

Tabel 2.1 Penelitian Terdahulu

| | |
|----------------------------|--|
| Judul Penelitian | Penyusunan Alternatif Solusi Fungsi – Fungsi yang Optimal pada Tahap Perencanaan Aplikasi XYZ dengan Pendekatan <i>Value Engineering</i> [5] |
| Metode | <i>Value Engineering</i> |
| Peneliti | Febe Sofia L Panjaitan |
| Hasil yang didapatkan | - Kerangka kerja <i>Value Engineering</i> terkait rekomendasi harga akhir platform aplikasi |
| Hubungan dengan penelitian | - Berisi kerangka kerja yang ada pada <i>Value Engineering</i> - Langkah – langkah penyusunan VECP sesuai dengan pada <i>Value Engineering</i> untuk Proyek TI - Pembuatan penyusunan pada teknik yang digunakan dalam <i>Value Engineering</i> (FAST Diagram, Pencarian Indeks Nilai) |

Tabel 2.2 Penelitian terdahulu (2)

| | |
|----------------------------|--|
| Judul Penelitian | COCOMO II Model Definision Manual[11] |
| Metode | COCOMO II |
| Peneliti | Barry Boehm, Chris Abts, Brad Clark, Sunita Devnami- Chulani |
| Hasil yang didapatkan | <ul style="list-style-type: none"> - Memberikan strategi dan keterhubungan antara COCOMO II dengan pengembangan estimasi usaha - Memberikan komposisi model aplikasi - Menghasilkan 2 Model untuk melakukan estimasi (The Early Design Model, Th Porst – Architecture Design Model) |
| Hubungan dengan penelitian | - Prosedur perhitungan Personnel Direct Cost yang terdiri dari perhitungan biaya per-orang, biaya kompensasi dan profit |

Tabel 2.3 Penelitian terdahulu (3)

| | |
|--------------------|---|
| Judul Penelitian | A Variant of COCOMO II for Improved Software Effort Estimation[12] |
| Metode | COCOMO II |
| Peneliti | Ziyad T. Abdulmehdi, M. S. Saleem Basha, Mohamed Jameel, and P. Dhavachelvan |
| Hasil yang didapat | <ul style="list-style-type: none"> - Beberapa pendekatan untuk model estimasi COCOMO II dengan melakukan <i>re-defining</i> usaha (<i>effort multiplier</i>), dan beberapa skala faktor untuk meningkatkan keakuratan COCOMO II - Menggunakan grafik yang merepresentasikan peningkatan skema estimasi COCOMO II dan juga membandingkan |

| | |
|---------------------|--|
| | tingkat error pada COCOMO II tradisional dengan model COCOMO II yang telah <i>diupdate</i> |
| Hubungan Penelitian | - Sebagai acuan informasi terkait perbandingan model estimasi biaya lain dengan COCOMO II |

2.2. Dasar Teori

Subbab ini menjelaskan mengenai konsep secara deskriptif mengenai teori – teori yang berkaitan dengan penelitian dan dijadikan sebagai pengetahuan dasar teoritis.

2.2.1. Aplikasi

Aplikasi sendiri memiliki pengertian sebagai suatu program yang siap digunakan dan dibuat untuk melaksanakan suatu fungsi atau akitvitas tertentu bagi pengguna jasa aplikasi serta pengguna aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Bergantung pada aktivitas yang dilakukan, aplikasi dapat memanipulasi teks, angka , audio grafik, atau kombinasi dari elemen ini. Beberapa paket aplikasi fokus pada satu tugas, seperti pengolah kata; lainnya aplikasi perangkat lunak terintegrasi berisi sekumpulan aplikasi[13].

Aplikasi dibundle dengan komputer dan perangkat lunak atau diterbitkan secara terpisah, serta dapat dikodekan sebagai aplikasi open-source atau universitas[9].

2.2.2. *E-Performance*

1. Pengertian *E-Performance*
e-Performance adalah sistem informasi manajemen kinerja dalam rangka penilaian presetasi kinerja pegawai yang lebih objektif, terukur, akuntabel, partisipasi dan transparan sehingga mampu mewujudkan pembinaan pegawai berdasarkan presetasi kerja dan sistem karir kerja PNS di lingkungan pemerintah[8].
2. Manfaat dan Fungsi penerapan *E-Performance*

Penggunaan sistem ini pada pemerintah daerah[14] :

- Memberikan informasi data Pegawai Negeri Sipil Daerah (PNSD)
- Memonitor aktivitas Pegawai Negeri Sipil Daerah (PNSD)
- Mengukur kinerja Pegawai Negeri Sipil Daerah (PNSD)
- Sebagai dasar pemberian Tunjangan Kinerja bagi Pegawai Negeri Sipil Daerah (PNSD)

3. Fungsi E-Performance

Beberapa fungsi yang ada pada sistem ini berlaku sama sesuai peraturan dan perundang undangan yang mengatur adanya aplikasi tersebut (PERPU No.46 th 2011). Fungsi yang dimaksud seperti :

- Untuk meningkatkan kinerja pegawai dengan pemantauan terhadap aktivitas yang mereka lakukan telah sesuai atau tidak
- Untuk melakukan penilaian terhadap masing masing jabatan yang ada pada pegawai daerah baik atasan maupun bawahan secara objektif.

2.2.3. Fungsi

Pengertian fungsi ditinjau dari analisis fungsi pada *Fuction Analsi, Value Engineering* adalah :

2.2.3.1. Pengertian Fungsi

Pada analisis fungsi, fungsi di gambarkan menjadi 2 definisi, dilihat dari segi kata kerja (verb) dan kata benda (noun) untuk mendefinisikan kebutuhan proyek, produk atau layanan. Dua kata yang digunakan untuk menggambarkan fungsi termasuk kata kerja aktif dan kata benda terukur. Nomina terukur mengidentifikasi sesuatu yang dapat dijelaskan dan dikuantifikasi. Meskipun fungsi proyek, produk atau layanan disediakan dalam bentuk paragraph deskriptif, definisi kata-kata benda dapat menjelaskan dengan ringkas dan jelas pada fungsi di waktu tertentu.

Seperti dalam kasus parit jalan raya, salah satu fungsi parit atau selokan adalah “menyalurkan air”. Kata saluran adalah kata kerja aktif, dan kata air adalah kata benda yang dapat diukur karena jumlah dan jenis air dapat digambarkan dan diukur. Jika parit atau selokan jalan raya juga diperlukan untuk “mendukung habitat” area focus fungsi parit mungkin berubah[15].

Fungsi diperoleh dari orientasi pelanggan. Ketika pelanggan mampu mendeskripsikan keinginan mereka, maka kebutuhan atau sumber daya dari deskripsi pelanggan terpenuhi[5].

2.2.3.2. Karakteristik Fungsi

Fungsi memiliki beberapa karakteristik dasar yaitu :

1. Fungsi mengacu pada pengguna atau performans
2. Sebuah produk memiliki beberapa fungsi
3. Fungsi merupakan solusi yang independent
4. Masing-masing fungsi pada produk yang sama adalah independent

2.2.4. Nilai

Nilai dari pengertian secara luas adalah sebagai pengembalian yang adil atau dalam barang, jasa atau uang untuk sesuatu yang dipertukarkan[16].

2.2.4.1. Pengertian Nilai

Nilai didefinisikan sebagai pengembalian yang adil atau setara dalam barang, jasa atau uang untuk sesuatu yang dipertukarkan. Dengan kata lain , nilai adalah rasio fungsi terhadap biaya; dimana ketika nilai meningkat akan berdampak pada meningkatnya fungsi atau terjadi pengurangan biaya, atau oleh keduanya[16].

Nilai dari suatu produk akan ditafsirkan dengan cara yang berbeda oleh tiap tiap pelanggan. Karakteristik umumnya adalah tingkat kinerja, kemampuan, daya tarik emosional dan lainnya yang berhubungan dengan menyebabkan biaya menjadi tinggi.

Dengan demikian untuk memaksimalkan fungsi produk terhadap biaya yaitu

$$\text{Value(Nilai)} = (\text{Performance} + \text{Capability}) / \text{Cost} = \text{Function} / \text{Cost}$$

Nilai bukan hanya dimanfaatkan untuk meminimalisir biaya. Dalam beberapa kasus, nilai suatu produk dapat ditingkatkan dengan meningkatkan fungsi (kinerja atau kemampuan) dan biaya selama fungsi yang ditambahkan meningkat lebih dari biaya tambahannya. Konsep nilai fungsional bisa menjadi penting. Nilai fungsional adalah biaya terendah untuk menyediakan fungsi yang diberikan[17].

2.2.4.2. Nilai dalam Proyek

Nilai memiliki peran menjadi penentu bagi proyek dalam menaksir nilai moneter dari sesuatu. Dengan nilai manajemen proyek dapat lebih mudah dilaksanakan dan dipahami oleh anggota yang terlibat dalam proyek tersebut, bagaimana proyek ini akan selesai, dan harus seperti apa proyek tersebut berjalan[18]. Ketika berbicara mengenai implementasi proyek, nilai ini berperan dalam memberikan manfaat proyek kepada pelanggan. Ketika proyek membawa beberapa nilai, layak untuk diterapkan dan dapat dikualifikasi dalam istilah bisnis. Pelanggan mendapat manfaat organisasi, sedangkan pelaksanaa proyek menerima beberapa laba bisnis[19].

Pada PMBOK, nilai didefinisikan sebagai sebuah konsep yang unik untuk organisasi dan mencakup jumlah total semua elemen nilai yang *tangible* dan *intangible*[20]. Menentukan elemen *tangible* relatif mudah dan dapat direduksi untuk menambah finansial proyek tersebut. Namun seringkali lebih penting memahami nilai yang *intangible* dari sebuah proyek untuk akhirnya dapat mengidentifikasi opsi biaya rendah yang berpotensi menambah nilai *intangible* yang signifikan bagi organisasi serta dapat menguntungkan semua pihak yang berkaitan dalam proyek[21].

2.2.5. Proyek

Sebuah proyek bisa didefinisikan “sebagai usaha yang dijalankan untuk membuat sebuah produk atau layanan yang unik” (PMI, 2000). Sebuah proyek dilaksanakan ketika suatu pekerjaan diselesaikan dengan beberapa metode, berbeda dengan kegiatan yang dilakukan sehari-hari[22].

Proyek memiliki luaran berupa produk atau layanan yang unik. Hal tersebut berarti tidak ada satupun proyek yang memiliki kesamaan dari proyek lain. Sekalipun hasil akhir proyek terbilang sama atau bahkan sejenis dan dikerjakan oleh tim yang sama, namun selalu ada yang menjadi pembeda, seperti lokasi, desain produk, maupun keadaan atau kondisi lingkungan hasil akhir proyek tersebut[20].

Berikut ini adalah karakteristik lain yang digunakan untuk memperjelas definisi proyek :

1. Usaha yang bersifat sementara, dengan memiliki awalan(*Start*) dan akhiran (*End*).
2. Sering dipecah menjadi beberapa sub – proyek (fase)
3. Menghasilkan sebuah produk atau layanan yang unik
4. Memiliki tujuan
5. Berhubungan dengan aktivitas
6. Merupakan instrumen perubahan

Sebuah proyek memiliki komponen utama yang didalamnya memiliki manajemen (pengelolaan) proyek, istilah umum yang berkaitan dengan proyek, metode dan alat, tim kerja, perencanaan (pembuatan scope / *deliverable*, waktu, biaya, dan kualitas), kebutuhan (*identified requirement*), keinginan (*unidentified requirement*), dan stakeholder[22].

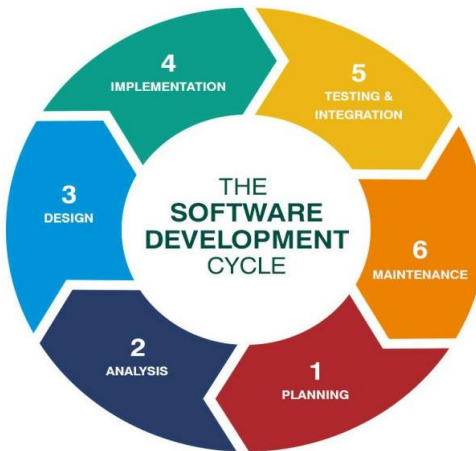
Dalam proyek dapat dibagi menjadi beberapa tipe / kategori seperti yang dijelaskan menurut Kerzer[20], yaitu :

1. *Individual projects* : Proyek ini merupakan proyek berdurasi pendek yang secara normal ditugaskan

- kepada individu yang memegang peran sebagai manager proyek dan functional manager
2. *Staff projects* : Proyek ini biasanya berada dalam lingkup departemen sebuah organisasi atau perusahaan. Proyek ini biasanya melibatkan seorang staf atau tenaga kerja dari masing-masing departemen. Namun kerja optimal jenis proyek ini adalah jika hanya melibatkan satu unit fungsi saja.
 3. *Special projects* : Proyek ini merupakan proyek khusus berdurasi pendek yang membutuhkan fungsi utama atau otoritas bagi individu atau unit tertentu.
 4. *Matrix or aggregate projects* : Proyek ini merupakan proyek yang membutuhkan input dari unit fungsional dan biasanya mengendalikan sumber daya dalam jumlah besar.

2.2.5.1. Siklus Proyek Teknologi Informasi

Software Development Life Cycle atau yang lebih dikenal dengan SDLC adalah siklus hidup pembuatan system. Metodologi ini membentuk suatu kerangka kerja perencanaan dan pengendalian pembuatan perangkat lunak[23].



Gambar 2.1 Siklus SDLC pada Proyek TI

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 sekumupulan kode.
5. *Testing*. Tahapan ini merupakan tahapan pengujian hasil kode. Pengujian dapat dilakukan dengan berbagai cara seperti *black-box testing*, *white-box testing*, dll.
6. *Maintenance*. Tahapan ini merupakan tahapan perbaikan, pembuatan atau pemeriksaan berkala yang dilakukan setelah program diimplementasikan.

2.2.6. Teknik Terkait

Di bawah ini menjelaskan mengenai teknik terkait yang digunakan selama penelitian pada setiap fase *Value Engineering*.

2.2.6.1. Benchmarking

Benchmarking adalah suatu proses mengidentifikasi “praktek terbaik” terhadap dua produk dan proses produksinya sehingga produk tersebut dikirimkan. Benchmarking memberikan wawasan yang diperlukan untuk membantu manajemen dalam memahami proses dan produknya baik dengan cara membandingkannya dengan Industri yang serupa maupun dengan Industri yang berbeda. Benchmarking dalam bahasa Indonesia sering disebut dengan tolok ukur atau patokan[24].

2.2.6.2. Perbandingan Metode Estimasi Biaya

Dibawah ini adalah hasil perbandingan metode untuk mengestimasi biaya pada pengembangan proyek aplikasi pada penelitian ini. Dalam tabel juga terdapat beberapa kekurangan dan kelebihan yang dicari sesuai referensi sebagai pertimbangan pemilihan metode estimasi yang digunakan pada penelitian ini.

Tabel 2.4 Perbandingan Metode Estimasi Biaya

| Estimation Method | Tahun | Deskripsi | Kelebihan | Kekurangan |
|-------------------|-------|---|---|---|
| COCO-MO | 1981 | COCOMO memberikan perkiraan biaya, tenaga, jadwal . Ini juga memungkinkan seorang perencana dengan mudah melakukan eksplorasi | Berikut ini adalah beberapa kelebihan metodologi COCOMO dibanding metodologi lain[6], [7], [26] a. Mampu melakukan mengestimasi biaya pembuatan aplikasi dengan keadaan kebutuhan software belum didefinisi- | a. Tidak dapat bekerja dengan baik pada framework tertentu seperti pendekatan <i>reuse-driven</i> , <i>object-oriented</i> dan sebagainya b. Tingkat |

| Estimation Method | Tahun | Deskripsi | Kelebihan | Kekurangan |
|-------------------|-------|--|---|--|
| | | <p>skenario dengan cepat menunjukkan efek penyesuaian persyaratan, sumber daya, dan staf mungkin memiliki perkiraan biaya dan jadwal (misalnya, untuk manajemen risiko atau penawaran kerja tujuan). Memiliki 10 parameter untuk melakukan estimasi biaya untuk Proyek TI [25]</p> | <p>kan b. Memiliki perhitungan optimis dan pesimistik pada data yang belum diketahui c. Implementasinya sangat sederhana dan efisien d. Memiliki teknik dokumentasi yang jelas</p> | <p>akurasi perhitungan rendah jika perhitungan dilakukan di fase awal pembuatan proyek c. Penggunaan jumlah baris kode sumber sebagai ukuran kompleksitas masih diperdebatkan. d. Keakuratan estimasi akhir dari model COCOMO sangat tergantung pada ketepatan perhitungan awal[7]</p> |
| COCO MO II | 1997 | <p>Dikembangkan untuk memasukkan pendekatan modern seperti penggunaan kembali, barang-barang komersial, memung-</p> | <p>Berikut adalah kelebihan yang ditemukan pada metode COCOMO II [26.], [27] a. Memiliki toleransi untuk perhitungan dengan berbagai ukuran aplikasi b. Dapat dijadikan sebuah <i>framework</i></p> | <p>a. Tidak dapat mengestimasi <i>effort</i> beberapa fase dalam SDLC[7]</p> |

| Estimation Method | Tahun | Deskripsi | Kelebihan | Kekurangan |
|-------------------|-------|---|--|---|
| | | <p>kinkan perencanaan untuk memperkirakan biaya, upaya, dan jadwal ketika merencanakan perangkat lunak. Terdiri dari tiga submodels, masing-masing menawarkan peningkatan keakuratan lebih jauh dalam perencanaan proyek dan proses desain[25].</p> | <p>c. Mampu bekerja pada ukuran proyek yang besar</p> | |
| Function Point | 1979 | <p>Allan Albrecht mengembangkan metode FP untuk mengukur produktivitas pembuatan perangkat lunak sistem proyek TI. Metode FP adalah alternatif untuk memperkirakan LOC[29].</p> | <p>a. estimasi dapat dengan mudah dipahami karena berbasis dari sudut pandang pengguna yang berada diluar sistem[29]</p> | <p>Berikut adalah beberapa kelemahan yang terdapat pada Function Point[30]</p> <ol style="list-style-type: none"> Membutuhkan banyak waktu Kebutuhan data harus didefinisikan terlebih dahulu Memerlukan ahli yang |

| Estimation Method | Tahun | Deskripsi | Kelebihan | Kekurangan |
|-------------------------|-------|--|--|---|
| | | | | <p>terlatih dalam melakukan perhitungan untuk mendapatkan hasil yang valid</p> <p>d. Tingkat akurasi rendah karena bergantung pada penilaian subjektif</p> |
| Earned Value Management | 1960 | <p>Earned value management (EVM) adalah teknik manajemen yang menghubungkan perencanaan sumber daya dengan jadwal dan persyaratan kinerja teknis. Semua pekerjaan direncanakan, dianggarkan, dan dijadwalkan dalam waktu – bertahap. EVM berfokus untuk mengidentifikasi</p> | <p>Kelebihan yang ada pada metode ini adalah[33]</p> <p>a. Dapat memanfaatkan kesesuaian anggaran / jadwal pada proyek</p> <p>b. Mampu menghitung kelebihan anggaran pada proyek</p> | <p>a. Tingkat keakuratan penentuan value / nilai rendah dan sulit</p> <p>b. Kesalahan melakukan perhitungan menghasilkan nilai yang lebih baik ketimbang aslinya (dan sebaliknya)[30]</p> |

| Estimation Method | Tahun | Deskripsi | Kelebihan | Kekurangan |
|-------------------|-------|--|-----------|------------|
| | | tren dan variasi kinerja dari rencana manajemen[32]. | | |

2.2.6.3. Constructive Cost Model II (COCOMO II)

Berangkat dari model estimasi yang dikembangkan Barry Boehm pada tahun 1981. COCOMO tersebut dihasilkan dari analisa banyak data dari banyak proyek perangkat lunak, kemudian pada tahun 1995 muncul versi terbaru dari model estimasi biaya COCOMO yaitu COCOMO II. Perbedaan dari kedua model estimasi biaya tersebut terlihat dari penggunaannya pada pembuatan modern perangkat lunak seperti *rapid development*, *database programming* dan lainnya. Model ini juga mendukung pembuatan model spiral. Jumlah parameter yang cukup banyak berdampak pada tingkat akurasi estimasi yang tinggi[26].

Pada COCOMO II, usaha dinyatakan dalam satuan *person-month* (PM). *Person – Month* adalah jumlah waktu perorangan yang dipakai untuk mengerjakan pembuatan perangkat lunak dalam satu bulan dimana nilainya adalah 152 jam per Person-Month[27]. Pada COCOMO II terdapat 3 model utama yang digunakan untuk estimasi sebuah perangkat lunak. Tiga model tersebut digunakan tergantung dari bagaimana siklus hidup pengembangan aplikasi dimulai pada proyek tersebut. Dibawah ini adalah beberapa model yang dimaksud :

a. Model Komposisi Aplikasi (*Application Composition Model*)

Model ini digunakan untuk estimasi usaha pada *prototyping project* yaitu adalah proyek pengembangan perangkat lunak dari penyusunan komponen yang telah ada sebelumnya (*existing component*). Dengan mengacu pada perhitungan estimasi dari *application point* yang berbobot

kemudian dibagi sesuai standar produktivitas *application point* tersebut. Hasil estimasi kemudian dipetakan sesuai tingkat kesulitan pengembangan *application point*.

Produktivitas yang dimaksud sangat bergantung dari pengalaman dan kemampuan tim pengembang serta kemampuan dari perangkat lunak (ICASE) yang digunakan selama pengembangan perangkat lunak.

Dibawah ini adalah tingkatan yang terdapat pada produktivitas yang dimaksud untuk *application point* yang ditawarkan oleh COCOMO II :

Tabel 2.5 Produktivitas *application point*

| | | | | | |
|--|----------|-----|---------|------|-----------|
| Pengalaman dan Kapabilitas Pengembangan | Very Low | Low | Nominal | High | Very High |
| Kematangan dan Kapabilitas ICASE | Very Low | Low | Nominal | High | Very High |
| PROD (NAP / month) | 4 | 7 | 13 | 25 | 50 |

Dari tingkatan tersebut kemudian akan dihitung dengan menggunakan persamaan usaha untuk sistem prototipe pada persamaan 2.1 :

$$PM = (NAP \times (1 - \%reuse/100)) / PROD \quad (2.1)$$

Keterangan :

1. PM : Estimasi usaha dinyatakan dengan satuan *person month*
2. NAP : Jumlah *application point* dari sistem yang dihasilkan
3. %reuse : Estimasi dari jumlah kode yang digunakan kemabali (*reuse code*) dalam pengembangan

4. PROD : Produktivitas *application point* yang sesuai pada tabel

b. Model Desain Awal (*Early Design Model*)

Model ini dapat digunakan pada tahap awal sebuah proyek, dengan belum terdapat desain arsitektur dari sistem yang tersedia. *Early Design Model* ini juga dapat dimanfaatkan untuk mencari beberapa pilihan kebutuhan yang nantinya dapat digunakan sebagai membandingkan kebutuhan yang berbeda pada tahap implementasi kebutuhan pengguna. Pada model ini diasumsikan bahwa kebutuhan pengguna telah disetujui serta tahap inisiasi proses desain sistem sudah dimulai. Hal ini bertujuan untuk mempercepat penghitungan estimasi biaya. Setelah itu asumsi disederhanakan.

Estimasi tahap ini memiliki persamaan seperti berikut :

$$\text{Usaha}(\text{Effort}) = A \times \text{Size}^B \times M \quad (2.2)$$

1. **A** : Koefisien dengan nilai **2.94**
2. **Size** : Dinyatakan dalam satuan KSLOC (Kilo Source Line of Codes) hasil KSLOC diperoleh dari perhitungan function point dari perangkat lunak
3. **B** : Berupa scale factors yang nilainya berbanding lurus dengan penambahan jenis proyek, dengan nilai 1.1 hingga 1,24 (tergantung dari kebaruan (novelty), fleksibilitas pengembangan, proses resolusi risiko yang digunakan, tingkat kohesi dari tim pengembang, dan tingkat kedewasaan proses dari organisasi

Dengan model ini, penyesuaian formula COCOMO II seperti berikut :

$$\mathbf{PM = 2.94 \times Size^{(1.1-1.24)} \times M} \quad (2.3)$$

Dengan nilai M berasal dari perhitungan :

$$\mathbf{M = PERS \times RCPX \times RUSE \times PDIF \times PREX \times FCIL \times SCED(2.4)}$$

1. M : Multiplier berbasis kepada 7 atribut proyek dan proses yang berpengaruh pada nilai estimasi
2. PERS : Kapabilitasn personel (personnel capability) yang bernilai dari 1 (very low) hingga 6 (very high)
3. RCPX : Reliabilitas dan kompleksitas produk (reliability and complexity product) yang memiliki skala 1 (very low) hingga 6 (very high)
4. RUSE : Kebutuhan penggunaan kembali (reuse required) yang memilki nilai dari 1 (very low) sampai 6 (very high)
5. PDIF : Pengalaman personal (personnerl experience) yang memilki nilai dari 1 (very low) sampai 6 (very high)
6. FCIL : Fasilitas pendukung (support facilities) yang memilki nilai dari 1 (very low) sampai 6 (very high)
7. SCED : Jadwal (scehdule) yang memilki nilai dari 1 (very low) sampai 6 (very high)

Perhitungan nilai multiplier diatas adalah gabungan dari *effort multiplier* untuk model ini didapat dari penjumlahan tiap *effort multiplier Post Design Model*. Berikut ini adalah nilai dan skala yang didapat dari tiap multiplier yang ada pada *Early Model*.

Tabel 2.6 Pemetaan skala nilai untuk cost driver COCOMO II

| PERS Cost Driver | | | | | | | |
|--|-----------|----------|------|---------|-------|-----------|------------|
| Hasil penjumlahan ACAP,PCAP,PCON | 3 - 4 | 5- 6 | 7- 8 | 9 | 10-11 | 12-13 | 14-15 |
| Skala level | Extra low | Very Low | Low | Nominal | High | Very high | Extra high |
| Hasil Perhitungan (<i>Effort Multiplier</i>) | 2.12 | 1.62 | 1.26 | 1.00 | 0.83 | 0.63 | 0.50 |
| RCPX Cost Driver | | | | | | | |
| Hasil penjumlahan RE-LY,DATA,CPLX, DOCU | 5-6 | 7- 8 | 9 | 10-11 | 12-13 | 14-15 | 16-18 |
| Skala level | Extra low | Very Low | Low | Nominal | High | Very high | Extra high |
| Hasil perhitungan (<i>Effort Multiplier</i>) | 0,49 | 0.6 | 0.83 | 1.00 | 1.33 | 1.91 | 2.72 |
| PDIF Cost Driver | | | | | | | |
| Hasil penjumlahan PDIF, TIME, STOR, PVOL | N/A | N/A | 8 | 9 | 10-12 | 13-15 | 16-17 |
| Skala level | Extra low | Very Low | Low | Nominal | High | Very high | Extra high |
| Hasil perhitungan (<i>Effort Multiplier</i>) | N/A | N/A | 0.87 | 1.00 | 1.29 | 1.81 | 2.61 |
| PREX Cost Driver | | | | | | | |
| Hasil penjumlahan APEX, LTEX, PLEX | 3-4 | 5-6 | 7-8 | 9 | 10-11 | 12-13 | 14-15 |
| Skala level | Extra low | Very Low | Low | Nominal | High | Very high | Extra high |
| Hasil perhitungan (<i>Effort</i>) | 1.59 | 1.33 | 1.22 | 1.00 | 0.87 | 0.74 | 0.62 |

| | | | | | | | |
|--|-----------|----------|------|---------|------|-----------|------------|
| <i>Multiplier</i>) | | | | | | | |
| FCIL Cost Driver | | | | | | | |
| Hasil penjumlahan TOOL, SITE | 2 | 3 | 4-5 | 6 | 7-8 | 9-10 | 11 |
| Skala level | Extra low | Very Low | Low | Nominal | High | Very high | Extra high |
| Hasil perhitungan (<i>Effort Multiplier</i>) | 1.43 | 1.30 | 1.10 | 1.0 | 0.87 | 0.73 | 0.62 |

Dibawah ini adalah detail skala dan nilai perhitungan yang digunakan dalam menentukan eksponen *scale factor* :

Tabel 2.7 Nilai yang ada pada *Scale Factor*

| Scale Factor | Very Low | Low | Nominal | High | Very high |
|--------------|----------|------|---------|------|-----------|
| PREC | 6.20 | 4.96 | 3.72 | 2.84 | 1.24 |
| FELX | 5.07 | 4.05 | 3.04 | 2.03 | 1.02 |
| RESL | 7.07 | 5.65 | 4.24 | 2.83 | 1.41 |
| TEAM | 5.48 | 4.38 | 3.29 | 2.19 | 1.10 |
| PMAT | 7.80 | 6.24 | 4.68 | 3.12 | 1.56 |

c. Model Penggunaan Kembali (*Reuse Model*)

Model ini digunakan pada estimasi usaha untuk proyek pengintegrasian kode baik kode yang digunakan kembali atau kode yang telah dihasilkan. Pada COCOMO II terdapat 2 jenis kode yang digunakan kembali (*reuse code*). Kode tersebut adalah :

- *Black Code* : kode ini kode yang dapat digunakan kembali tanpa harus dilakukan pemahaman atau perubahan ketika akan digunakan.
- *White Code* : jenis kode ini penggunaannya diperlukan beberapa modifikasi untuk dapat beradaptasi dengan kode baru atau komponen lain yang telah digunakan sebelumnya. Kode ini memerlukan pemahaman serta perubahan saat akan digunakan pada proses pengembangan aplikasi.

Pada model penggunaan kembali COCOMO II memiliki persamaan estimasi usaha seperti berikut :

$$PM_{AUTO} = (ASLOC \times AT / 100) / ATPROD // \text{Estimasi untuk kode yang telah dihasilkan}(\textit{generated code}) \quad (2.5)$$

1. ASLOC : Total baris kode yang digunakan kembali (*reused code*) yang di-

generate secara otomatis

2. AT : Prosentasi *reused code* yang dihasilkan secara otomatis
3. ATPROD : Produktivitas tim pengembang yang melakukan pengintegrasian kode. Dengan nilai yang didefinisikan sebesar 2400 oleh B. Boehm

d. Model Pasca Arsitektur (*Post- Architecture Model*)

Dari ketiga model yang telah disebutkan, model inilah yang memiliki penjelasan detail mengenai estimasi usaha yang diawali dari tahap desain arsitektur awal hingga proses perawatan (*maintenance*). Model ini digunakan ketika desain arsitektur awal sistem hingga struktur sub-sistem sudah diketahui. Berikut ini adalah formula yang digunakan untuk Model Pasca Arsitektur :

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i \quad (2.6)$$

Dengan nilai E diperoleh dari :

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j \quad (2.7)$$

Keterangan :

1. A : Koefisien yang bernilai 2,94
2. Size : Dihasilkan dari perkiraan ukuran modul perangkat lunak. Hal ini didapat dari bantuan perhitungan UFP (*Unadjusted Function Point*) setelah itu diubah menjadi satuan KSLOC dengan 1 KSLOC bernilai 1000 SLOC
3. E : Total penjumlahan 5 *scale factors* (SF) dari skala ekonomis dan disekonomis relatif yang terdapat

pada proyek perangkat lunak dengan ukuran berbeda

4. EM : Total dari 17 *Effort Multipliers* pasca arsitektur
5. B : Koefisien dengan nilai 0.91
6. SF : *Scale Factors* pasca arsitektur

2.2.6.4. Perhitungan Bobot

Perhitungan bobot disini adalah terkait pemberian nilai dari sebuah fungsi/ fitur dengan mempertimbangkan setiap biaya fitur dengan keseluruhan biaya total fitur yang ada pada sebuah aplikasi[5]. Prinsip dari perhitungan tersebut diambil dari formula yang ada untuk mengetahui bobot prosentase beban kerja pada sebuah proyek. Perhitungan bobot yang dimaksud seperti persamaan yang ada di bawah ini :

$$\text{Bobot} = \frac{\text{Biaya fitur}}{\text{Biaya total fitur}} \times 100\% \quad (2.8)$$

Perhitungan bobot tersebut nantinya dimanfaatkan untuk memilih objek penelitian yang kemudian menjadi dasar pembuatan FAST diagram pada fase *Function Analysis*[15]. Seperti yang telah dijelaskan pembuatan diagram FAST berguna untuk mengetahui objek permasalahan sekaligus untuk mengetahui batasan proyek berdasarkan masukan yang diterima pengguna.

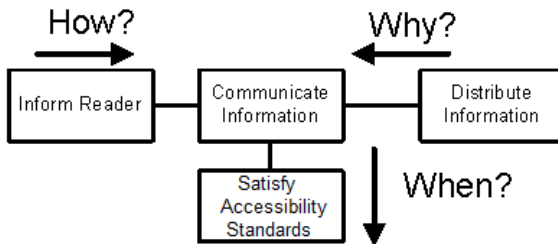
2.2.6.5. F.A.S.T Diagram

1. Pengertian FAST Diagram

Diagram Function Analysis System Technique digunakan sebagai diagram untuk menggambarkan hubungan antara fungsi-fungsi pada proyek, produk maupun proses atau layanan berdasarkan pertanyaan Bagaimana dan Mengapa.

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*?)”[15]



Gambar 2.2 Penggunaan 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.

2. Keuntungan penerapan FAST Diagram

Penggunaan F.A.S.T dapat mendukung komunikasi antar tim [15]. Hal ini akan membantu tim untuk memperoleh beberapa hal berikut ini :

1. Mengurutkan daftar acak dari fungsi yang dimiliki
2. FAST memperlihatkan hubungan spesifik dari setiap fungsi. Hal ini dapat menunjukan bahwa semua fungsi yang ada akan saling terhubung.
3. Mengidentifikasi fungsi yang hilang

4. Untuk mencari fungsi yang hilang atau terlewat dilakukan dengan cara menggunakan logika *how/why* , selain hal tersebut FAST juga dapat digunakan untuk mengklarifikasi deskripsi fungsi kata kerja/ benda dan menyediakan cara yang dapat menggantikannya untuk menjelaskan fungsi tersebut.
5. Mensimplifikasi daftar fungsi yang ditampilkan
6. Membantu mencari fungsi duplikat atau fungsi dengan deskripsi yang serupa
7. Hal ini dapat membantu untuk meminimalkan risiko terhadap munculnya fungsi yang mirip dan tidak perlu pada sebuah aplikasi/ produk.
8. Mengidentifikasi fungsi dasar, proses dan produk dari proyek
9. Membantu menyusun ruang lingkup penelitian
10. Ruang lingkup penelitian suatu objek sangat luas, hal tersebut dapat diatasi dengan menggunakan diagram FAST. Fungsi yang terletak pada sebelah kanan garis yang ada pada diagram FAST diasumsikan sebagai fungsi yang telah ada sebelumnya dan bukan untuk dipelajari
11. Memperdalam pemahaman pada masalah
12. FAST dapat menunjukkan hubungan setiap fungsi dari segi sistem keseluruhan secara objektif.
13. Sebagai media untuk menggambarkan analisa dari sebuah aplikasi

2.2.6.6. 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 pembuatan 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[16].

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

$$Nt/Np = \frac{Biaya\ awal - Biaya\ akhir}{Total\ Anggaran} \times 100\% \quad (2.9)$$

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[34] :

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

2.2.7. *Value Engineering*

2.2.7.1. *Pengertian Value Engineering*

Value Engineering, yang juga merujuk pada *Value Methodology*(VM) atau *Value Analysis*(VA) adalah proses yang sistematis, digunakan untuk mengembangkan nilai pada proyek melalui analisis fungsi[36]. Metode ini berfokus pada tujuan dilakukannya rekayasa nilai adalah jika pemasar mengharapkan produk menjadi lebih praktis dalam jangka waktu tertentu sehingga rancangan dapat bertahan dalam waktu tertentu. Dalam banyak kasus *Value Engineering* ini mengidentifikasi dan menghilangkan biaya yang tidak perlu, sehingga mampu meningkatkan nilai yang diperoleh pelanggan. Hasil akhir dari penerapan *Value Engineering* ini adalah *Value Change Proposal* (VECP) atau proposal usulan perubahan nilai[37].

Menurut studi yang dilakukan oleh SAVE International, value engineering memiliki persamaan untuk meningkatkan nilai proyek sebagai berikut :

$$\text{Value} = \text{Performance} / \text{Cost} \quad (2.10)$$

Dari persamaan diatas, didapat bahwa tujuan value engineering untuk meningkatkan kinerja dan mengurangi biaya jika mungkin, tetapi bagaimanapun jika biaya masih sama dan kinerja proyek meningkat dalam hal operasi dan kepuasan klien, maka nilai project akan meningkat juga[36].

2.2.7.2. Keuntungan *Value Engineering*

Penerapan *Value Engineering* pada proyek mampu memberikan keuntungan sebagai berikut :

1. *Value to Sponsor and Stakeholder – Project manager* dan tim dapat mendemokan nilai yang akan diperoleh selama masa pengerjaan proyek
2. *Enhanced Professionalism – Project manager* dan tim dapat mengasah kemampuan, terkhusus pada analisis, penentuan dan pembuktian nilai untuk meningkatkan *image* project manager dan kemampuan dalam memimpin.
3. *Predictable result* – dengan menggunakan *value engineering* hasil dari sebuah proyek nantinya dapat diperkirakan pada tahap perencanaan proyek.
4. *Repeatable Success – Project manager* dapat mengulang keberhasilan proyek seperti yang pernah dilakukan sebelumnya, sehingga meningkatkan *image* perusahaan atau organisasi sehingga semakin terpercaya dimata publik.
5. *Enhanced business value – Project manager* dan tim tidak hanya memberikan apa yang

diekspektasikan oleh organisasi namun juga memberikan nilai bisnis yang memberikan keuntungan yang terbaik

6. *Clear project vision and focus* – dengan level pre-project teknik dan proses, *project manager* dan stakeholder dapat menjalin kerja sama dengan satu tujuan yang sama.
7. *Leadership* – Dapat membantu memperkuat jiwa kepemimpinan dan kerjasama tim
8. *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
9. Penghematan dapat diterapkan ke semua unit produksi
10. Pengurangan biaya yang tinggi, biaya produksi berikutnya, dan biaya konsekuensi yang terkait dengan operasi dan dukungan dapat direalisasikan.

2.2.7.3. Proses Value Engineering

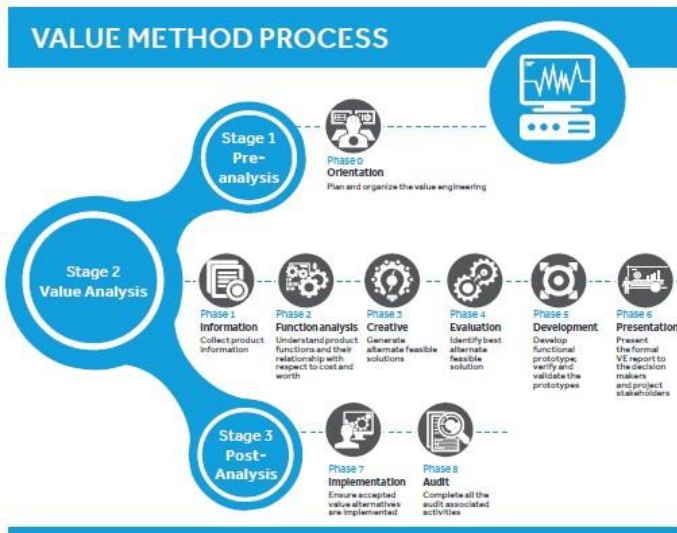
Menurut hasil penelitian SAVE Internasional ditahun 2007, Value Engineering terdiri dari 3 level/ stage dan 9 fase proses yang diterapkan pada perubahan proyek[38].

a. Pre – Analysis

Level ini adalah level pertama yang ada pada *value engineering*. Level ini didalamnya dilakukan kegiatan perencanaan analisis *value* yang digunakan untuk level Selanjutnya.

0. Orientation Phase

Pada fase ini, persiapan dan pemahaman tentang *value engineering* dilakukan. Luaran dari fase ini adalah dokumentasi tentang kondisi terkini mengenai perusahaan pemegang proyek.



Gambar 2.3 Alur Proses Value Engineering

b. Value Analysis

Kegiatan di level ini adalah menjalankan perencanaan yang diatur pada level Pre – Analysis. Level ini terdiri dari 6 fase proses yaitu :

1. Information Phase

Aktivitas yang dilakukan pada fase ini adalah pengumpulan semua informasi dari proyek yang dibutuhkan. Informasi yang dimaksud adalah penentuan tujuan serta batasan (*timeline* dan anggaran) dari proyek.

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

2. *Function Analysis Phase*

Aktivitas yang dilakukan dalam fase ini adalah menganalisa kebutuhan klien baik fungsi dan yang hal lainnya (nilai dan biaya) terkait dengan tujuan proyek. Perbandingan nilai / biaya mengidentifikasi efisiensi fungsional dari proyek. Dengan tinggnya biaya / nilai menunjukkan potensi untuk melakukan peningkatan nilai[39]. Teknik yang dapat digunakan pada fase ini adalah FAST (*Function Analysis System Technique*).

3. *Creativity Phase*

Aktivitas yang dilakukan pada fase ini adalah mencari beberapa solusi yang dapat diterapkan untuk meningkatkan nilai proyek. Data masukan dari fase ini sangat bergantung pada hasil yang berasal dari fase sebelumnya, yaitu berdasarkan pendefinisian tujuan proyek (*Information Phase*) dan fungsi, biaya, nilai, sistem produk, fasilitas lain (*Function Analysis Phase*).

Fase ini akan menghasilkan skenario yang tidak memiliki ketentuan jumlah untuk dikembangkan pada fase berikutnya. Skenario ini akan menjadi isian yang dimuat pada dokumen VECP

4. *Evaluation Phase*

Fase evaluasi atau *Evaluation Phase* adalah fase yang dilakukan untuk mengevaluasi skenario yang telah dibuat pada fase sebelumnya. Skenario ini nantinya akan dihitung dengan menggunakan metode perhitungan indeks nilai. Hasil perhitungan ini menunjukkan apakah skenario solusi tersebut layak dilakukan atau tidak. 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.

Luaran dari fase ini adalah solusi (skenario) disertai kelebihan dan kelemahan yang dimuat pada dokumen VECP.

5. *Development Phase*

Analisa kembali pada fase ini dilakukan untuk solusi yang telah dipilih pada fase sebelumnya. Hal ini bertujuan untuk meningkatkan kualitas solusi yang dipilih.

Keputusan penerapan solusi terpilih ini berada ditangan *project manager*. Jika solusi tersebut disetujui untuk diterapkan, *project manager* akan melakukan perencanaan implementasi proyek serta membagi *jobdesk* dan tanggungjawab untuk setiap tim proyek.

Luaran fase ini adalah perencanaan penerapan proyek, prosedur, *milestone*, *timeline* yang telah dibuat oleh *project manager* untuk menghindari ketidaksesuaian implementasi proyek.

6. *Presentation Phase*

Aktivitas presentasi kepada *stakeholder* untuk memberikan pemahaman terhadap alternatif solusi yang dipilih serta kebutuhan yang berubah atau tidak setelah solusi tersebut diterapkan nantinya.

Dokumen VECP ini diberikan pada *stakeholder* untuk kemudian disetujui dan dijadikan acuan pada penerapan usulan *value engineering*.

c. **Post Analysis**

Level/ tahapan ini aktivitas penerapan rencana proyek telah berlangsung.

1. *Implemetation Phase*

Adalah fase penerapan proyek berdasarkan acuan dokumentasi VECP yang telah disetujui semua pihak yang terlibat dalam proyek terutama *stakeholder*.

2. *Audit Phase*

Fase audit dilakukan untuk memastikan apakah semua aktivitas proyek telah berjalan sesuai dengan dokumen VECP, baik mencakup pemenuhan kebutuhan, kesesuaian *timeline* dan lain lain.

2.2.8. Value Engineering Change Proposal

2.2.8.1. Pengertian Value Engineering Change Proposal

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[38].

2.2.8.2. Konten Value Engineering Change Proposal

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 [40]

1. Apakah VECP yang anda susun telah menggambarkan perbedaan antara skenario awal dengan modifikasi yang diusulkan, yang keuntungan dan kerugian

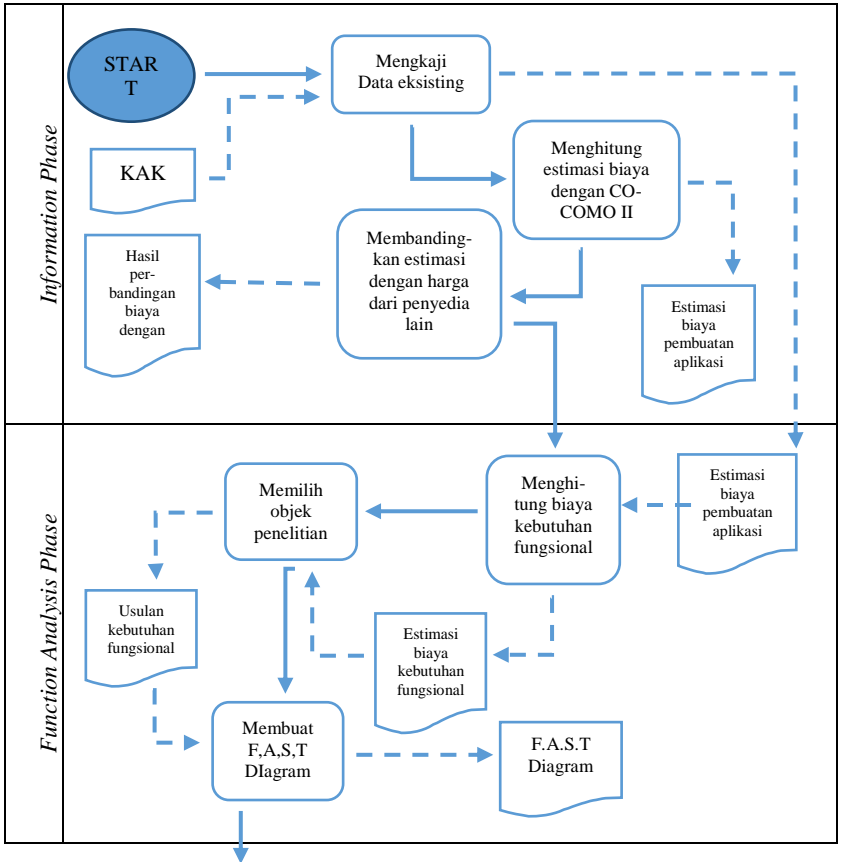
2. relatifnya juga telah dipertimbangkan dan menunjukkan adanya perubahan ? Hal ini harus dijelaskan secara rinci
3. Sudahkan VECP tersebut berisi spesifikasi rinci tentang fungsi yang akan diubah?
4. Apakah VECP tersebut telah berisi perkiraan penghematan biaya sesuai rekomendasi anda dibandingkan dengan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

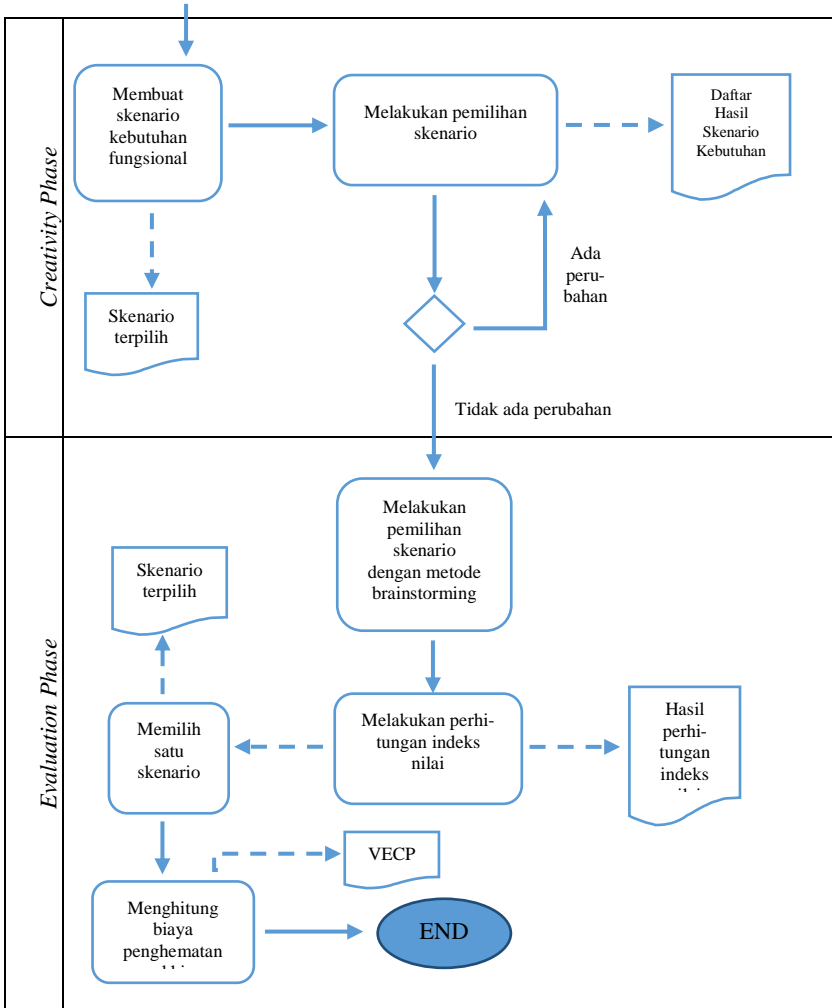
Jika anda telah melakukan beberapa daftar diatas, maka dapat dipastikan bahwa VECP anda sudah lengkap dan dapat memberikan informasi yang jelas kepada pihak penerima VECP.

Halaman ini sengaja dikosongkan

BAB III METODOLOGI

3.1. Diagram Metodologi





Gambar 3.1 Digaram Pengerjaan Tugas Akhir

3.2. Uraian Metodologi

3.2.1. Value – Analysis

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

a. Information Phase

Pada tahap *Information Phase* **bertujuan untuk mengetahui perbandingan biaya** yang ditawarkan oleh pihak ketiga (termasuk dalam kategori pembuatan aplikasi dengan biaya mahal atau tidak).

Seperti yang terlihat pada Error! Reference source not found., a ktivitas pertama yang dilakukan adalah mengkaji data ek-sisting proyek. Data ini didapat dari kegiatan interview terhadap pihak ketiga yang menangani perencanaan proyek, mengenai data apa saja yang mereka punya terkait dari proyek tersebut. Karena proyek ini masih dalam tahapan perencanaan, maka data yang didapat adalah KAK (Kerangka Acuan Kerja) yang lebih berfokus membahas kepada anggaran biaya keseluruhan proyek melibatkan gaji /payrate setiap tim proyek beserta aktivitas yang akan dilakukan namun masih bersifat general. Data tersebut kemudian dipelajari dan diperiksa apakah telah sesuai dengan ketentuan data yang diperlukan pada tahap *Information Phase* seperti yang didefinisikan di bawah ini :

1. *Timeline* pembuatan aplikasi
2. *Resource* yang dibutuhkan selama proyek pembuatan aplikasi berjalan.
3. Total Biaya pembangunan aplikasi
4. Harga aplikasi tiap fitur
5. Perbandingan aplikasi dengan aplikasi lain
6. Pembagian tugas untuk setiap SDM selama pengerjaan aplikasi

Ketika data yang dibutuhkan pada proses ini tidak sesuai dengan data dan informasi yang dijelaskan sebelumnya, peneliti melengkapinya dengan menggunakan beberapa teknik yang bisa digunakan untuk mendukung proses (fase) yang ada pada *Value Engineering*. Berikut adalah teknik yang digunakan untuk melengkapi data yang diperlukan :

1. COCOMO II

Jika kebutuhan data mengenai harga tiap fitur pada aplikasi belum diketahui, berbagai macam metode estimasi biaya bisa digunakan, namun sesuai dengan penelitian ini,

metode yang digunakan adalah metode COCOMO II. Teknik ini digunakan untuk membuat perhitungan estimasi biaya terhadap keseluruhan biaya aplikasi, dan harga aplikasi tiap fitur

2. *Benchmarking*

Teknik ini dibutuhkan untuk mengetahui apakah biaya yang ditawarkan oleh pihak ketiga atau perhitungan estimasi biaya yang dilakukan dengan perhitungan menggunakan metode COCOMO II termasuk dalam kategori mahal atau tidak pada proyek pembuatan aplikasi. Aktivitas yang dilakukan adalah membandingkan harga aplikasi serupa yang dikembangkan oleh pihak lain.

Berikut ini adalah pemetaan proses, *input*, *output* pada *Information phase* :

Tabel 3.1 Input, output, dan proses yang ada pada information phase

| Input | Proses / Aktivitas | Output |
|---|--|---|
| <ul style="list-style-type: none"> • BAB II : Dasar Teori • BAB III : Metodologi Penelitian • BAB I : Latar belakang permasalahan dan tujuan penelitian • Data kondisi Ek- | <ul style="list-style-type: none"> • Mengkaji data kondisi eksisting • Melakukan perhitungan estimasi biaya pembuatan dengan metode COCOMO II • Melakukan perbandingan estimasi | <ul style="list-style-type: none"> ▪ BAB I : Ketentuan objek yang akan diteliti, sasaran yang dicapai, ruang lingkup (batasan masalah) pada penelitian ▪ Usulan kebutuhan <i>fungsional</i> pihak ketiga ▪ Estimasi biaya pembangunan |

| Input | Proses / Akti- vitas | Output |
|--|-------------------------------------|---|
| sisting <ul style="list-style-type: none"> • KAK | biaya dengan penyedia jasa | aplikasi <ul style="list-style-type: none"> • Perbandingan estimasi biaya dengan penye- dia jasa / pihak ketiga |

b. *Function Analysis Phase*

Aktivitas pada tahap ini diperlukan untuk mengetahui objek atau kebutuhan fungsional yang menjadi objek kajian dan digunakan pada tahap Selanjutnya. Hal yang dilakukan peneliti adalah melakukan perhitungan biaya untuk masing masing kebutuhan fungsional menggunakan teknik estimasi biaya COCOMO II (perhitungan tersebut telah dilakukan pada fase sebelumnya). Setelah mendapatkan hasil perhitungan harga, proses dilanjutkan dengan pembobotan untuk masing – masing kebutuhan fungsional hal ini dilakukan untuk melihat tingkat fungsi yang membutuhkan biaya yang besar.

Setelah mengetahui peringkat melalui proses pembobotan, tahap terakhir adalah membuat F.A.S.T Diagram untuk objek yang dipilih. Pembuatan diagram ini untuk mengetahui keterkaitan komponen sistem antara satu sama lain, dan membuat fungsi tersebut mampu untuk berjalan.

Tabel 3.2 Input, ouput dan proses pada Function Analysis Phase

| Input | Proses / Aktivitas | Output |
|---|--|---|
| <ul style="list-style-type: none"> ▪ BAB I : Ke- tentuan objek yang akan diteliti, sasaran yang dicapai, ruang lingkup (batasan masa- | <ul style="list-style-type: none"> ▪ Melakukan perhitungan biaya masing- masing kebu- tuhan fungsional ▪ Melakukan | <ul style="list-style-type: none"> ▪ Estimasi biaya tiap kebutuhan fungsional ▪ Perhitungan bobot tiap fungsional |

| Input | Proses / Aktivitas | Output |
|--|--|--|
| <p>lah) pada penelitian</p> <ul style="list-style-type: none"> ▪ Usulan kebutuhan fungsional pihak ketiga ▪ Estimasi biaya pembangunan aplikasi • Perbandingan estimasi biaya dengan penyedia jasa / pihak ketiga | <p>pemilihan objek penelitian</p> <ul style="list-style-type: none"> • Pembangunan FAST diagram | <ul style="list-style-type: none"> ▪ Penentuan objek penelitian • Diagram FAST pada objek penelitian |

c. *Creativity Phase*

Aktivitas yang dilakukan pada fase ini adalah membuat skenario dari objek yang terpilih pada tahap sebelumnya (Function Analysis Phase). Pembuatan skenario didasarkan pada ketentuan bahwa, skenario mampu melakukan penghematan biaya atau mengurangi harga pembuatan aplikasi tanpa merubah fungsi serta keterkaitan komponen sistem yang membuat fungsi tersebut berjalan. Pembuatan skenario tidak mempunyai batasan jumlah. Pembuatan skenario dilakukan dengan perhitungan COCOMO II untuk mengetahui usaha (*effort*) serta waktu pada masing masing fitur.

Aktivitas berikutnya adalah melakukan validasi pada pihak ketiga dengan teknik brainstorming untuk mengetahui apakah skenario yang diusulkan peneliti disetujui oleh pihak ketiga. Tahapan ini juga memiliki manfaat lain untuk menampung usulan dari pihak ketiga terkait dengan skenario yang telah diusulkan.

Jika terdapat perubahan baik penambahan usulan dari pihak ketiga, tahapan ini akan diulang dari aktivitas pembuatan ske-

nario, kemudian dilanjutkan dengan teknik brainstorming dengan pihak ketiga hingga terjadi kesepakatan antara peneliti dan pihak ketiga pada skenario yang diusulkan.

Luaran proses ini adalah skenario yang berhasil dan disetujui oleh pihak ketiga. Ketentuan skenario yang disetujui tidak memiliki keterbatasan jumlah. Dengan demikian skenario yang dihasilkan lebih dari 1.

Tabel 3.3 Input, output dan proses Creativity Phase

| Input | Proses / Aktivitas | Output |
|---|--|---|
| <ul style="list-style-type: none"> ▪ Estimasi biaya tiap kebutuhan fungsional ▪ Perhitungan bobot tiap fungsional ▪ Penentuan objek penelitian • Diagram FAST pada objek penelitian | <ul style="list-style-type: none"> ▪ Membuat skenario fungsi pada aplikasi • Melakukan validasi skenario dengan metode brainstorming pada pihak ketiga sebagai rujukan/ dasar penentuan hasil skenario | <ul style="list-style-type: none"> ▪ Daftar hasil skenario yang dibuat • Catatan hasil brainstorming terkait skenario |

d. *Evaluation*

Fase ini adalah fase terakhir dari pengerjaan penelitian. *Evaluation Phase* pada metodologi ini memiliki rangkaian aktivitas berupa :

1. Melakukan perhitungan indeks nilai
Aktivitas ini dilakukan dengan menampilkan perbandingan total harga yang dihasilkan pada pembuatan skenario dan sebelum pembuatan skenario dilakukan. Kemudian aktivitas Selanjutnya adalah melakukan perhitungan indeks nilai untuk mengetahui apakah hasil setiap skenario tersebut dapat dipertimbangkan untuk diterapkan pada fase berikutnya.
2. Memilih satu skenario / solusi
Dari hasil perhitungan indeks nilai, akan didapatkan status pertimbangan penerapan *Value Engineering* untuk setiap skenario. Aktivitas Selanjutnya adalah memilih salah satu skenario/ solusi ini dengan menggunakan grafik hubungan waktu dan biaya. Skenario yang akan dipilih berasal dari hasil pencarian indeks nilai yang lolos kriteria layak dipertimbangkan pada perhitungan ini. Skenario yang diambil nantinya adalah skenario yang memiliki hasil hubungan waktu dan biaya yang paling optimal (biaya rendah dan waktu pengerjaan cepat).
3. Menghitung penghematan biaya akhir
Setelah skenario dipilih dari perhitungan hubungan waktu dan biaya, aktivitas Selanjutnya adalah menyusun proposal perubahan VECP yang didalamnya berisi hal hal seperti :
 - Perbandingan total harga skenario terpilih dibanding dari pihak ketiga
 - Informasi kelebihan kekurangan yang dimiliki oleh perhitungan hasil *value engineering* dengan perhitungan dari harga yang ditawarkan oleh pihak ketiga
 - Menyertakan detail kebutuhan fungsional dan harga tiap fungsi yang diperlukan dari hasil perhitungan sebelum dan sesudah diterapkannya *value engineering*

Luaran dari fase ini didokumentasikan ke dalam Value Engineering Change Proposals (VECP) atau proposal usulan perubahan Value Engineering sebagai pedoman untuk pihak yang terlibat dalam proyek untuk kemudian dapat diimplementasikan sesuai dengan biaya dan timeline waktu pengerjaan proyek. Peneliti dapat memberikan masukan pada pihak terlibat dan kemudian dilakukan pendokumentasian hasil perencanaan kedalam buku Tugas Akhir.

Tabel 3.4 Input, output dan proses yang ada pada Evaluation Phase

| Input | Proses / Aktivitas | Output |
|---|---|---|
| <ul style="list-style-type: none"> ▪ Daftar hasil skenario yang dibuat • Catatan hasil brainstorming terkait skenario | <ul style="list-style-type: none"> ▪ Melakukan perhitungan indeks nilai ▪ Memilih satu skenario penghematan • Menghitung penghematan biaya akhir | <ul style="list-style-type: none"> ▪ Dokumentasi Pencarian Indeks Nilai • Dokumen VECP (Lampiran) |

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. Tahap Perencanaan Proyek

Bagian ini menjelaskan kegiatan yang dilakukan dalam tahap perencanaan proyek untuk studi kasus penelitian. Tahap perencanaan proyek pada penelitian ini bertujuan untuk mengetahui penghematan proyek pengembangan aplikasi *E-Performance*.

Tahap perencanaan yang dimaksud meliputi kegiatan/ fase mengikuti fase yang ada pada tahapan *Value Engineering* dengan batasan perencanaan dilakukan hingga tahap *Value Analysis* – fase evaluasi. Berikut adalah luaran perencanaan proyek yang dihasilkan pada fase *Value Engineering* yang dilakukan pada penelitian ini :

Tabel 4.1 Pemetaan kegiatan perencanaan proyek

| Fase kegiatan | Aktivitas | Luaran perencanaan |
|--------------------------|---|---|
| <i>Information Phase</i> | Estimasi biaya untuk menentukann biaya yang dibutuhkan dalam proyek | <ul style="list-style-type: none"> • Estimasi biaya proyek • Estimasi <i>resource</i> (sumber daya manusia) • Penggalian kebutuhan fitur aplikasi dari KAK dan dokumentasi proyek serupa dari tim pengembang |

| | | |
|--------------------------------|--|--|
| <i>Function Analysis Phase</i> | Estimasi biaya untuk mengetahui biaya yang dibutuhkan dalam mengembangkan setiap fitur yang ada dalam aplikasi | <ul style="list-style-type: none"> • Estimasi biaya fitur • Estimasi <i>resource</i> (sumber daya manusia) |
| <i>Creativity Phase</i> | Melakukan penghematan biaya / waktu dengan membuat alternatif skenario | <ul style="list-style-type: none"> • Alternatif solusi yang menunjukkan penghematan biaya/ waktu untuk pengembangan fitur pada aplikasi |
| <i>Evaluation Phase</i> | Melakukan pemilihan alternatif skenario sebagai rekomendasi untuk didokumentasikan pada VECP | <ul style="list-style-type: none"> • Rekomendasi yang dihasilkan dari alternatif solusi yang dipilih dalam dokumen VECP (<i>Value Change Proposals</i>) |

4.2. Penjelasan Aplikasi

Aplikasi e-Performance adalah sistem informasi manajemen kinerja berbasis website dalam rangka penilaian kinerja pegawai yang lebih objektif, terukur, akuntabel, partisipatif dan transparan, sehingga terwujud manajemen pegawai berdasarkan kinerja dan sistem pada Pemerintah daerah setempat. Pengerjaan proyek tersebut direncanakan akan diselesaikan selama 6 bulan oleh tim pengembang dengan fungsi utama dari aplikasi ini adalah sebagai tempat *monitoring* dan evaluasi kinerja pegawai dan dapat diakses sesuai dengan batasan tiap jabatan pegawai negeri. Terdapat beberapa fungsi lain yang dibutuhkan oleh *client* seperti yang ada pada Tabel 4.2

Tabel 4.2 Kebutuhan fungsional aplikasi

| No | Kebutuhan Fungsional |
|-----|--|
| 1 | Aplikasi dapat menampilkan informasi <i>log</i> pengguna |
| 2 | Aplikasi dapat menampilkan informasi perangkat daerah |
| 3 | Aplikasi dapat meng- <i>input</i> data pegawai sesuai tingkat jabatan |
| 4 | Aplikasi dapat menampilkan daftar aktivitas monitoring capaian kerja pegawai |
| 5 | Aplikasi dapat melakukan <i>update</i> data pegawai |
| 6 | Aplikasi dapat memberikan evaluasi berupa kuis dalam bentuk form soal untuk evaluasi pegawai pada kurun waktu tertentu |
| 7 | Aplikasi dapat menampilkan indikator yang berhasil dicapai untuk masing masing pegawai |
| 8. | Aplikasi yang dapat menampilkan informasi tugas yang dilakukan setiap pegawai |
| 9. | Aplikasi yang dapat menginputkan aktivitas pegawai |
| 10. | Aplikasi yang dapat menampilkan rapor NKI |
| 11. | Aplikasi yang dapat menampilkan capaian kegiatan |
| 12. | Aplikasi yang dapat menampilkan data sasaran program dan kegiatan PD (Perangkat Daerah) |
| 13. | Aplikasi yang dapat menampilkan status aktivitas perangkat daerah |
| 14. | Aplikasi yang dapat menampilkan nilai akhir tiap pegawai daerah |
| 15. | Aplikasi dapat memperbanyak aktivitas/ tugas yang sama untuk beberapa SKPD bersangkutan |

Untuk beberapa fungsi pada aplikasi diatur dan hanya dapat diakses sesuai jabatan yang dimiliki oleh customer yaitu pemerintah daerah setempat

Berikut ini adalah jenis user yang menggunakan aplikasi *e-Performance* :

1. Admin PD
2. Pegawai Daerah (terdapat beberapa tingkat, mengikuti jabatan yang dimiliki oleh pemerintah daerah setempat)
3. Super Admin

Dari penjelasan kebutuhan fungsi pada aplikasi seperti yang ada pada Tabel 4.2

Tim pembangun aplikasi kemudian mengusulkan beberapa fitur untuk memenuhi kebutuhan *client* seperti :

1. Fitur yang dapat menampilkan halaman berisi informasi dan detail pengguna (setiap tanggal, *username*, tipe pengguna, dan deskripsi)
2. Fitur yang menampilkan daftar perangkat daerah beserta detail tiap PD (kode, nama dan alamat)
3. Fitur yang dapat membantu pengguna untuk melakukan *input* dan monitor data pegawai. Fitur ini dilengkapi beberapa menu yang digunakan untuk melihat informasi jabatan staf, pegawai lapangan, dan lain lain
4. Fitur yang dapat membantu memonitoring capaian tiap aktivitas pegawai dengan menampilkan daftar aktivitas yang dikategorikan sesuai jabatan staf.
5. Fitur yang dapat digunakan untuk melakukan *checklist* data SKPD berdasarkan beberapa data seperti untuk *update* data pegawai, penilaian kinerja, indikator kinerja dll.
6. Fitur yang dapat digunakan untuk membantu pengguna dalam melakukan monitoring pekerja dengan menyediakan soal-soal yang berisi tentang perilaku kerja, dan monitoring nilai perilaku kerja.

7. Fitur yang dapat digunakan untuk menampilkan nilai indikator tujuan sasaran dari setiap pegawai
8. Fitur yang dapat memberi akses membuka dan menutup tes perilaku, melakukan *input* aktivitas, memverifikasi aktivitas pegawai, melakukan sinkronisasi capaian sub kegiatan dan capaian pentahapan. Serta membantu untuk monitoring keterlibatan tiap aktivitas pegawai. Dan yang terakhir fungsi ini juga dilengkapi dengan dapat menampilkan rapor dari nilai akhir dari setiap karyawan perangkat daerah

4.3. Value Analysis

4.3.1. Information Phase

Tahapan ini berisi persiapan pemahaman KAK yaitu dilakukan pengecekan apakah KAK tersebut telah terkumpul sesuai dengan kebutuhan peneliti untuk pengerjaan tugas akhir dan bahan yang dibutuhkan pada proses pengerjaan *value engineering*, atau belum. Berikut ini adalah data yang harus ada pada KAK agar dapat dilanjutkan ke tahap berikutnya :

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

Apabila dari ke 6 daftar tersebut ada salah satu yang tidak ada seperti harga tiap fitur maka peneliti akan melakukan perhitungan biaya estimasi untuk harga fitur menggunakan metode estimasi biaya yang telah dipilih yaitu COCOMO II

a. **Perhitungan Estimasi Biaya dengan COCOMO II**

Pada fase ini peneliti mulai melakukan pemetaan terhadap data eksisting yang didapatkan untuk mengetahui jumlah total pengerjaan proyek, estimasi biaya yang dikeluarkan, beserta SDM yang dibutuhkan. Untuk menghitung estimasi biaya yang dikeluarkan selama pelaksanaan proyek dihitung dengan menggunakan metode COCOMO II. Penjelasan langkah – langkah menggunakan COCOMO II adalah sebagai berikut :

Langkah 1. Menentukann Model COCOMO II

Seperti yang telah dijelaskan pada Dasar Teori mengenai macam macam model yang terdapat pada COCOMO II, peneliti membuat checklist untuk menentukann model COCOMO II mana yang akan digunakan dalam perhitungan estimasi usaha dan biaya untuk tahap perencanaan proyek pembuatan aplikasi *e-performance* ini.

Pemilihan Model didasarkan dengan Kriteria berikut :

- Tahapan pada siklus pengembangan aplikasi. Sesuai dengan dokumentasi yang diperoleh dari pihak ketiga beserta topik yang dipilih peneliti, untuk saat ini pembuatan aplikasi masih berada pada tahap perencanaan dengan dokumen yang tersedia hanya berupa KAK, dan kebutuhan fungsional yang didapat dari hasil observasi dari pihak client/ user
- Pembangunan aplikasi tidak menggunakan kode yang telah diterapkan pada aplikasi serupa yang ditangani oleh tim pengembang
- Ketersediaan dokumen beserta anggota tim pengembang terdiri dari anggota tim baru

Sesuai dengan kriteria yang peneliti acu dari literatur model COCOMO II, peneliti memilih *Early Design Model* yang digunakan untuk menghitung estimasi usaha dan biaya untuk pembangunan aplikasi *e-performance* ini.

Dengan demikian peneliti, mengacu pada formula persamaan COCOMO II pada *Early Design Model* seperti yang tertera pada persamaan dibawah ini :

$$PM (Effort) = 2.94 \times Size^E \times M \quad (4.1)$$

Dimana :

$$E = B + (0,01 * \sum_{j=1}^5 SF_j) \quad (4.2)$$

dan

$$M = \prod_{i=1}^7 EM \quad (4.3)$$

Keterangan :

- PM (Ef- : Hasil estimasi usaha yang memiliki satuan
fort) *person-month*
- Size : Ukuran baris kode yang memiliki satuan
KSLOC (*Kilo Source Line of Code*).
- E : Eksponen yang didapat dari penjumlahan 5
scale factors
- M : *Effort Multiplier* adalah variabel yang menen-
tukkan/ mengatur usaha, person month untuk
menunjukkan ukuran produk perangkat lunak
yang sedang dalam pengembangan. Masing
masing model pada COCOMO II memiliki
jumlah Effort Multiplier yang berbeda. Pada
Early Design Model, Effort Multiplier yang
digunakan berjumlah 7 dari total 32 *Effort*
Multiplier yang ada pada COCOMO II
- SF : *Scale Factor*

Langkah 2. Menghitung variable *Size* dengan menggunakan *Unjusted Function Point*

Sesuai dengan formula perhitungan Usaha pada COCOMO II, *Size* dengan satuan KSLOC dari aplikasi dapat dihitung langsung dengan menggunakan UFP, atau bisa dengan menggunakan bantuan software untuk *generate* total SLOC dari aplikasi. Karean peneliti tidak mendapatkan *source code* dari tim pengembang, peneliti menggunakan UFP untuk menghitung SLOC pada aplikasi.

Sesuai dengan literatur yang peneliti dapatkan, langkah pertama dalam menentukan ukuran UFP adalah memetakan *input*/ masukan dari aplikasi berdasarkan kategori kategori yang ada pada UFP seperti[41] :

1. *Internal Logical File (ILF)* : fungsi *input* yang berkaitan dengan penyimpanan data. Contoh elemen : file yang selalu diperbaharui tiap saat.
2. *External Interface File (EIF)*: Fungsi *input* yang berkaitan dengan komunikasi data dengan perangkat atau mesin lain. Contoh elemen : *Real Time Embedded*, penarikan data dari aplikasi lain
3. *External Input (EI)*: Fungsi yang berkaitan dengan *interfae*/tampilan antar muka yang digunakan pengguna untuk memasukkan data pada aplikasi. Contoh elemen : masukan *input* dari aplikasi yang dilakukan oleh pengguna, (*login, edit data*), *Error Message, Action Button*
4. *External Output (EO)*: Fungsi yang berkaitan dengan *ouput* yang dihasilkan aplikasi untuk pengguna. Contoh elemen : *notification message*, tabel, grafis, *file download*
5. *External Inquiry (EQ)*: Fungsi yang berkaitan dengan *query* terhadap data yang disimpan. Contoh elemen : *Drop down list*

Dari pemetaan kategori tersebut, peneliti menyesuaikan aktivitas aplkasi dengan kriteria yang ada untuk mengetahui kategori masing - masing aktivitas. Untuk mendokumentasikannya peneliti merancang tabel hasil pemetaan kategori *input* seperti pada template Tabel 4.3 dibawah ini :

Tabel 4.3 Template tabel hasil pemetaan kategori input

| Fitur/ Fungsi | Tipe Masukan | | | | |
|---------------|--------------|-----|----|----|----|
| | ILF | EIF | EI | EO | EQ |

Keterangan :

- Fitur/ Fungsi : Diisi dengan fitur/ fungsi yang terdapat pada aplikasi
- Tipe Masukan : Diisi dengan jumlah tipe masukan. Contoh jika pada suatu fungsi memiliki 2 EO, maka isi kolom EO dengan 2, begitu seterusnya

Hasil tersebut kemudian akan dipetakan kembali untuk mengetahui jenis kompleksitas masukan. Untuk memudahkan mengetahui jumlah DET/ RET/ FTR yang dimiliki sebuah fungsi, peneliti membuat EF Diagram yang berisi tabel/ database yang digunakan aplikasi. Pemetaan tersebut didokumentasikan seperti yang terdapat pada template **Tabel 4.4**

Tabel 4.4 Pemetaan kompleksitas masukan dalam perhitungan UFP

| Fitur/ Fungsi | Tipe Masukan | | | | | | | | | |
|------------------|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | ILF | | EIF | | EI | | EO | | EQ | |
| | RET | DET | RET | DET | FTR | DET | FTR | DET | FTR | DET |
| | | | | | | | | | | |
| Total | | | | | | | | | | |

Keterangan :

- Fitur/ Fungsi : Diisi dengan fitur/ fungsi yang terdapat pada aplikasi
- Tipe Masukan : Diisi dengan jumlah tipe masukan. Contoh jika pada suatu fungsi memiliki 2 EO, dengan FTR (dalam hal ini bisa dari atribut yang unik dari database) terhubung dengan 2 database/ tabel setiap tabel memiliki 3 atribut maka dapat ditulis 2 pada kolom EO, dan 6 pada kolom FTR

Kemudian setiap kategori input, dipetakan berdasarkan tingkat kompleksitas (*low, average, high*) dengan tabel spesifikasi dari tiap tingkatan kompleksitas *input*, seperti **Tabel 4.5** :

Penentuan kompleksitas dari sebuah input dilakukan dengan menganalisa DET, RET, dan FTR. Berikut ini adalah pengertian dari istilah yang dimaksud[42] :

1. DET (*Data Element Type*) adalah merupakan jenis elemen data yang dikenal oleh pengguna sebagai *field* yang unik dan tidak berulang (*non repeatable data field*).
2. FTR (*File Type Reference*) jenis file yang dibaca oleh masukan bertipe EI, EO, dan EQ
3. RET (*Record Element Type*) adalah subgroup dari *Data Element Type* yang dikenal pengguna dalam tipe masukan ILF dan EIF. Contoh dari jenis record ini adalah Tabel Pegawai yang memiliki sub tabel master_pegawai maka record tersebut dihitung 2 RET

Tabel 4.5 Kompleksitas kategori input data element

| <i>Internal Logical Files dan External Interface Files</i> | | | |
|---|------------|--------------------|-------------|
| | | Elemen Data | |
| <i>Record Element Type</i> | 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> |
| <i>External Output dan External Inquiry</i> | | | |
| | | Elemen Data | |
| File Type Refrence | 1-5 | 6-19 | 20+ |
| 0 atau 1 | <i>Low</i> | <i>Low</i> | <i>Avg</i> |

| <i>Internal Logical Files dan External Interface Files</i> | | | |
|--|------------|--------------------|-------------|
| 2-3 | <i>Low</i> | <i>Avg</i> | <i>High</i> |
| 4+ | <i>Avg</i> | <i>High</i> | <i>High</i> |
| <i>External Input</i> | | | |
| | | Elemen Data | |
| File Type Reference | 1-4 | 5-15 | 16+ |
| 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> |

Hasil pemetaan kategori kompleksitas *input* kemudian dijumlahkan hingga mendapat hasil total poin dengan mengalikan jumlah *input* dengan bobot yang dimiliki setiap kategori *input*. Untuk bobot kategori *input* seperti template .

Tabel 4.6 di bawah ini.

Tabel 4.6 Template Perhitungan UFP

| Tipe Fungsi | Bobot Kompleksitas | | | | | | | | |
|--------------------------------|---------------------------|-----------|-----------|----------------|-----------|-----------|----------------|-----------|-----------|
| | Simple | | | Average | | | Complex | | |
| | Cou nt | Bob ot | Poi nt | Cou nt | Bob ot | Poi nt | Cou nt | Bob ot | Poi nt |
| Exter- nal <i>Inputs</i> | | 3 | | | 4 | | | 6 | |
| Exter- nal <i>Output</i> | | 4 | | | 5 | | | 7 | |

| | | | | | | | | | |
|--------------------------|--|---|--|--|----|--|--|----|----------|
| Internal Logical File | | 7 | | | 10 | | | 15 | |
| External Interface Files | | 5 | | | 7 | | | 10 | |
| External Inquiry | | 3 | | | 4 | | | 6 | |
| TOTAL | | | | | | | | | ? |

Keterangan :

Count : Diisi dengan jumlah *input* yang sesuai dengan kolom. Contoh : Jika terdapat 5 *input* 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 diisi di salah satu kolom saja yang bertanda tanya (?).

Tahap terakhir perhitungan UFP adalah menjumlah semua kategori *input* yang telah dikalikan dengan bobot yang dimiliki oleh masing masing *input* dengan persamaan :

$$UFP = ILF + EIF + EI + EO + EQ$$

Langkah 3. Mengubah satuan UFP kedalam SLOC

SLOC (*Source line of code*) adalah matrik yang didapat dari hasil perhitungan UFP yang kemudian dikonversi sesuai dengan bahasa pemrograman yang digunakan untuk membuat aplikasi seperti persamaan dibawah ini :

$$\text{Size} = \text{UFP} \times \text{SLOC Conservation Rate}$$

Keterangan :

- Size : Hasil perkailain dari variable yang ada pada persamaan.
- UFP : Hasil perhitungan *Unjausted Function Point*
- Conversion Rate : Sesuai dengan bahasa pemrograman yang digunakan tim pengembang, dengan satuan SLOC

Baik COCOMO II pada *Early / Post Design Model* rasio konfersi ukuran SLOC memiliki nilai yang sama. Berikut ini adalah ukuran rasio konversi hasil UFP kedalam SLOC seperti yang tertera pada **Tabel 4.7** dibawah ini

Tabel 4.7 Tabel konversi UFP menjadi SLOC

| Bahasa Pemrograman | SLOC |
|---------------------------|------|
| Access | 38 |
| Ada 83 | 71 |
| Ada 95 | 49 |
| Assembly – Basic | 320 |
| Basic - ANSI | 64 |
| Basic - Complied | 91 |
| C | 128 |
| C ++ | 55 |
| Cobol | 91 |
| Database -- Default | 40 |
| Fifth Generation Language | 4 |
| Fortran 77 | 107 |
| HTML 3.0 | 15 |
| Java | 53 |
| Jovial | 107 |

| Bahasa Pemrograman | SLOC |
|--------------------|------|
| Lisp | 64 |
| Pascal | 91 |

Langkah 4. Menghitung Usaha / *Effort*

Untuk mengetahui usaha yang dibutuhkan dalam pembangunan aplikasi menggunakan persamaan sebagai berikut :

$$\text{Usaha}(\textit{Effort}) = A \times \textit{Size}^E \times M$$

Keterangan :

- A : Konstanta bernilai 2.94 (mengacu pada CO-COMO II.2000)
- Size : Ukuran baris kode yang dihitung dengan UFP, yang memiliki satuan KSLOC
- E : Eksponen yang didapat dari penjumlahan 5 *scale factors*
- M : *Effort Multiplier* adalah variabel yang menentukann/ mengatur usaha, person month untuk menunjukkan ukuran produk perangkat lunak yang sedang dalam pengembangan. Masing masing model pada COCOMO II memiliki jumlah *Effort Multiplier* yang berbeda. Pada *Early Design Model*, *Effort Multiplier* yang digunakan berjumlah 7 dari total 32 *Effort Multiplier* yang ada pada COCOMO II

Scale Factor yang dimaksud pada persamaan untuk mencari usaha berdasarkan faktor - faktor yang berbasis pada rasionalitas dari dan mempengaruhi produktivitas proyek. *Scale factor* sendiri memiliki 5 variasi dan setiap variasi tersebut memiliki rentang tingkatan nilai, dari *very low* sampai *extra high*. Berikut adalah variasi *scale factor* yang ada pada COCOMO II [11]:

1. *Precedentness (PREC)* adalah salah satu variasi *scale factor* yang membahas tentang kesamaan / kemiripan proyek pengembangan perangkat lunak terdahulu dibanding dengan pengembangan perangkat lunak yang dilakukan sekarang. Semakin tinggi kemiripan proyek nilai PREC semakin tinggi
2. *Development Flexibility (FLEX)* variasi yang membahas mengenai kelenturan/ fleksibilitas pengembangan perangkat lunak
3. *Architecture / Risk Resolution (RESL)* variasi yang membahas mengenai penyelesaian risiko yang dialami selama pengembangan perangkat lunak
4. *Team Cohesion (TEAM)* variasi yang mengulas mengenai sumber turbulensi proyek dalam mengsinkronisasi *stakeholder* proyek dengan pengguna dan anggota tim proyek yang lain, faktor lain yang mempengaruhi nilai kohesi tim juga dapat berasal dari budaya *stakeholder*, pengalaman keakraban sebuah tim pengembang dan lain sebagainya.
5. *Process Maturity (PMAT)* variasi terakhir pada faktor skala yang digunakan untuk mengetahui kematangan sebuah rekayasa perangkat lunak pada periode waktu tertentu

Sedangkan untuk pengertian *Effort Multiplier / cost driver (EM)* yang digunakan oleh COCOMO II adalah sebuah faktor yang digunakan untuk menentukan usaha nominal yang menggambarkan proyek perangkat lunak yang sedang dikembangkan. Untuk setiap model pada COCOMO II memiliki jumlah EM yang berbeda. Berikut ini adalah EM yang ada pada 2 model COCOMO II [11].

Tabel 4.8 Effort Multiplier dari Early Design dan Post Architecture COCOMO II

| <i>Early Design Cost Driver</i> | <i>Post- Architecture Cost Driver</i> |
|---------------------------------|---------------------------------------|
| PERS | ACAP, PCAP, PCON |

| | |
|------|---------------------------|
| RCPX | RELY, DATA, CLPX, DOCU |
| RUSE | RUSE |
| PDIF | TIME, STOR, PVOL |
| PREX | APEX, PLEX, LTEX |
| FCIL | TOOL, SITE |
| SCED | SCED |

Untuk mencari semua nilai dari *Scale Factor* dan *effort Multiplier* peneliti mengacu pada literatur yang mengarahkan untuk membuat koesioner yang kemudian diisi oleh tim pengembang/ yang terkait dengan proyek pembangunan perangkat lunak pada penelitian ini[43]. Koesioner dibuat menyesuaikan dengan kriteria penilaian baik dari *Effort multiplier* dan *Scale factor* yang berasal dari literatur COCOMO II yang Selanjutnya akan dilampirkan pada Bab Lampiran pada Tugas Akhir ini.

Langkah 5. Estimasi Biaya Total

Selanjutnya akan dihitung estimasi biaya total untuk pembangunan aplikasi. Selain melakukan estimasi usaha, COCOMO II juga menyediakan perhitungan estimasi biaya, dengan syarat estimasi usaha telah dilakukan kemudian melakukan pendistribusian usaha yang telah dihitung kedalam setiap aktivitas yang dilakukan selama proses pengembangan aplikasi berlangsung.

Pendistribusian usaha aktivitas memiliki satuan prosentase. Penentuan pembagian porsi usaha dapat dilakukan melalui observasi dari pengalaman tim pengembang atau berdasarkan hasil penelitian yang dapat ditemukan pada literatur terkait. Karena peneliti belum mendapatkan rincian mengenai jumlah prosentase usaha setiap aktivitas, peneliti mengacu pada hasil literatur[44]. Proyek yang akan dibuat memiliki skala kecil-sedang(*small-medium*), berikut ini adalah pembagian prosentase aktivitas seperti yang didokumentasikan pada **Tabel 4.9**

Tabel 4.9 Prosentase Distribusi usaha pada aktivitas proyek

| Software Development Phases | |
|-----------------------------|------------|
| Aktivitas (%) | |
| Requirements | 1,6 |
| Specifications | 7,5 |
| Design | 6 |
| Implementation | 52 |
| Acceptance & deployment | 5,5 |
| Project management | 3,8 |
| Configuration management | 4,3 |
| Documentation | 8,4 |
| Training and support | 1 |
| Integration testing | 7 |
| Quality assurance | 0,9 |
| Evaluation and Testing | 2 |
| Total | 100 |

Kemudian peneliti membagi total *effort* yang telah diketahui ke setiap aktivitas proyek dengan mengalikan setiap prosentase aktivitas dengan total effort yang dihitung sebelumnya. Hasil dari perhitungan tersebut didokumentasikan pada template tabel dibawah ini

Tabel 4.10 Template hasil pendistribusian aktivitas

| Software Development Phases | | Effort; (%*pm) |
|--------------------------------------|-----|----------------|
| Aktivitas (%) | | |
| Requirements | 1,6 | |
| Specifications | 7,5 | |
| Design | 6 | |
| Implementation | 52 | |
| Acceptance & deployment | 5,5 | |
| Ongoing life-cycle activities | | |
| Project manage- | 3,8 | |

| | | |
|-----------------------------------|------------|--|
| ment | | |
| Configuration management | 4,3 | |
| Documentation | 8,4 | |
| Training and support | 1 | |
| Quality and testing phases | | |
| Integration testing | 7 | |
| Quality assurance | 0,9 | |
| Evaluation and Testing | 2 | |
| Total | 100 | |

Keterangan :

Aktivitas : Diisi dan disesuaikan dengan aktivitas pada proyek pengembangan

Prosentase (%) : Prosentase aktivitas yang diambil dari literatur. Bisa diisi sesuai dengan *framework* yang digunakan oleh pihak pengembang atau sesuai dengan pertimbangan lain

Effort : Hasil perkalian dari usaha total yang telah dihitung sebelumnya dengan tiap tiap prosentase aktivitas

Hasil dari pembagian setiap aktivitas dapat dijadikan dasar untuk mengetahui biaya yang diperlukan pada aktivitas yang ada dalam proyek dengan menyesuaikan *pay rate/ gaji* setiap orang yang bertanggung jawab pada setiap aktivitas proyek. Standar gaji yang diambil mengacu pada *Indonesian Salary Guide*.

Tabel 4.11 Standar Gaji yang digunakan dalam proses perhitungan biaya total

| Jabatan | Pendidikan/ Pengalaman | Gaji/bulan | Gaji/hari | Gaji/jam |
|------------------------|---------------------------|---------------|--------------|------------|
| <i>Project manager</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 |

Seluruh hasil perhitungan estimasi biaya total pembuatan aplikasi kemudian didokumentasikan kedalam tabel yang dirancang peneliti pada dibawah ini

Tabel 4.12 Template Hasil perhitungan Estimasi Biaya Total

| Aktivitas | Jabatan | Pekerja (Person) | Hari | Standar gaji (bulan) | Total |
|-----------------------------|----------------|-------------------------|-------------|-----------------------------|--------------|
| <i>Software Development</i> | | | | | |
| Requirement | | | | | |
| Specification | | | | | |
| Design | | | | | |
| Implementation | | | | | |
| Acceptance & Deployment | | | | | |
| <i>On Going life-cylce</i> | | | | | |
| Project management | | | | | |
| Configuration management | | | | | |
| Documentation | | | | | |
| Training and support | | | | | |

| Aktivitas | Jabatan | Pekerja (Person) | Hari | Standar gaji (bulan) | Total |
|----------------------------|---------|------------------|------|----------------------|-------|
| <i>Quality and testing</i> | | | | | |
| Integration testing | | | | | |
| Quality assurance | | | | | |
| Evaluation and Testing | | | | | |
| TOTAL | | | | | |

a. **Perbandingan pengembangan aplikasi serupa dari penyedia jasa lain**

Tujuan kegiatan ini untuk mengetahui apakah pengembangan aplikasi dengan fungsi yang sama dari pihak ketiga/ pengembang memiliki harga yang mahal ketimbang penyedia jasa lain.

Hal yang pertama peneliti lakukan adalah menyiapkan daftar fungsi yang terdapat pada aplikasi yang dikembangkan. Kemudian melakukan *benchmark* kepada pihak lain (penyedia jasa lain) terkait fungsi dan harga yang mereka tawarkan untuk aplikasi tersebut. Fungsi yang dimaksud terdapat pada **Tabel 4.13**. Tidak lupa untuk fungsi tambahan yang ditawarkan penyedia jasa lain akan dijadikan sebagai pertimbangan dalam melakukan estimasi harga aplikasi *e-Performance* tersebut.

Tabel 4.13 Fungsi Pembanding

| No | Kebutuhan Fungsional |
|----|--|
| 1 | Aplikasi dapat menampilkan informasi perangkat daerah |
| 2 | Aplikasi dapat meng- <i>input</i> data pegawai sesuai tingkat jabatan |
| 3 | Aplikasi dapat melakukan <i>update</i> data pegawai |
| 4 | Aplikasi dapat memberikan evaluasi berupa kuis dalam bentuk form soal untuk evaluasi pegawai pada kurun waktu tertentu |

| | |
|-----|--|
| 5 | Aplikasi dapat menampilkan indikator yang berhasil dicapai untuk masing masing pegawai |
| 6. | Aplikasi yang dapat menampilkan informasi tugas yang dilakukan setiap pegawai |
| 7. | Aplikasi yang dapat menginputkan aktivitas pegawai |
| 8. | Aplikasi yang dapat menampilkan capaian kegiatan |
| 9. | Aplikasi yang dapat menampilkan status aktivitas perangkat daerah |
| 10. | Aplikasi yang dapat menampilkan nilai akhir tiap pegawai daerah |

Peneliti Selanjutnya membuat *template* tabel yang berisi *checklist* ketersediaan fungsi yang sama dari fungsi ditawarkan dari pihak penyedia jasa lain seperti pada Tabel 4.14.

Tabel 4.14 Template perbandingan dengan aplikasi dari penyedia lain

| No | Fungsi Pembanding | Aplikasi A | Aplikasi B |
|--------------|-------------------|------------|------------|
| | | | |
| | | | |
| Total | | | |

Keterangan :

Fungsi Pembanding : Diisi dan disesuaikan fungsi yang telah didefinisikan pada **Tabel 4.13** Fungsi Pembanding

Aplikasi A/ B : Diisi dengan *checklist* (V) ketersediaan fungsi yang ditawarkan pada aplikasi A maupun B sesuai dengan fungsi pembanding. Jika fungsi pembanding tidak dapat ditemukan pada salah satu aplikasi dapat diisi dengan (-)

Total : Diisi dengan total harga/ estimasi harga yang diberikan oleh penevedia jasa lain

4.3.2. Function Analysis Phase

a. Perhitungan Biaya Kebutuhan Fungsional

Tahap ini berisi aktivitas menghitung kebutuhan fungsional dengan menggunakan COCOMO II. Hasil perhitungan yang berupa perhitungan *effort* dan waktu pengerjaan yang didapat, peneliti kemudian memasukkannya kepada setiap jabatan dan aktivitas pada proyek. Hal ini dilakukan sebagai masukan awal untuk proses perhitungan kebutuhan fungsional yang ada pada aplikasi. Berikut adalah detail pembagian *effort* dan waktu kepada aktivitas dan jabatan proyek yang tersedia pada Tabel 4.15

Tabel 4.15 Template Detail Pembagian effort dan waktu pengerjaan

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (per-bulan) | Total |
|------------------------------------|-----------------|--------|-------|--------------------------|-------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Bisnis | | | | |
| <i>Specifications</i> | Analisis bisnis | | | | |
| <i>Design</i> | Analisis bisnis | | | | |
| <i>Implementation</i> | Programmer | | | | |
| <i>Acceptance & deployment</i> | Tehnisi | | | | |
| TOTAL | | | | | |

Keterangan :

Kelompok : Diisi sesuai dengan aktivitas pada siklus pembangunan *software* sesuai pada penjelasan yang ada di Bab 2. Contoh isian : *Requirement*

Jabatan : Diisi sesuai jabatan yang akan bertanggung-jawab pada aktivitas tersebut. Contoh isian : *Business analyst*.

Person : Diisi sesuai jumlah pekerja yang disesuaikan dengan hasil perhitungan *effort* yang diperoleh melalui teknik *COCOMO II*. Contoh : *effort* = 8 orang. Requirement = 1

- orang; Coding = 3 orang, dll.
- Days : Diisi sesuai dengan jumlah waktu pengerjaan masing-masing aktivitas, yang disesuaikan dengan total waktu pengerjaan yang diperoleh dari teknik COCOMO II. Contoh : Waktu = 80 hari. Requirement = 15 hari; Coding = 35 hari, dll.
- Standar Gaji : Diisikan sesuai dengan harga gaji perhari sesuai dengan jabatan masing-masing. Contoh : Rp 400.000.
- Total : Diisi dengan hasil perkalian dari kolom person, days dan standart gaji. Contoh : person = 2; days = 15 hari; Gaji = Rp 400.000; Total = Rp 12.000.000
- TOTAL : Diisikan sesuai dengan jumlah total biaya masing-masing aktivitas.

b. Pemilihan Objek Penelitian

Aktivitas yang dilakukan pada fase ini adalah menghitung dan mengurutkan kebutuhan fungsional berdasarkan hasil perhitungan COCOMO II, Biaya dan perhitungan Bobot. Adapun persamaan Bobot yang akan dihitung adalah :

$$\text{Bobot} = \frac{\text{Biaya fitur}}{\text{Biaya total fitur}} \times 100\% \quad (4.4)$$

Hasil dari perhitungan bobot tersebut kemudian dimasukkan kedalam tabel yang tertera sesuai pada template pada Tabel 4.16 :

Tabel 4.16 Template Estimasi biaya kebutuhan Fungsional dan hasil Bobot tiap Fungsi

| No | Kebutuhan Fungsional | COCOMO II | Biaya | Bobot |
|----|----------------------|-----------|-------|-------|
| 1. | | | | |
| 2. | | | | |
| 3. | | | | |

| No | Kebutuhan Fungsional | COCOMO II | Biaya | Bobot |
|--------------|----------------------|-----------|-------|----------|
| TOTAL | | | | ? |

Keterangan :

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

COCOMO II : Diisi dengan nilai *COCOMO II* yang diperoleh dari hasil perhitungan. Contoh : 45.

Biaya : Diisikan dengan estimasi biaya setiap fungsi 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.

Setelah mengetahui urutan bobot dan biaya setiap fungsi, diperlukan indikator lain untuk menentukann objek yang akan dikerjakan selama Tugas Akhir. Indikator tersebut berkaitan dengan lama waktu pengerjaan fungsi dan tingkat kerumitan aplikasi website yang sedang dibangun berdasarkan lama waktu pengembangan sebuah website[45] seperti pada Tabel 4.17

Tabel 4.17 Kategori Ukuran Aplikasi berdasarkan waktu pengembangan

| Total Jam Pengembangan | Kategori |
|------------------------|--------------------------|
| <600 | <i>Simple/ Sederhana</i> |
| 600-900 | <i>Complex</i> |

| Total Jam Pengembangan | Kategori |
|------------------------|---------------------|
| >900 | <i>Very Complex</i> |

Berdasarkan indikator diatas, peneliti memilih Complex – Very Complex dengan lama pengerjaan sekitar 600 hingga diatas 900 jam. Sesuai indikator yang telah dijelaskan, peneliti dapat menentukan objek yang kemudian dibuatkan F.A.S.T Diagram dan menjadi objek yang akan dikerjakan selama Tugas Akhir. Hasil dari pemilihan objek penelitian tersebut didokumentasikan kedalam Tabel 4.18 yang tertera pada template tabel dibawah ini :

Tabel 4.18 Template Penentuan Objek Penelitian

| No | Kebutuhan Fungsional | Hours | PM | Estimnasi Biaya | Keterangan |
|----|----------------------|-------|----|-----------------|------------|
| | | | | | |
| | | | | | |

Keterangan :

- No : Diisi sesuai 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 informasi Saldo.
- Hours : Diisikan dengan jumlah waktu yang dibutuhkan untuk menyelesaikan masing-masing aktivitas pengerjaan fungsi tersebut dalam satuan jam. Contoh : 209 jam.
- PM : Diisi dengan TOTAL Person Month yang dihasilkan dari perhitungan usaha setiap fitur
- Estimasi Biaya : Total harga yang dihasilkan dalam perhitungan kebutuhan fungsional. Contoh : Fungsi A memiliki harga total sebesar Rp14.002.000 yang kemudian dituliskan

pada kolom estimasi biaya sebesar yang tertera

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

c. **Pembuatan F.A.S.T Diagram**

Seperti yang telah dijelaskan pada Bab 3 Metodologi, untuk mengetahui fungsi apa yang harus ada atau yang tidak diperlukan pada aplikasi, menggunakan diagram FAST, di bawah ini dijelaskan bagaimana penggunaan diagram FAST pada pengerjaan Tugas Akhir ini :

Langkah – langkah pembuatan FAST Diagram

Pembangunan F.A.S.T diagram dimulai dengan menentukann fungsi. Pembangunan ini didasarkan pada pemberian jawaban terhadap beberapa pertanyaan berikut ini [46]:

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 4.1) dan *Task Oriented* (Gambar 4.2). Dalam FAST Diagram terdapat 2 klasifikasi yang berbeda untuk mendefinisikan fungsi, yakni :

1. *Basic Function*

Mendeskrripsikan 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*

Keberadaannya 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 [15]:

- *Critical Path Function*

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

- *Higher Order Function*

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

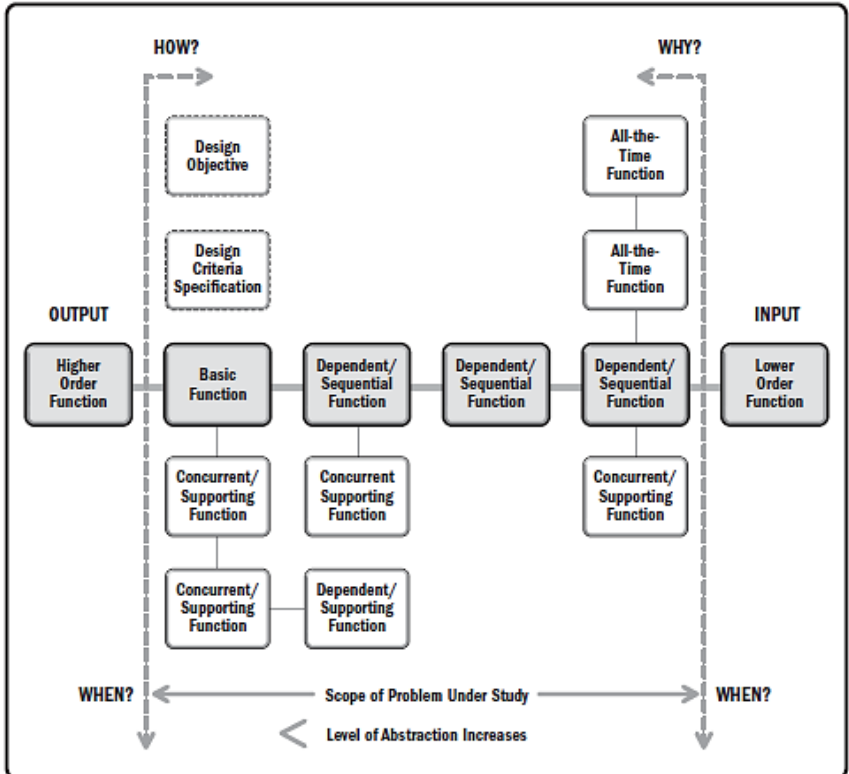
- *Lower Order Function*

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.

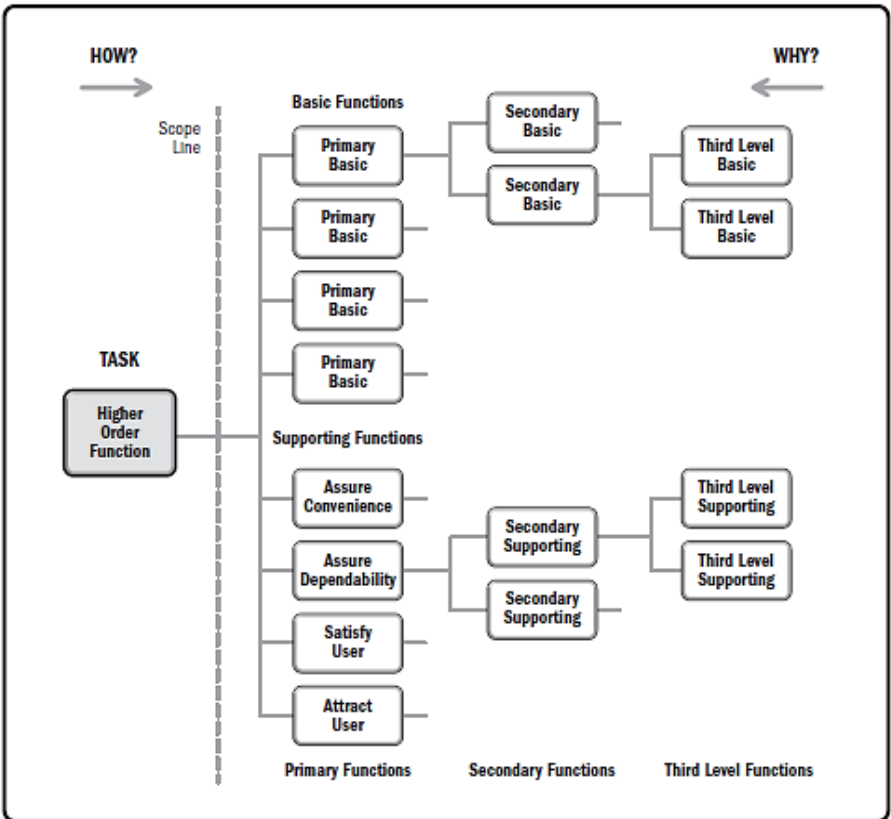
Pada FAST Diagram terdapat beberapa *note* yang perlu dipahami :

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

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

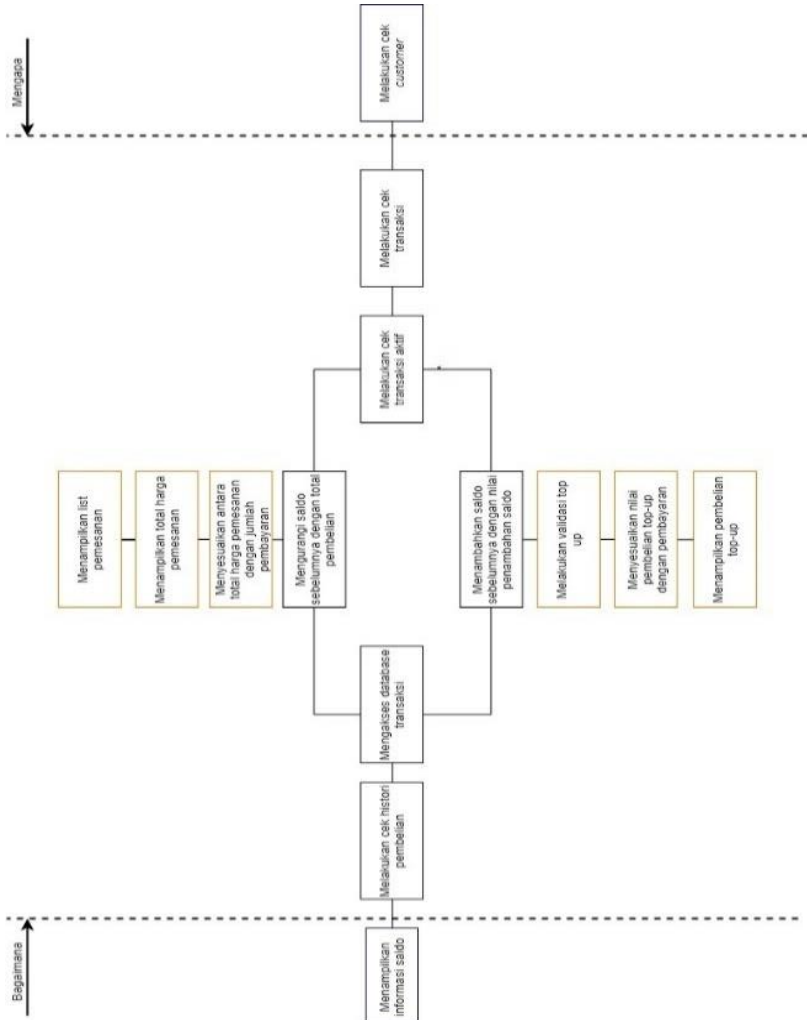


Gambar 4.1 Technical Oriented FAST[5]



Gambar 4.2 Customer (TASK) Oriented FAST[5]

Berikut ini adalah hasil penggunaan diagram F.A.S.T yang telah dibuat sesuai dengan mengikuti *Task* pada fungsi aplikasi proyek TI dari penelitian sebelumnya seperti yang tertera pada Gambar 4.3 di bawah ini.



Gambar 4.3 Contoh penggunaan F.A.S.T Diagram pada pembuatan aplikasi [5]

Pada pembuatan FAST Diagram ini, peneliti berfokus pada pembuatan FAST Diagram sesuai pada objek yang terpilih dari proses perhitungan pemilihan objek yang dilakukan sebelumnya, yaitu berdasarkan pada hasil perhitungan bobot pada setiap fungsi/ fitur aplikasi.

4.3.3. Creative Analysis Phase

Dalam fase ini dilakukan kegiatan pengolahan data dan pembuatan skenario solusi. Beberapa alternatif solusi yang diuji cobakan pada penelitian berfokus pada penghematan biaya. Alternatif solusi yang digunakan pada penelitian ini adalah :

1. Penggunaan skenario dengan merubah isi variabel *cost driver* dalam formula COCOMO II. *Cost driver* menggambarkan karakteristik pengembangan *software* yang mempengaruhi hasil *effort* / usaha (dalam satuan *Person Month*). Dalam *cost driver* memiliki beberapa *rating* untuk menunjukkan nilai usaha yang dibutuhkan disebut dengan *Effort Multiplier* (EM). Nilai *rating Effort Multiplier* yang diubah memiliki hubungan non fungsionalitas pada proses pengembangan aplikasi yang berhubungan pada kecakapan tim pengembang terhadap aplikasi yang akan dibuat. Beberapa *rating* EM yang diubah nilainya adalah :
 - *Personnel Capability* (PERS)
EM yang didapatkan dari model *Early Design* yang berasal dari gabungan *rating* EM pada model *Post Design* seperti yang dijelaskan pada BAB II dasar teori. *Rating* ini menggambarkan kemampuan yang dimiliki oleh tim pengembang baik dari segi analisis, kemampuan dalam membuat program, serta keberlanjutan anggota tim pengembang aplikasi (*personnel continuity*). Nilai *rating* mengacu dari hasil pengisian koesioner penilaian *Cost Driver* yang dijelaskan pada BAB V Implementasi yang kemudian diubah untuk menghasilkan *effort* yang lebih rendah dibandingkan dengan sebelum uji coba skenario dilakukan.

Hasil skenario tersebut akan didokumentasikan dalam bentuk tabel seperti pada Tabel 4.19

Tabel 4.19 Template tabel skenario Effort Multiplier

| Kategori | Effort Multiplier | Nominal | Skala | Hasil kalkulasi |
|---------------------------------|-------------------|---------|-------|-----------------|
| | | | | |
| Total | | | | |
| Total Effort Multiplier sebelum | | | | |

Keterangan :

Kategori : Diisikan sesuai jenis kategori *Effort Multiplier* pada model *Early Design*

Effort Multiplier : Diisikan sesuai dengan 17 *effort multiplier* yang ada pada COCOMO II

Nominal : Diisikan sesuai dengan nilai rating mulaih dari 0-5

Skala : Diisi mengikuti nominal yang dihasilkan. Contoh untuk Kategori PERS terdiri dari EM ACAP, PCAP, dan PCON, setiap EM bernilai 1,4, dan 3. Kemudian nominal tersebut dijumlah dan dipetakan sesuai dengan skala hasil penjumlahan nominal.

Hasil Kalkulasi : Diisi dengan nilai yang diperoleh dari skala. Contoh : Untuk skala pada kategori PERS memiliki nilai High, maka nilai hasil kalkulasi berisi 0,83

Total : Diisi dengan hasil penjumlahan kolom Hasil kalkulasi untuk mendapatkan EM yang diperlukan

Total EM sebelum : Diisi dengan EM yang dihasilkan pada perhitungan sebelum skenario dilakukan

2. Skenario berikutnya adalah mengubah nilai *Scale Factors* yang terdapat pada eksponen E pada persamaan dalam mencari *effort* yang telah dijelaskan pada BAB II dasar teori. *Scale Factor* memiliki 5 kategori yang berperan dalam mempengaruhi hasil *effort* pada perhitungan COCOMO II. Kategori yang diubah dalam skenario adalah :

- *Team Cohesion* (TEAM)

Salah satu kategori *scale factor* yang menggambarkan kemampuan tim dalam mengatasi permasalahan dalam proyek akibat masalah komunikasi terhadap pengguna, *stakeholder* serta di dalam tim pengembang itu sendiri.

Dari hasil skenario akan didokumentasikan dalam bentuk tabel seperti yang tertera pada Tabel 4.20

Tabel 4.20 Template tabel hasil skenario Scale Factor

| Scale Factor | Skala | Hasil kalkulasi |
|-----------------------------|-------|-----------------|
| | | |
| Total | | |
| Total Scale Factors Sebelum | | |

Keterangan :

Scale Factor : Diisikan sesuai dengan 5 SC yang dimiliki COCOMO II

Skala : Diisi dengan skala yang didapat dari hasil skenario, dengan rentan skala dari Very Low hingga Very High. Contoh : untuk kategori SC Precedents memiliki skala

Nominal, maka pada kolo Skala dituliskan Nominal

- Hasil Kalkulasi : Diisi dengan nilai yang diperoleh dari skala. Contoh : Untuk skala pada kategori Precedents memiliki nilai Nominal, maka nilai hasil kalkulasi berisi 3,72
- Total : Diisi dengan hasil penjumlahan kolom Hasil kalkulasi untuk mendapatkan SC yang diperlukan
- Total SC : Diisi dengan SC yang dihasilkan pada sebelum perhitungan sebelum skenario dilakukan

Untuk kedua sekario pertama, setelah dilakukan pengubahaan pada *Scale Factor/ Effort Multiplier* kemudian dilanjutkan dengan menghitung usaha seperti biasa yang telah dijelaskan pada sub bab sebelumnya.

Hasil pada fase ini kemudian didokumentasikan kedalam tabel seperti yang tertera pada template Tabel 4.21 di bawah ini :

Tabel 4.21 Template Hasil Alternatif Kebutuhan Fungsi

| No. | Kebutuhan Fungsional | Alternatif | Kode Skenario |
|-----|----------------------|------------|---------------|
| | | | |

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

Kode Ske- : Diisi menggunakan format kode skenario
nario : SK-Nomor Alternatif-Nomor urut. Contoh
: SK-AL1-001

4.3.4. Evaluation Phase

Fase ini dilakukan untuk mengetahui apakah alternatif solusi yang telah dikerjakan pada fase sebelumnya layak atau tidak dilanjutkan pada fase Selanjutnya dalam *Value Engineering*. Kegiatan yang dilakukan adalah mendokumentasikan hasil alternatif tersebut kedalam Tabel 4.22. Jumlah tabel mengikuti jumlah skenario yang dibuat, seperti jika total skenario yang ada berjumlah 5, maka harus ada 5 tabel yang dibuat untuk mendokumentasikan hasil perubahan biaya pada skenario tersebut.

Tabel 4.22 Template Hasil Perubahan Biaya berdasarkan Skenario

| No. | Kebutuhan Fungsional | Biaya awal | Alternatif Kebutuhan Fungsional | Biaya Akhir |
|-----|----------------------|------------|---------------------------------|-------------|
| | | | | |
| | | | | |
| | TOTAL : | | TOTAL : | |

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 infor-

masi 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.

3. Pencarian Indeks Nilai

Aktivitas yang dilakukan pada tahapan ini adalah mendokumentasikan perubahan yang dihasilkan sebelum dan setelah perhitungan skenario. Masing masing skenario dihitung dan kemudian dimasukkan pada Tabel 4.23 untuk mengetahui skenario yang layak dilanjutkan

Tabel 4.23 Tenplate Hasil Pencarian Indeks

| No | Alternatif Kebutuhan Fungsional | Hasil Pencarian Indeks | Keterangan |
|----|---------------------------------|------------------------|------------|
| | | | |
| | | | |

Keterangan :

- No : Diisikan sesuai urutan jenis fungsi. Contoh : 1,2, dst.
- Alternatif Kebutuhan Fungsional : Diisikan sesuai dengan skenario yang berhasil dibuat. Contoh : Penambahan SDM.

Hasil Pen- : Diisikan dengan nilai yang diperoleh setelah
 carian In- perhitungan dilakukan. Contoh : 1.
 deks

Keterangan : Diisikan dengan keterangan sesuai kategori
 berikut :



: $Nt/Np < 1$



: $Nt/Np = 1$



: $Nt/Np > 1$

Pemilihan alternatif didasarkan pada skenario yang berhasil
 masuk pada kategori $Nt / Np > 1$ dari hasil perhitungan indeks
 yang dilakukan.

4. **Pemilihan Rekomendasi**

Pemilihan rekomendasi didasarkan dengan rumusan masalah
 yang ada pada Bab 1 yaitu solusi yang memiliki biaya yang
 paling minimum dan dengan waktu tersingkat yang dihasilkan
 untuk pengerjaan proyek.

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. Pembahasan Data Eksisting

Sub bab ini berisi mengenai pembahasan dokumen eksisting yang didapat dari tim pengembang berupa dokumen KAK (Kerangka Acuan Kerja) .

5.1.1.1. Kebutuhan Fungsional Aplikasi

Dari dokumen eksisting yang ada, diketahui bahwa terdapat 7 fungsi yang dibutuhkan *client* pada aplikasi yang akan dikembangkan oleh tim pengembang seperti yang ada pada Tabel 5.1.

Tabel 5.1 Kebutuhan Fungsional Aplikasi

| No | Kebutuhan Fungsional |
|----|--|
| 1 | Aplikasi dapat menampilkan informasi perangkat daerah |
| 2 | Aplikasi dapat meng- <i>input</i> data pegawai sesuai tingkat jabatan |
| 3 | Aplikasi dapat melakukan <i>update</i> data pegawai |
| 4 | Aplikasi dapat memberikan evaluasi berupa kuis dalam bentuk form soal untuk evaluasi pegawai pada kurun waktu tertentu |
| 5 | Aplikasi dapat menampilkan indikator yang berhasil dicapai untuk masing masing pegawai |
| 6. | Aplikasi yang dapat menampilkan informasi tugas yang dilakukan setiap pegawai |
| 7. | Aplikasi yang dapat menginputkan aktivitas pegawai |
| 8. | Aplikasi yang dapat menampilkan capaian kegiatan |
| 9. | Aplikasi yang dapat menampilkan status aktivitas |

| No | Kebutuhan Fungsional |
|-----|---|
| | perangkat daerah |
| 10. | Aplikasi yang dapat menampilkan nilai akhir tiap pegawai daerah |

5.1.1.2. Pihak yang Terlibat dalam Proyek Pembuatan Aplikasi

Selama proyek pembuatan aplikasi ini dijalankan ada beberapa pihak yang memegang jabatan yang diusulkan oleh tim pengembang beserta gaji dan lama mereka bekerja untuk tiap jabatan tersebut sesuai . Penjelasan tersebut dimuat pada Tabel 5.2.

Tabel 5.2 Pihak yang terlibat dalam Proyek

| No | Resource (Jabatan) | SATUAN | RATE / SAT | QTY |
|----|--------------------|----------|------------|-----|
| 1 | Team Leader | Man Days | Rp600.000 | 1 |
| 2 | Analisis Sistem | Man Days | Rp450.000 | 2 |
| 3 | Programmer | Man Days | Rp400.000 | 3 |
| 4 | Penguji Program | Man Days | Rp250.000 | 1 |
| 5 | Perawat Aplikasi | Man Days | Rp150.000 | 1 |
| 6 | Administrasi | Man Days | Rp100.000 | 1 |

Berdasarkan daftar yang ada pada Tabel 5.2 pembagian tugas untuk masing masing jabatan adalah sebagai berikut :

1. *Team Leader* : merupakan penanggung jawab utama proyek, jabatan tertinggi dari semua jabatan dan bertugas untuk *memonitor* kerja setiap jabatan yang lain agar sesuai dengan sumber daya dan waktu yang telah ditentukan
2. *Analisis sistem / System Analyst* : merupakan bagian yang bertugas untuk melakukan analisis sistem yang ada pada penda setempat.

3. Programmer : merupakan bagian yang bertugas untuk membuat dan mengembangkan aplikasi
4. Penguji Program / *Program Tester* : merupakan pihak yang bertanggung jawab untuk melakukan *testing* program yang telah dibuat oleh programmer
5. Perawat aplikasi / *Application Maintenance* : pihak yang melakukan perawatan (baik *bug*, atau *error* dll) aplikasi setelah tahap pembuatan aplikasi selesai.
6. Adminstrasi : pihak yang bertugas melakukan pencatatan keuangan serta sesuatu yang bersifat administratif selama proyek berlangsung dan membuat laporan yang dibutuhkan selama dan setelah proyek dilaksanakan.

5.1.1.3. Tahapan Pelaksanaan Proyek Pembangunan Aplikasi

Dari KAK yang didapat dari pihak ketiga, berikut ini usulan tahapan kegiatan sebagaimana yang tertera 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 | Pemeliharaan (termasuk Jaminan Security) |
| 11. | Proyek Selesai |

5.1.1.4. Waktu Pelaksanaan Proyek Pembangunan Aplikasi

Sesuai yang dijelaskan pada KAK, pengerjaan proyek ini memakan waktu selama 6 bulan yang dibagi dalam 3 bulan untuk analisis sistem dari *e-Performance* untuk *client* dan 3 bulan Selanjutnya dilakukan pembangunan dan pembuatan aplikasi. Keseluruhan kegiatan yang telah disebutkan membutuhkan biaya dengan total Rp220.000.000, biaya tersebut belum termasuk dengan biaya kegiatan perawatan/ maintenance aplikasi. Berikut adalah penjelasan detail pembagian kerja usulan dari pihak ketiga tertera pada Tabel 5.4

Tabel 5.4 Detail pembagian workload perbulan untuk tiap jabatan

| No | Resource (Jabatan) | BULAN | 1 | | | | 2 | | | | 3 | | | |
|----|---------------------------------------|----------------------|---|---|---|---|---|---|---|---|---|-----------------|---|---|
| | | Minggu | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| | | Budget | | | | | | | | | | | | |
| 1 | Team Leader | Rp72.000.000 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 2 | Analisis Sistem | Rp54.000.000 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 3 | Programmer | Rp72.000.000 | | | | | | | | | 5 | 5 | 5 | 5 |
| 4 | Penguji Program | Rp10.000.000 | | | | | | | | | | | | |
| 5 | Administrasi | Rp12.000.000 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| | Total biaya | Rp220.000.000 | | | | | | | | | | | | |
| | belum termasuk maintenance program | | | | | | | | | | | | | |
| 4 | | | | 5 | | | | 6 | | | | Total Manday | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | | | |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 120 | | |
| | | | | | | | | | | | | 60 | | |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | | | | | 60 | | |
| | | | | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 40 | | |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 120 | | |

Pengisian hari pada tiap pekerja didapat dari asumsi bahwa dalam satu bulan setiap pekerja menghabiskan waktu 20 hari dan sudah termasuk dipotong hari libur. Belum untuk aktivitas pengamanan/ *security* pada aplikasi.

5.1.2. Pembahasan Data Koesioner Penilaian COCOMO II

Dibawah ini adalah penjabaran hasil lembar penilaian *scale factor* dan *effort multiplier*. Responden dalam pengisian lembar penilaian adalah salah satu tim pengembang pada proyek pengembangan aplikasi *e-Performance*. Untuk melihat detail pertanyaan lembar penilaian terdapat pada Lampiran B

5.1.2.1. Perhitungan *Scale Factors*

Pada Tabel 5.5 Berisi nilai 5 *Scale Factors* dari aplikasi *e-Performance* yang akan dikembangkan oleh pihak pengembang. Nilai tersebut adalah bobot dari hasil penilaian masing-masing *scale factors* sesuai yang dijelaskan pada BAB 2 – Dasar Teori.

Tabel 5.5 Nilai Scale Factor untuk aplikasi e-Performance

| <i>Scale Factor</i> | Jawaban pada Koesioner | Nilai hasil |
|---------------------|------------------------|--------------|
| PREC | Nominal | 3,72 |
| FLEX | Very High | 4,05 |
| RESL | Low | 1,41 |
| TEAM | Low | 4,38 |
| PMAT | Nominal | 4,68 |
| Total | | 19,44 |

5.1.2.2. Perhitungan *Effort Multipliers*

Berbeda dengan perhitungan *Effort Multipliers* pada *Post-Architecture Model*, pada *Early Design Model* hanya memiliki 7 kategori saja. Namun ketujuh kategori tersebut dibentuk dari gabungan dari 17 *effort multiplier* pada *Post-Architecture Model*. Untuk *cost driver* dari *early model* adalah gabungan dari *cost driver* yang dijumlahkan dan hasilnya dialokasikan untuk skala penilaian *Early Design Model* dari tingkatan sangat rendah (Very Low) sampai ekstra tinggi (*Extra High*).

Beberapa hasil kalkulasi dapat dilihat pada BAB 2 – Dasar Teori. Dibawah ini adalah pembahasan mengenai menentukann nilai hasil pada *effort multiplier*.

Tabel 5.6 Nilai Skala cost driver

| VL (Very Low) | L (Low) | N (Nominal) | H (High) | VH (Very High) | EH (Extra High) |
|------------------|------------|----------------|-------------|-------------------|--------------------|
| 1 | 2 | 3 | 4 | 5 | 6 |

Tabel 5.7 Hasil perhitungan Effort Multiplier

| <i>Cost Driver (Early Design Model)</i> | <i>Cost Driver (Post-Arch Model)</i> | Nilai Skala Cost Driver | Total Skala Cost Driver | <i>Effort Multiplier</i> |
|---|--|----------------------------|----------------------------|--------------------------|
| PERS | ACAP | 4 | 9 | 1 |
| | PCAP | 3 | | |
| | PCON | 2 | | |
| RUSE | RUSE | 1,07 | 1,07 | 1,07 |
| RECPX | RELY | 3 | 11 | 0,83 |
| | DATA | 3 | | |
| | CPLX | 3 | | |
| | DOCU | 2 | | |
| PDIF | TIME | 3 | 10 | 1,29 |
| | STOR | 3 | | |
| | PVOL | 4 | | |
| PREX | AEXP | 2 | 11 | 0,87 |
| | PEXP | 4 | | |
| | LTEX | 5 | | |
| FCIL | TOOL | 4 | 9 | 0,73 |
| | SITE | 5 | | |
| Total | | | | 0,56 |

5.2. Function Phase Analysis

Subbab ini menjelaskan bagaimana alur pengerjaan dari *Function Phase Analysis* penelitian Tugas Akhir. Berdasarkan hasil estimasi biaya setiap fungsi yang dibutuhkan aplikasi, kemudian akan dilanjutkan dengan pembuatan F.A.S.T Dia-

gram. Diagram ini ditujukan untuk mengetahui keterkaitan antar komponen agar dapat menjalankan aplikasi seperti yang diharapkan. Sebelum pembangunan F.A.S.T Diagram, peneliti menentukann objek (dalam hal ini kebutuhan fungsional aplikasi) yang layak untuk dibuat dalam F.A.S.T Diagram dengan menggunakan metode nilai bobot[5].

Tools yang dimanfaatkan untuk pembuatan F.A.S.T. Diagram pada penelitian ini adalah draw.io.

5.3. Hambatan penelitian

Berdasarkan kelengkapan data dan penggunaan metode pada penelitian Tugas Akhir ini, peneliti menghadapi beberapa kesulitan selama pengerjaannya seperti :

1. Masih terbatasnya literatur terkait *value engineering* pada dunia *Project Management IT*. Literatur didapatkan berdasarkan fase – fase yang telah diterjemahkan pada penelitian sebelumnya dengan Topik serupa. Namun untuk tingkat kevalidan dari penelitian sebelumnya masih dapat ditambah atau dikurangi.
2. Kelengkapan data KAK yang didapat dari pihak pengembang belum sepenuhnya lengkap. Tidak ditemukan informasi terkait sumber daya yang dibutuhkan untuk pembangunan aplikasi, sehingga peneliti harus menentukann estimasi biaya berdasarkan usaha, dan *mandays* untuk mengetahui tingkat kemahalan pembangunan aplikasi dengan dibandingkan dengan biaya aplikasi serupa
3. Sulitnya mencari harga untuk aplikasi serupa karena setiap aplikasi yang ditujukan untuk Pemerintah Daerah memiliki regulasi tersendiri, walaupun ada beberapa fungsi yang serupa namun beberapa sisanya berbeda. Dan tidak semua pengembang yang telah mengembangkan aplikasi serupa mau memberikan rincian biaya dari proyek tersebut.
4. Peneliti kesulitan mencari literatur terkait estimasi biaya selain yang digunakan pada penelitian sebelumnya.

Dikarenakan peneliti memilih metode estimasi biaya *software* COCOMO II dibutuhkan pemahaman terkait siklus pengembangan proyek, karena pada hal ini COCOMO II memiliki beberapa model yang disesuaikan dengan tahapan pada siklus hidup pengembangan aplikasi. Hal ini juga sangat dipengaruhi dengan kelengkapan data/ dokumentasi dari pengembangan proyek.

Halaman ini sengaja dikosongkan

BAB VI

HASIL DAN PEMBAHASAN

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

6.1. *Information Phase*

6.1.1. Pengkajian Data Masa Lalu

Mengenai studi kasus yang digunakan untuk penelitian ini adalah mengenai aplikasi E-performance yang fungsi utamanya untuk melakukan penilaian kinerja kepada seluruh jabatan pada sebuah instansi pemerintah dengan terstruktur dan objektif. Aplikasi ini dibangun berbasis web dengan pembagian hak akses fitur untuk setiap jabatan.

Dengan objektif yang dimaksud, pihak ketiga kemudian memberikan rekomendasi fitur yang bisa dijadikan tools untuk mencapai tujuan untuk penilaian kinerja pegawai.

Berikut adalah rekomendasi fitur yang diusulkan oleh pihak ketiga berdasarkan tujuan client untuk melakukan penilaian kinerja :

1. Menampilkan Data Utama
Berisi fitur tambahan yang menampilkan informasi log dan daftar perangkat daerah yang ada pada instansi terkait.
2. Master Pegawai
Fitur yang digunakan untuk menampilkan informasi terkait :
 - Informasi Pegawai yang meliputi NIP, Nama, Tingkat, Tanggal Aktif dan jabatan yang dimiliki pegawai tersebutSelain itu fitur ini difungsikan untuk memonitor dan *input* data pegawai.
3. Aktivitas

Fitur ini digunakan untuk menampilkan aktivitas yang dilakukan berdasarkan jabatan, hal tersebut dilakukan untuk memudahkan kegiatan monitoring setiap capaian aktivitas pegawai. Informasi yang terdapat pada fitur ini berupa :

- Informasi aktivitas berdasarkan jabatan, berisi tentang nama pegawai jabatan dan aktivitas yang harus dilaksanakan pegawai bersangkutan.
- Informasi capaian aktivitas pegawai, berisi mengenai status aktivitas pegawai (terlaksana / belum).

4. Sinkronisasi Data E-Monev

Fitur yang ditujukan untuk checklist kesesuaian data. Data yang dimaksud adalah sebagai berikut :

- *Update* data emonev
- Sinkronisasi realisasi sp2d
- *Update* data pelaporan *online*
- Indikator kinerja

5. Perilaku Kerja

Fitur ini digunakan untuk pelaksanaan tes kompetensi yang berisikan soal soal terkait perilaku kerja. Selain itu fitur ini juga dapat menampilkan informasi yang digunakan selama monitoring pelaksanaan penilaian perilaku kerja.

6. Pemberian Tugas

Fitur yang ditujukan untuk menampilkan informasi hasil perekapan tugas tiap pegawai. Informasi yang dimaksud adalah :

- Nama pegawai yang bersangkutan beserta jabatan
- Deskripsi aktivitas yang dilakukan
- Sasaran/ objek kerja yaitu yang bertanggung jawab

- dan *checklist* aktivitas

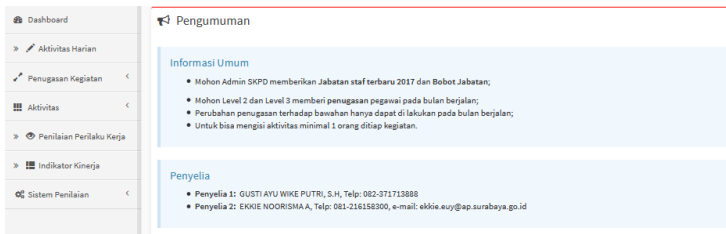
7. Sistem Penilaian

Berikut ini adalah beberapa fungsi yang ada pada fitur Sistem Penilaian :

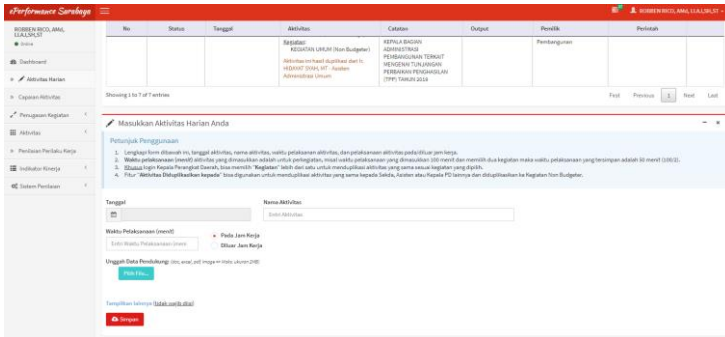
- Menampilkan status perangkat daerah (dengan status : dalam antrian, proses antrian, verifikasi, selesai)
- Menampilkan hasil nilai akhir untuk masing masing pegawai
- Menampilkan nilai akhir untuk semua pegawai dalam suatu instansi

a. Tampilan Aplikasi yang Diharapkan

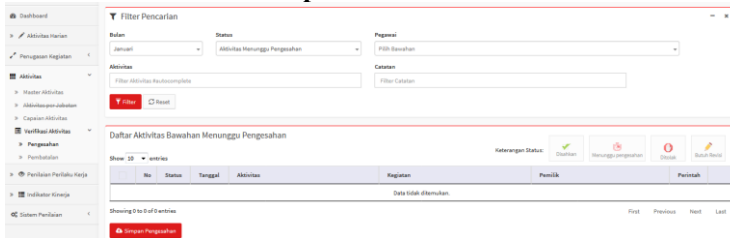
Penggunaan aplikasi *e-performance* sudah banyak diterapkan oleh beberapa Pemerintah Daerah, seperti wilayah Surabaya, Madiun, dan lainnya. Untuk tampilan/ desain yang nantinya digunakan untuk aplikasi yang akan dikembangkan ini mengacu pada tampilan *e-performance* Surabaya seperti yang tertera pada Gambar 6.1, Gambar 6.2, Gambar 6.3



Gambar 6.1 Tampilan antar muka pembuka aplikasi e-Performance



Gambar 6.2 Tampilan antar muka yang diharapkan untuk halaman rekam aktivitas e-Performance



Gambar 6.3 Tampilan antar muka yang diharapkan pada halaman Pengesahan aplikasi e-Performance

6.1.2. Perhitungan Estimasi Biaya dengan Constructive Cost Model (COCOMO II)

Berdasarkan dokumen KAK yang diberikan oleh pihak ketiga, estimasi harga aplikasi didasarkan pada jumlah sumber daya yang dimiliki beserta jabatan yang dibutuhkan. Sedangkan untuk penelitian Tugas Akhir ini berfokus pada estimasi harga aplikasi berdasarkan fitur/ fungsi yang akan dibuat, oleh karena itu, peneliti menggunakan estimasi yang berfokus untuk mengetahui *effort* atau usaha kedalam biaya menggunakan metode COCOMO II.

Tahapan yang dilakukan peneliti untuk memulai perhitungan COCOMO II adalah melakukan perhitungan *sizing* dengan menggunakan UFP untuk kemudian dikonversi ke satuan SLOC. Untuk memudahkan mengetahui tipe *input*/masukan untuk UFP peneliti membuat beberapa *Physical Diagram*

Flow Data untuk mengetahui aliran data yang memungkinkan digunakan untuk setiap fitur pada aplikasi *e-Performance*. Pendokumentasian DFD yang dimaksud bisa lebih detail berada pada Lampiran A.

Berdasarkan DFD yang dibuat, aliran data yang ada pada aplikasi tersebut adalah sebagai berikut :

1. Untuk dapat memulai menggunakan aplikasi, pengguna akan memasukkan informasi yang dibutuhkan berupa akun yang memiliki username dan password. Untuk beberapa tingkatan jabatan, tidak memiliki username dan password dapat langsung mengakses aplikasi ini.
2. Aktivitas berikutnya hanya ada pada jenis akses admin SKPD yaitu menambahkan pegawai, atau memperbaiki data pegawai jika diperlukan. Jika ingin menambah pegawai, admin SKPD akan memasukkan informasi terkait pegawai seperti NIP, Nama, jabatan, dan lain sebagainya. Memperbarui data pegawai biasanya dilakukan ketika seorang pegawai tersebut bersetatus sudah tidak bekerja/ pensiun atau dengan alasan lain.
3. Setelah berhasil masuk, untuk pegawai negeri dengan jabatan tertentu (Pejabat Penilai) dapat melakukan pemberian tugas untuk pegawai dengan jabatan yang berada dibawahnya. Aliran data yang dibutuhkan untuk memberikan penugasan mengacu pada peraturan Pemerintah Daerah terkait, seperti SKP, Sasaran Program dan lainnya
4. Tugas yang diberikan Pejabat penilai adalah tugas yang harus dilaksanakan oleh Pejabat yang dinilai dan harus dibuktikan dengan mengunggah tugas yang telah mereka selesaikan dengan memasukkan tugas tersebut pada rekapan aktivitas. Data yang dimasukkan dapat berupa jenis tugas yang dilakukan, kemudian bukti fisik yang diunggah, dapat berbentuk dokumen, foto, atau sebagainya.
5. Pejabat penilai dapat melakukan verifikasi terhadap unggahan aktivitas yang telah diselesaikan oleh pejabat

yang dinilai. Jika sesuai dengan tugas yang diberikan, pejabat penilai dapat mengesahkan aktivitas pejabat yang dinilai, sebaliknya jika tidak sesuai petugas penilai akan membatalkan aktivitas tersebut.

6. Tentunya untuk mengetahui aktivitas mana saja yang telah selesai/ yang belum dapat dilihat oleh semua jenis pejabat. Informasi yang akan ditampilkan berupa rekap penugasan dengan berbagai status seperti aktivitas telah disahkan, digagalkan, atau menunggu untuk disahkan.
7. Untuk hak akses Admin SKPD dapat mengetahui detail perangkat daerah berisi informasi mengenai berbagai perangkat daerah, alamat serta informasi yang dibutuhkan lainnya
8. Admin SKPD Selanjutnya juga dapat mengetahui informasi terkait indikator penilaian yang berisi tujuan sasaran program, dan kegiatan serta capaian kegiatan masing masing pegawai, untuk memonitoring kinerja baik dari pejabat yang dinilai maupun pejabat penilai.
9. Setiap beberapa waktu tertentu, semua pegawai daerah melakukan penilaian. Penilaian yang dilakukan terdiri dari beberapa aspek seperti aspek kuantitas, efektivitas waktu, kualitas, efesiensi biaya. Penilaian juga dilakukan untuk kompetensi dari tiap pegawai diantaranya penilaian integritas, tanggung jawab, komitmen, kerjasama kreativitas dan lain lain
10. Setelah melakukan penilaian, semua pejabat akan mendapatkan laporan hasil kinerja selama periode waktu tertentu dalam bentuk rapor baik rapor individu tiap pegawai atau rapor untuk satuan SKPD yang digenerate oleh admin SKPD

Langkah 1. Menentukann Model COCOMO II

Langkah Selanjutnya telah dijelaskan pada BAB 4 Perencanaan untuk perhitungan estimasi biaya aplikasi menggunakan COCOMO II dengan jenis *Early Design Model*

Langkah 2. Menghitung *variable Size* dengan menggunakan *Unjusted Function Point*

Dari DFD yang telah dibuat, kemudian dikategorikan berdasarkan masing masing komponen/ tipe inputan dari 5 hal berikut untuk setiap fungsi.

Tabel 6.1 Hasil rekap tipe masukan setiap fitur

| Fitur/ Fungsi | Tipe Masukan | | | | |
|---|--------------|-----|----|----|----|
| | ILF | EIF | EI | EO | EQ |
| Melakukan update data pegawai | 1 | - | 3 | 1 | 1 |
| Memasukkan aktivitas | 1 | - | 3 | 1 | 1 |
| Melakukan persetujuan aktivitas | 1 | - | 2 | 2 | 1 |
| Melakukan pembatalan aktivitas | 1 | - | 2 | 2 | 1 |
| Menampilkan capaian aktivitas | 1 | - | - | 1 | - |
| Menampilkan perangkat daerah | 1 | - | - | 1 | - |
| Menampilkan capaian kegiatan | - | 1 | - | 2 | - |
| Menampilkan indikator kerja | - | 1 | - | 2 | 1 |
| Menampilkan rekap pengu- gasan | 1 | - | 1 | 1 | 1 |
| Menampilkan sasaran program | - | 1 | - | 2 | 1 |
| Melakukan penilaian | 3 | - | 3 | 3 | 2 |
| Mengenerate Rapor | 1 | - | 2 | 1 | 1 |
| Melakukan Pencarian ber- dasarkan filer tertentu | 3 | - | 5 | 5 | 1 |
| Login | 1 | - | 2 | 1 | 1 |
| Ganti Password | 1 | - | 2 | 1 | 1 |

Kemudian setelah mengetahui jumlah setiap tipe masukan yang ada setiap fungsi, dilanjutkan untuk menghitung kom-

pleksitas masukan berdasarkan ERD yang telah dibuat. Untuk melihat detail ERD, dapat dilihat pada lampiran A-2.

Dibawah ini adalah tabel hasil rekap untuk mengetahui tipe kompleksitas masukan dengan menghitung jumlah elemen data dan *data record*

Tabel 6.2 Hasil penghitungan elemen data dan record data setiap fitur

| Fitur/ Fungsi | Tipe Masukan | | | | | | | | | |
|--|--------------|---------|-----|---------|-----|---------|-----|---------|-----|---------|
| | ILF | | EIF | | EI | | EO | | EQ | |
| | RET | DE T | RET | DE T | FTR | DE T | FTR | DE T | FTR | DE T |
| Melakukan update data pegawai | 1 | 37 | - | - | 3 | 20 | 1 | 20 | 1 | 20 |
| Memasukkan aktivitas | 1 | 14 | - | - | 3 | 14 | 1 | 14 | 1 | 14 |
| Melakukan persetujuan aktivitas | 1 | 14 | - | - | 3 | 14 | 2 | 14 | 1 | 14 |
| Melakukan pembatalan aktivitas | 1 | 14 | - | - | 2 | 14 | 2 | 14 | 1 | 14 |
| Menampilkan capaian aktivitas | | 14 | - | - | - | - | 1 | 24 | - | - |
| Menampilkan perangkat daerah | 1 | 20 | - | - | - | - | 1 | 20 | - | - |
| Menampilkan capaian kegiatan | - | - | 1 | 14 | - | - | 2 | 14 | - | - |
| Menampilkan indikator kerja | - | - | 1 | 4 | - | - | 2 | 4 | 1 | 4 |
| Menampilkan rekap penugasan | - | 22 | - | - | 1 | 22 | 1 | 22 | 1 | 22 |
| Menampilkan sasaran program | - | - | 1 | 33 | - | - | 2 | 33 | 1 | 33 |
| Melakukan penilaian | - | 27 | - | - | 3 | 27 | 3 | 27 | 2 | 27 |
| Mengenerate Rapor | - | 29 | - | - | 2 | 29 | 1 | 29 | 1 | 29 |
| Melakukan Pencarian berdasarkan filer tertentu | - | 81 | - | - | 5 | 81 | 5 | 81 | 1 | 81 |

| Tipe Masukan | | | | | | | | | | |
|----------------|-----|---------|-----|---------|-----|---------|-----|---------|-----|---------|
| Fitur/ Fungsi | ILF | | EIF | | EI | | EO | | EQ | |
| | RET | DE T | RET | DE T | FTR | DE T | FTR | DE T | FTR | DE T |
| Login | - | 4 | - | - | 1 | 4 | 1 | 4 | 1 | 4 |
| Ganti Password | - | 4 | - | - | 1 | 4 | 1 | 4 | 1 | 21 |

Setelah mengetahui jumlah dari data elemen setiap fitur, Selanjutnya setiap jenis kompleksitas akan dikalikan dengan bobot yang telah dijelaskan pada BAB 4 untuk mengetahui total UFP dari usulan aplikasi pihak ketiga

Tabel 6.3 Hasil perhitungan bobot kompleksitas aplikasi

| Tipe Fungsi | Bobot Kompleksitas | | | | | | | | |
|--------------------------|--------------------|-------|-------|---------|-------|-------|---------|-------|------------|
| | Simple | | | Average | | | Complex | | |
| | Count | Bobot | Point | Count | Bobot | Point | Count | Bobot | Point |
| External Inputs | 2 | 3 | 6 | 4 | 4 | 16 | 5 | 6 | 30 |
| External Output | 4 | 4 | 16 | 9 | 5 | 45 | 3 | 7 | 21 |
| Internal Logical File | 10 | 7 | 70 | 2 | 10 | 20 | 1 | 15 | 15 |
| External Interface Files | 3 | 5 | 15 | 0 | 7 | 0 | 0 | 10 | 0 |
| External Inquiry | 7 | 3 | 21 | 4 | 4 | 16 | 2 | 6 | 12 |
| TOTAL | | | | | | | | | 284 |

Langkah 3. Mengubah satuan UFP ke SLOC

Selanjutnya setelah mengetahui hasil UFP, yang dilakukan berikutnya adalah mengkonversi satuan UFP menjadi SLOC dengan cara : $UFP \times SLOC \text{ Conversation Rate}$. Karena penggunaan bahasa pemrograman pada aplikasi yang diusulkan oleh pihak pengembang adalah PHP maka conversa-

tion rate yang digunakan sebesar 53 (dengan asumsi bahwa PHP memiliki kemiripan struktur dengan JAVA)[47].

$$\text{SLOC} = \text{UFP} \times \text{Conversation Rate}$$

Hasil :

$$\text{SLOC} = 284 \times 53 = \mathbf{15052}$$

Karena satuan yang digunakan oleh *variable size* adalah KSLOC, maka nilai SLOC harus lebih dulu dirubah menjadi KSLOC dengan membagi nilai dengan 1000.

Hasil :

$$\text{KSLOC} = 15052 / 1000 = \mathbf{15,052}$$

Langkah 4. Menghitung Usaha / Effor (*Person Month*)

Langkah perhitungan Selanjutnya adalah menghitung usaha yang dinyatakan dalam *person – month* dengan formula yaitu :

$$\text{PM} (\text{Effort}) = 2,94 \times \text{Size}^E \times M \quad (6.1)$$

Untuk nilai E yang didapat dari persamaan dibawah ini dengan melibatkan jumlah nilai pada *scale factors*.

$$E = B + (0,01 * \sum_{j=1}^5 SF_j) \quad (6.2)$$

$$\begin{aligned} E &= 0,91 + (0,01 * 19,44) \\ &= \mathbf{1,104} \end{aligned}$$

Kemudian menghitung formula utama dalam menghitung *effort / Person Month*

$$\begin{aligned} \text{PM} &= 2,94 \times 15,052^{1,104} \times 0,56 \\ &= \mathbf{32,9} \end{aligned}$$

Setelah didapat nilai *effort*, lama pengerjaan proyek dapat diketahui dengan persamaan berikut :

$$\text{Duration} = 3 \times PM^{\frac{1}{3}} \quad (6.3)$$

$$\begin{aligned} \text{Durasi} &= 3 \times 32,9^{1/3} \\ &= 9,61 \text{ bulan} \end{aligned}$$

Dengan total durasi selama 9,90 bulan untuk pengerjaan pengembangan aplikasi tersebut, dapat diketahui durasi untuk setiap aktivitas pengembangan dengan mendistribusikan durasi ke setiap prosentase aktivitas.

Selanjutnya untuk mengetahui berapa orang yang terlibat dalam pengembangan aplikasi tersebut, COCOMO II memiliki persamaan perhitungan sebagai berikut :

$$\text{Average Staff} = \text{Effort} / \text{TDEV} \quad (6.4)$$

Persamaan TDEV didapat dari formulasi perhitungan sebagai berikut :

$$\text{TDEV} = [C \times (\text{Effort})^{D + 0,2 \times (E - B)}] \quad (6.5)$$

Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 32,9^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 11,5 \text{ bulan} \end{aligned}$$

Dengan diketahuinya hasil perhitungan TDEV, rata rata pegawai yaitu :

$$\begin{aligned} \text{Average Staff} &= 32,9 / 11,5 \\ &= 3 \text{ staff.} \end{aligned}$$

Dari persamaan tersebut, perhitungan pegawai/ staff yang diperlukan selama pelaksanaan proyek sebanyak 3 orang. Dengan asumsi bahwa setiap aktivitas pada pengembangan tersebut dikerjakan oleh 3 orang.

Langkah 5. Melakukan Estimasi Biaya Total

Untuk mengetahui biaya total yang diperlukan dalam pengembangan aplikasi, diperlukan pembagian aktivitas dalam satuan prosentase berdasarkan total usaha, untuk kemudian akan dil-

akukan perhitungan upah/ biaya yang dikeluarkan untuk setiap orang yang terlibat dalam aktivitas proyek.

Berikut ini adalah hasil pembagian prosentase distribusi aktivitas untuk pengembangan proyek aplikasi *e-Performance* :

Tabel 6.4 Distribusi prosentase aktivitas

| Software Development Phases | Effort; (%*pm) | |
|-------------------------------|----------------|-------------|
| Aktivitas (%) | | |
| Requirements | 1,6 | 0,53 |
| Specifications | 7,5 | 2,47 |
| Design | 6 | 1,97 |
| Implementation | 52 | 17,10 |
| Acceptance & deployment | 5,5 | 1,81 |
| Ongoing life-cycle activities | | |
| Project management | 3,8 | 1,25 |
| Configuration management | 4,3 | 1,41 |
| Documentation | 8,4 | 2,76 |
| Training and support | 1 | 0,33 |
| Quality and testing phases | | |
| Integration testing | 7 | 2,30 |
| Quality assurance | 0,9 | 0,30 |
| Evaluation and Testing | 2 | 0,66 |
| Total | 100 | 32,9 |

Hasil pen-distribusian aktivitas dijadikan sebagai dasar untuk membagi lama durasi proyek yang dijadikan sebagai pengali dengan standar gaji yang ada disetiap aktivitas proyek. Namun dalam beberapa aktivitas pengembangan proyek tersebut tidak ditemukan pada pihak ketiga, maka tidak dilaksanakan dan person/ staff yang dialokasikan tidak ada.

Tabel 6.5 Hasil perhitungan biaya total pengerjaan proyek pengembangan aplikasi e-Performance

| Aktivitas | Jabatan | per- son | Mon th | Standar gaji (bu- lan) | Total |
|--|--------------------|---------------------|-------------------|---------------------------------------|--------------------------|
| <i>Software Development</i> | | | | | |
| <i>Require- ment</i> | Analisis Sistem | 3 | 0,15 | Rp7.000.00 0,00 | Rp 3.229.598,1 7 |
| <i>Specifica- tion</i> | Analisis Sistem | 3 | 0,72 | Rp7.000.00 0,00 | Rp 15.138.741, 41 |
| <i>Design</i> | Analisis Sistem | 3 | 0,58 | Rp7.000.00 0,00 | Rp 12.110.993, 13 |
| <i>Implemen- tation</i> | Pro- grammer | 3 | 5,00 | Rp8.000.00 0,00 | Rp119.956.5 03,35 |
| <i>Ac- ceptance & De- ployment</i> | Teknisi | 3 | 0,53 | Rp8.000.00 0,00 | Rp 12.687.707, 09 |
| <i>On Going life-cylce</i> | | | | | |
| <i>Project manage- ment</i> | Project Manager | 3 | 0,37 | Rp20.000.0 00,00 | Rp 21.915.130, 42 |
| <i>Documen- tation</i> | Adminis- trator | 3 | 0,81 | Rp15.000.0 00,00 | Rp 4.844.397,2 5 |
| <i>Training and sup- port</i> | Teknisi | 3 | 0,10 | Rp15.000.0 00,00 | Rp 4.325.354,6 9 |
| <i>Quality and testing</i> | | | | | |
| <i>Evaluation and Test- ing</i> | Tester | 3 | 0,19 | Rp8.000.00 0,00 | Rp 4.613.711,6 7 |
| TOTAL | | | | | Rp 198.822.137 ,17 |

Dari hasil perhitungan estimasi menggunakan metode CO-COMO II untuk proyek pembangunan aplikasi *e-Performance* yang dilakukan dalam penelitian ini adalah sebesar **Rp 198.822.137,17** dengan beberapa pertimbangan yaitu :

1. Rata – rata pekerja yang dimasukkan mengikuti formula yang ada pada COCOMO II yaitu 3 orang untuk seluruh aktivitas proyek
2. Adanya beberapa aktivitas yang tidak dihitung/ tidak dialokasikan pegawai karena menyesuaikan aktivitas pada KAK pihak ketiga

6.1.3. Perbandingan Estimasi Biaya dengan Penyedia Jasa

Perbandingan estimasi biaya dengan penyedia jasa digunakan untuk menilai apakah estimasi yang telah dikeluarkan oleh pihak ketiga termasuk dalam kategori mahal/ tidak. Peneliti mencari dan melakukan perbandingan dari beberapa penyedia jasa sejenis, kemudian memberikan pertanyaan terkait biaya pembangunan aplikasi sesuai dengan usulan pihak ketiga sesuai studi kasus yang digunakan.

Berikut ini adalah fungsi pembandingan yang ditawarkan oleh pihak lain :

Tabel 6.6 Fungsi pembandingan yang ditawarkan pihak lain

| No | Fungsi Pembanding | Aplikasi A | Aplikasi B |
|----|---|------------|------------|
| 1 | Aplikasi dapat menampilkan informasi perangkat daerah | V | V |
| 2 | Aplikasi dapat meng- <i>input</i> data pegawai sesuai tingkat jabatan | V | V |
| 3 | Aplikasi dapat melakukan <i>update</i> data pegawai | V | V |
| 4 | Aplikasi dapat memberikan evaluasi berupa kuis dalam bentuk | V | V |

| No | Fungsi Pemanding | Aplikasi A | Aplikasi B |
|-------|--|------------|------------|
| | form soal untuk evaluasi pegawai pada kurun waktu tertentu | | |
| 5. | Aplikasi dapat menampilkan indikator yang berhasil dicapai untuk masing masing pegawai | V | V |
| 6. | Aplikasi yang dapat menampilkan informasi tugas yang dilakukan setiap pegawai | V | V |
| 7. | Aplikasi yang dapat menginputkan aktivitas pegawai | V | V |
| 8. | Aplikasi yang dapat menampilkan capaian kegiatan | V | V |
| 9. | Aplikasi yang dapat menampilkan status aktivitas perangkat daerah | V | V |
| 10. | Aplikasi yang dapat menampilkan nilai akhir tiap pegawai daerah | V | V |
| Total | | 111 JT | 120 JT |

a. Aplikasi A

Aplikasi *e-Performance* ini ditawarkan oleh Konsultan IT PT. TATI. Aplikasi tersebut dirancang untuk pemerintah daerah Madiun dan dapat diakses di berbagai *device (multi device)* . Pengembangan aplikasi yang ditawarkan oleh perusahaan tersebut sebesar Rp 111.000.000 dengan detail sebagai berikut :

1. Desain

- Desain disediakan dari penyedia berupa tampilan untuk aplikasi website, beserta tampilan android
2. Menyediakan layanan absen dengan fasilitas GPS untuk melacak PNS untuk acara diluar kantor
 3. Aplikasi dalam bentuk website
 4. Multi Device
Dapat diakses melalui *desktop* dan android
 5. Memberikan tambahan dukungan berupa 3 bulan *maintenance* gratis, *update* fitur , *security* profesional, menyediakan dukungan pelatihan.
 6. Durasi pengerjaan pengembangan aplikasi membutuhkan estimasi waktu sekitar 5-6 bulan

b. Aplikasi B

Aplikasi ini ditujukan untuk penilaian kinerja Kota Surabaya, dikembangkan oleh Departemen IT Surabaya. Pengembangan aplikasi ini menghabiskan biaya berkisar 80- 120 juta . aplikasi ini telah terintegrasi dengan aplikasi pemerintah daerah yaitu SI Monev untuk mengambil data capaian kegiatan dan yang berhubungan dengan aktivitas yang dilakukan oleh pegawai PNS. Dalam aplikasi tersebut memiliki detail diantaranya :

1. Desain
Tampilan antarmuka dapat menyesuaikan *client* dan menerima berbagai update untuk tampilan antarmuka
2. Aplikasi dalam bentuk website
3. Memberikan tambahan dukungan berupa *maintenance*, *update* optimasi setiap bulan garansi 100%
4. Lama pengerjaan proyek sekitar 3-4 bulan belum termasuk *maintenance* dan *update*

Berdasarkan fitur yang ditawarkan oleh pihak ketiga dan informasi yang didapat oleh peneliti dari hasil observasi dan *benchmark* ke penyedia lain, dilakukan dokumentasi seperti pada Tabel 6.7 berikut :

Tabel 6.7 Hasil perbandingan fitur dan estimasi biaya terhadap penyedia jasa lain

| Teknik / penyedia jasa lain | Waktu pengerjaan (Bulan) | Estimasi Biaya |
|--|--------------------------|------------------|
| COCOMO II | 10 | Rp198.822.137,00 |
| Pihak ketiga | 6 | Rp220.000.000,00 |
| PT Tata Cipta Teknologi Informasi (TATI) | 6 | Rp111.000.000,00 |
| Dinas Kominfo Surabaya | 3 | Rp120.000.000,00 |

Jika dibandingkan dengan estimasi harga yang ditawarkan oleh pihak ketiga sebesar Rp220.000.000,00 (belum termasuk *maintenance*) dapat disimpulkan bahwa pembangunan aplikasi oleh pihak ketiga **lebih mahal** dibanding dengan perhitungan estimasi COCOMO II ataupun dengan penyedia jasa lain. Hasil tersebut dapat dijadikan acuan dalam pengambilan studi kasus pengembangan *e-Performance* untuk kabupaten XYZ dengan menggunakan rekayasa nilai (*Value Engineering*) yang berfokus untuk mengoptimalkan dalam hal ini penghematan biaya pengembangan aplikasi.

6.2. Function Phase Analysis

Pada fase ini, pembuatan diagram F.A.S.T tidak langsung dilakukan, pembuatan diagram hanya didasarkan pada fitur/kebutuhan fungsional yang membutuhkan biaya paling banyak. Untuk mengetahui apakah setiap kebutuhan fungsional dapat dibuat ke dalam diagram F.A.S.T, peneliti menggunakan metode perhitungan bobot. Hasil dari diagram F.A.S.T tersebut nantinya akan menjadi dasar pada pengolahan komponen lain pada tahap berikutnya.

6.2.1. Perhitungan Biaya Kebutuhan Fungsional

Komponen yang dibutuhkan untuk membangun diagram F.A.S.T adalah pendefinisian kebutuhan fungsional aplikasi.

Berikut ini adalah kebutuhan yang dibutuhkan berdasarkan dari usulan pihak ketiga :

1. Update data pegawai, memerlukan tipe masukan :
 - a. EI : Form input data pegawai (H)
 - b. EO : Laporan keberhasilan *update* (H)
 - c. ILF : Database Pegawai (A)
 - d. EIF : -
 - e. EQ : Rekap *update* data pegawai
2. *Input* aktivitas untuk semua pegawai memerlukan tipe masukan :
 - a. EI : Form *input* aktivitas (H)
 - b. EO : Informasi detail aktivitas (L)
 - c. ILF : Database Aktivitas, Pegawai (L)
 - d. EIF : -
 - e. EQ : Melihat aktivitas pegawai (L)
3. Aplikasi mampu melakukan persetujuan memerlukan tipe masukan :
 - a. EI : Form verifikasi aktivitas (A)
 - b. EO : Informasi persetujuan aktivitas(A)
 - c. ILF : Database Pegawai, aktivitas (L)
 - d. EIF : -
 - e. EQ : Melihat aktivitas pegawai yang telah disetujui (L)
4. Aplikasi mampu melakukan pembatalan aktivitas memerlukan tipe masukan :
 - a. EI : Form verifikasi aktivitas (A)
 - b. EO : Informasi pembatalan aktivitas (A)
 - c. ILF : Database Pegawai, Aktivitas (L)
 - d. EIF : -
 - e. EQ : Melihat aktivitas pegawai yang berhasil disetujui (L)
5. Aplikasi dapat menampilkan capaian aktivitas pegawai PNS, memerlukan tipe masukan :
 - a. EI : -
 - b. EO : Informasi capaian aktivitas (A)

- c. ILF : Database Aktivitas, Pegawai (L)
 - d. EIF :-
 - e. EQ :-
6. Aplikasi dapat menampilkan informasi perangkat daerah, memerlukan tipe masukan :
- a. EI :-
 - b. EO : Informasi perangkat daerah (A)
 - c. ILF : Database Aktivitas, Pegawai (L)
 - d. EIF :-
 - e. EQ :-
7. Aplikasi dapat menampilkan capaian realisasi kegiatan yang dimiliki pemerintah daerah, memerlukan tipe masukan :
- a. EI :-
 - b. EO : Informasi capaian kegiatan (A)
 - c. ILF :-
 - d. EIF : Database Kegiatan (L)
 - e. EQ :-
8. Aplikasi mampu menampilkan indikator penilaian untuk para pegawai, memerlukan tipe masukan :
- a. EI :
 - b. EO : Informasi indikator penilaian (L)
 - c. ILF :-
 - d. EIF : Database Indikator (L)
 - e. EQ : Melihat indikator bobot penilaian (L)
9. Aplikasi dapat menampilkan pemberian tugas untuk pejabat yang dinilai dari pejabat penilai sesuai dengan SKP pemerintah daerah setempat, memerlukan tipe masukan :
- a. EI :-
 - b. EO : Informasi detail Penugasan
 - c. ILF : Database Pegawai, Aktivitas (L)
 - d. EIF :-
 - e. EQ : Melihat penugasan (A)

10. Aplikasi yang dapat menampilkan sasaran Program untuk tiap kegiatan/ aktivitas baik dari bobot serta aspek aspek yang dinilai, memerlukan tipe masukan :
 - a. EI : -
 - b. EO : Informasi detail sasaran program (H)
 - c. ILF : -
 - d. EIF : Sasaran Program (L)
 - e. EQ : Melihat sasaran program (A)
11. Aplikasi mampu mengadakan penilaian untuk setiap pegawai, memerlukan tipe masukan :
 - a. EI : Form soal, aspek penilaian (H)
 - b. EO : Pop up pemberitahuan, *act-button* (H)
 - c. ILF : Database Soal, Aspek, Penilaian (A)
 - d. EIF : -
 - e. EQ : Melihat soal penialain (H)
12. Aplikasi dapat membantu melakukan pencarian sesuai beberapa kriteria yang diperlukan baik atkvitias, pegawai, ataupun tugas, memerlukan tipe masukan :
 - a. EI : Masukan tipe pencarian, kriteria pencarian (*Sort by*) (H)
 - b. EO : Informasi detail pencarian (H)
 - c. ILF : Database Pegawai, Aktivitas, Perangkat Daerah, Penugasan (H)
 - d. EIF : -
 - e. EQ : Filter Pencarian (H)
13. Login memerlukan tipe masukan :
 - a. EI : Masukan akun (H)
 - b. EO : Beranda *e-Performance* (H)
 - c. ILF : Database Akun (L)
 - d. EIF : -
 - e. EQ : Profil pegawai(L)
14. Ganti password memerlukan tipe masukan :
 - a. EI : Masukan password baru (H)
 - b. EO : Pop up pemberitahuan (H)
 - c. ILF : Database akun (L)

- d. EIF : -
- e. EQ : Profil pegawai (L)
Untuk mengetahui estimasi fungsi yang dibutuhkan peneliti menggunakan metode COCOMO II. Kemudian peneliti menghitung tingkat kompleksitas tipe masukan hingga menghasilkan UFP seperti pada Tabel 6.8 :

Tabel 6.8 Hasil perhitungan UFP untuk setiap kebutuhan fungsional

| Fitur/ Fungsi | Tipe Masukan | | | | | | | | | | | | | | | TOTAL UFP | |
|--|--------------|---|---|-----|---|---|----|---|---|----|---|---|----|---|---|-----------|----|
| | ILF | | | EIF | | | EI | | | EO | | | EQ | | | | |
| | L | A | H | L | A | H | L | A | H | L | A | H | L | A | H | | |
| Melakukan update data pegawai | | 1 | | | | | | | 1 | | 1 | | | 1 | | | 25 |
| Memasukkan aktivitas | 1 | | | | | | | | 1 | 1 | | | 1 | | | | 20 |
| Melakukan persetujuan aktivitas | 1 | | | | | | | 1 | | | 1 | | 1 | | | | 19 |
| Melakukan pembatalan aktivitas | 1 | | | | | | | 1 | | | 1 | | 1 | | | | 19 |
| Menampilkan capaian aktivitas | 1 | | | | | | | | | | 1 | | | | | | 12 |
| Menampilkan perangkat daerah | 1 | | | | | | | | | | 1 | | | | | | 12 |
| Menampilkan capaian kegiatan | | | | 1 | | | | | | | 1 | | | | | | 10 |
| Menampilkan indikator kerja | | | | 1 | | | | | | 1 | | | 1 | | | | 12 |
| Menampilkan rekap penugasan | 1 | | | | | | | 1 | | | 1 | | | 1 | | | 20 |
| Menampilkan sasaran program | | | | 1 | | | | | | | | 1 | | 1 | | | 16 |
| Melakukan penilaian | | 1 | | | | | | | 1 | | | 1 | | | 1 | | 29 |
| Mengenerate Rapor | 1 | | | | | | | | 1 | | 1 | | | 1 | | | 22 |
| Melakukan Pencarian berdasarkan filer tertentu | 1 | | | | | | | | 1 | | | 1 | | 1 | | | 34 |
| Login | 1 | | | | | | 1 | | | 1 | | | 1 | | | | 17 |
| Ganti Password | 1 | | | | | | 1 | | | 1 | | | 1 | | | | 17 |

Hasil pada kolom TOTAL UFP telah dikalikan dengan bobot dari kompleksitas masukan (*skala Low- High*) untuk tiap kebutuhan fungsionalitas.

Berdasarkan perhitungan UFP beserta *cost driver* (*Scale Factors* dan *Effort Multiplier*), Selanjutnya dilakukan perhitungan effort COCOMO II dengan menggunakan rumus yang sama seperti yang dijelaskan pada dasar teori. Nilai *scale factor* dan *Effort Multiplier* yang digunakan berasal dari perhitungan estimasi usaha keseluruhan aplikasi dan memiliki karakteristik yang sama untuk perhitungan setiap kebutuhan fungsional. Langkah berikutnya peneliti melakukan perhitungan COCOMO II dengan memasukkan setiap variabel seperti *scale factor* yang bernilai **19,94** dan untuk nilai *effort multiplier* sebesar **0,56**. Konversi yang digunakan untuk menentukann SLOC juga dikali dengan konversi nilai JAVA sebesar **53**.

Untuk alokasi orang/ *person* untuk seluruh fungsi dialokasikan sesuai hasil rata rata perhitungan staff, sedangkan untuk membagi alokasi waktu setiap fase berdasarkan hasil perkalian dari prosentase aktivitas untuk fase *software development* dengan total alokasi durasi yang dibutuhkan dalam pengembangan sebuah fitur.

6.2.1.1. Mengupdate data pegawai

Seperti perhitungan sebelumnya, perhitungan untuk fungsi update data pegawai adalah sebagai berikut :

1. UFP : 25
2. KSLOC : 1,325
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,325^{1,104} \times 0,56 \\ &= 4,4 \text{ Person-month} \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned} \text{Duration} &= 3 \times 4,4^{1/3} \\ &= 4,92 \text{ bulan} \\ &= 98.4 \text{ (hari)} \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 4,44^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 6 \text{ bulan (hasil pembulatan)} \end{aligned}$$

$$\begin{aligned} \text{Average Staff} &= \text{Effort} / \text{TDEV} \\ &= 4,92 / 6 \\ &= 1 \text{ (hasil pembulatan)} \end{aligned}$$

Tabel 6.9 Hasil perhitungan biaya untuk kebutuhan fungsional Update Pegawai

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-----------------------------|-----------------|---------|--------|--------------------------|-----------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,08 | Rp 7.000.000,00 | Rp 551.040,00 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,37 | Rp 7.000.000,00 | Rp 2.583.000,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,30 | Rp 7.000.000,00 | Rp 2.066.400,00 |
| <i>Implementat</i> | Program | 1 | 2,56 | Rp | Rp |

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|------------------------------------|---------|---------|--------|--------------------------|---------------------|
| <i>ion</i> | mer | | | 8.000.000,00 | 20.467.200,00 |
| <i>Acceptance & deployment</i> | Tehnisi | 1 | 0,27 | Rp 8.000.000,00 | Rp 2.164.800,00 |
| TOTAL | | | | | Rp27.832.440 |

Total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional telah dilakukan seperti yang terdapat pada Tabel 6.9

6.2.1.2. Menginputkan Aktivitas

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 20
2. KSLOC : 1,060
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,060^{1,104} \times 0,56 \\ &= 3,5 \text{ Person-month} \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned} \text{Duration} &= 3 \times 3,5^{1/3} \\ &= 4,55 \text{ bulan} \\ &= 91 \text{ hari} \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\text{TDEV} = 3,67 \times 3,5^{0,28 + 0,2 \times (1,104 - 0,91)}$$

$$= 5 \text{ bulan (hasil pembulatan)}$$

$$\text{Average Staff} = \text{Effort} / \text{TDEV}$$

$$= 3.5 / 5$$

$$= 1 \text{ (hasil pembulatan)}$$

Berikut adalah total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada Tabel 6.10

Tabel 6.10 Hasil perhitungan biaya untuk kebutuhan fungsional Input Aktivitas

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-----------------------------|-----------------|---------|--------|--------------------------|-----------------|
| Software Development | | | | | |
| <i>Requireme nts</i> | Analisis Sistem | 1 | 0,07 | Rp 7.000.000,00 | Rp509.600,00 |
| <i>Specificatio ns</i> | Analisis Sistem | 1 | 0,34 | Rp 7.000.000,00 | Rp2.388.750,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,27 | Rp 7.000.000,00 | Rp1.911.000,00 |
| <i>Implementa tion</i> | Program mer | 1 | 2,37 | Rp 8.000.000,00 | Rp18.928.000,00 |

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|------------------------------------|---------|---------|--------|--------------------------|---------------------|
| <i>Acceptance & deployment</i> | Tehnici | 1 | 0,25 | Rp 8.000.000,00 | Rp2.002.000,00 |
| TOTAL | | | | | Rp25.739.350 |

6.2.1.3. Melakukan Persetujuan Aktivitas

Seperti perhitungan sebelumnya, perhitungan untuk fungsi seperti menginputkan aktivitas adalah sebagai berikut :

1. UFP : 19
2. KSLOC : 1,007
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 1,007^{1,104} \times 0,56 \\
 &= 3 \quad \text{Person Month}
 \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned}
 \text{Durasi} &= 3 \times 3^{1/3} \\
 &= 4,33 \quad \text{Bulan} \\
 &= 86.6 \quad \text{Hari}
 \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata-rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 3^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 5 \text{ bulan (hasil pembulatan)} \end{aligned}$$

$$\begin{aligned} \text{Average Staff} &= \text{Effort} / \text{TDEV} \\ &= 5 / 6 \\ &= 1 \text{ (hasil pembulatan)} \end{aligned}$$

Berikut adalah total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.11**

Tabel 6.11 Hasil perhitungan biaya untuk kebutuhan fungsional Melakukan persetujuan aktivitas

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|------------------------------------|-----------------|---------|--------|--------------------------|---------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,07 | Rp 7.000.000,00 | Rp 484.960,00 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,32 | Rp 7.000.000,00 | Rp 2.273.250,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,26 | Rp 7.000.000,00 | Rp 1.818.600,00 |
| <i>Implementation</i> | Programmer | 1 | 2,25 | Rp 8.000.000,00 | Rp 18.012.800,00 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,24 | Rp 8.000.000,00 | Rp 1.905.200,00 |
| TOTAL | | | | | Rp24,494,810 |

6.2.1.4. Melakukan Pembatalan Aktivitas

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 19
2. KSLOC : 1,007
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,007^{1,104} \times 0,56 \\ &= 3 \quad \text{Person Month} \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned} \text{Durasi} &= 3 \times 3^{1/3} \\ &= 4,33 \quad \text{Bulan} \\ &= 86.6 \quad \text{Hari} \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 3^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 5 \text{ bulan (hasil pembulatan)} \\ \text{Average Staff} &= \text{Effort} / \text{TDEV} \\ &= 3/5 \\ &= 0,6 \end{aligned}$$

= 1 (hasil pembulatan)

Berikut adalah total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.12**

Tabel 6.12 Hasil perhitungan biaya untuk kebutuhan fungsional Pembatalan Aktivitas

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|------------------------------------|-----------------|---------|--------|--------------------------|---------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,07 | Rp 7.000.000,00 | Rp 484.960,00 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,32 | Rp 7.000.000,00 | Rp 2.273.250,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,26 | Rp 7.000.000,00 | Rp 1.818.600,00 |
| <i>Implementation</i> | Programmer | 1 | 2,25 | Rp 8.000.000,00 | Rp 18.012.800,00 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,24 | Rp 8.000.000,00 | Rp 1.905.200,00 |
| TOTAL | | | | | Rp24,494,810 |

6.2.1.5. Menampilkan informasi capaian Aktivitas

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 12

2. KSLOC : 0,636
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned} \text{Effort} &= 2,94 \times 0,636^{1,104} \times 0,56 \\ &= 1,8 \text{ Person Month} \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned} \text{Durasi} &= 3 \times 1,8^{1/3} \\ &= 3,65 \text{ Bulan} \\ &= 73 \text{ Hari} \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata-rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 1,8^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 4 \text{ bulan (hasil pembulatan)} \end{aligned}$$

$$\begin{aligned} \text{Average Staff} &= \text{Effort} / \text{TDEV} \\ &= 1,8 / 4 \\ &= 1 \text{ (hasil pembulatan)} \end{aligned}$$

**Tabel 6.13 Hasil perhitungan biaya untuk kebutuhan fungsional
Menampilkan Capaian Aktivitas**

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|------------------------------------|-----------------|---------|--------|--------------------------|---------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,06 | Rp 7.000.000,00 | Rp 408.800,00 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,27 | Rp 7.000.000,00 | Rp 1.916.250,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,22 | Rp 7.000.000,00 | Rp 1.533.000,00 |
| <i>Implementation</i> | Programmer | 1 | 1,90 | Rp 8.000.000,00 | Rp 15.184.000,00 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,20 | Rp 8.000.000,00 | Rp 1.606.000,00 |
| TOTAL | | | | | Rp20.648.050 |

Total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.13**

6.2.1.6. Menampilkan informasi Perangkat Daerah

Sebagai perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 12
2. KSLOC : 0,636
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned} \text{Effort} &= 2,94 \times 0,636^{1,104} \times 0,56 \\ &= 1,8 \text{ Person Month} \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned} \text{Durasi} &= 3 \times 1,8^{1/3} \\ &= 3,65 \text{ Bulan} \\ &= 73 \text{ Hari} \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata-rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 1,8^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 4 \text{ bulan (hasil pembulatan)} \end{aligned}$$

$$\begin{aligned} \text{Average Staff} &= \text{Effort} / \text{TDEV} \\ &= 1,8 / 4 \\ &= 1 \text{ (hasil pembulatan)} \end{aligned}$$

Berikut adalah total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.14**

Tabel 6.14 Hasil perhitungan biaya untuk kebutuhan fungsional Informasi Perangkat Daerah

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|------------------------------------|-----------------|---------|--------|--------------------------|---------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,06 | Rp 7.000.000,00 | Rp 408.800,00 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,27 | Rp 7.000.000,00 | Rp 1.916.250,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,22 | Rp 7.000.000,00 | Rp 1.533.000,00 |
| <i>Implementation</i> | Programmer | 1 | 1,90 | Rp 8.000.000,00 | Rp 15.184.000,00 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,20 | Rp 8.000.000,00 | Rp 1.606.000,00 |
| TOTAL | | | | | Rp20.648.050 |

6.2.1.7. Menampilkan capaian Kegiatan

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 10
2. KSLOC : 0,530
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned} \text{Effort} &= 2,94 \times 0,53^{1,104} \times 0,56 \\ &= 1,5 \text{ Person Month} \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned} \text{Durasi} &= 3 \times 1,5^{1/3} \\ &= 3,43 \text{ Bulan} \\ &= 68.6 \text{ Hari} \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 1,5^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 4 \text{ bulan (hasil pembulatan)} \end{aligned}$$

$$\begin{aligned} \text{Average Staff} &= \text{Effort} / \text{TDEV} \\ &= 1,5 / 4 \\ &= 1 \text{ (hasil pembulatan)} \end{aligned}$$

Berikut adalah total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada.

Tabel 6.15.

Tabel 6.15 Hasil perhitungan biaya untuk kebutuhan fungsional Capaian Kegiatan

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-----------------------------|---------|---------|--------|--------------------------|-------|
| Software Development | | | | | |

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|------------------------------------|-----------------|---------|--------|--------------------------|---------------------|
| <i>Requirements</i> | Analisis Sistem | 1 | 0,05 | Rp 7.000.000,00 | Rp 384.160,00 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,26 | Rp 7.000.000,00 | Rp 1.800.750,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,21 | Rp 7.000.000,00 | Rp 1.440.600,00 |
| <i>Implementation</i> | Programmer | 1 | 1,78 | Rp 8.000.000,00 | Rp 14.268.800,00 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,19 | Rp 8.000.000,00 | Rp 1.509.200,00 |
| TOTAL | | | | | Rp19.403.510 |

6.2.1.8. Menampilkan indikator

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 12
2. KSLOC : 0,636
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 0,636^{1,104} \times 0,56 \\
 &= 1,8 \text{ Person Month}
 \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned} \text{Durasi} &= 3 \times 1,8^{1/3} \\ &= 3,65 \text{ Bulan} \\ &= 73 \text{ Hari} \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata-rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 1,8^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 4 \text{ bulan (hasil pembulatan)} \end{aligned}$$

$$\begin{aligned} \text{Average Staff} &= \text{Effort} / \text{TDEV} \\ &= 1,8 / 4 \\ &= 1 \text{ (hasil pembulatan)} \end{aligned}$$

Berikut adalah total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.16**

Tabel 6.16 Hasil perhitungan biaya untuk kebutuhan fungsional Menampilkan Indikator

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-----------------------------|-----------------|---------|--------|--------------------------|---------------|
| <i>Software Development</i> | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,06 | Rp 7.000.000,00 | Rp 408.800,00 |

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-------------------------------------|-----------------|---------|--------|--------------------------|-------------------------|
| <i>Specifica-tions</i> | Analisis Sistem | 1 | 0,27 | Rp 7.000.000 ,00 | Rp 1.916.250, 00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,22 | Rp 7.000.000 ,00 | Rp 1.533.000, 00 |
| <i>Implemen-tation</i> | Pro-grammer | 1 | 1,90 | Rp 8.000.000 ,00 | Rp 15.184.000 ,00 |
| <i>Acceptance & deploy-ment</i> | Tehnisi | 1 | 0,20 | Rp 8.000.000 ,00 | Rp 1.606.000, 00 |
| TOTAL | | | | | Rp20.721.090 |

6.2.1.9. Menampilkan informasi Pemberian Tugas

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 20
2. KSLOC : 1,060
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 1,060^{1,104} \times 0,56 \\
 &= 3,1 \text{ Person Month}
 \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned}
 \text{Durasi} &= 3,1 \times 3,1^{1/3} \\
 &= 4,37 \text{ Bulan} \\
 &= 87,4 \text{ Hari}
 \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata-rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned}
 \text{TDEV} &= 3,67 \times 3,1^{0,28 + 0,2 \times (1,104 - 0,91)} \\
 &= 5 \text{ bulan (hasil pembulatan)}
 \end{aligned}$$

$$\begin{aligned}
 \text{Average Staff} &= \text{Effort} / \text{TDEV} \\
 &= 13,1 / 5 \\
 &= 1 \text{ (hasil pembulatan)}
 \end{aligned}$$

Berikut adalah total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.17**

Tabel 6.17 Hasil perhitungan biaya untuk kebutuhan fungsional Menampilkan Informasi Pemberian Tugas

| Kelompok aktivitas | Jabatan | Persn | Mont h | Standar Gaji (per-bulan) | Total |
|-----------------------------|-----------------|-------|--------|--------------------------|--------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,07 | Rp 7.000.000 | Rp 489.440 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,33 | Rp 7.000.000 | Rp 2.294.250 |
| <i>Design</i> | Analisis | 1 | 0,26 | Rp | Rp |

| Kelompok aktivitas | Jabatan | Persn | Mont h | Standar Gaji (per-bulan) | Total |
|------------------------------------|------------|-------|--------|--------------------------|---------------------|
| | Sistem | | | 7.000.000 | 1.835.400 |
| <i>Implementation</i> | Programmer | 1 | 2,27 | Rp 8.000.000 | Rp 18.179.200 |
| <i>Acceptance & deployment</i> | Tehnici | 1 | 0,24 | Rp 8.000.000 | Rp 1.922.800 |
| TOTAL | | | | | Rp24.721.090 |

6.2.1.10. Menampilkan Sasaran Program

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 16
2. KSLOC : 0,848
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 0,848^{1,104} \times 0,56 \\
 &= 2,5 \text{ Person Month}
 \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned}
 \text{Durasi} &= 3 \times 2,5^{1/3} \\
 &= 4,07 \text{ Bulan} \\
 &= 81.4 \text{ Hari}
 \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 2,5^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 5 \text{ bulan (hasil pembulatan)} \end{aligned}$$

$$\begin{aligned} \text{Average Staff} &= \text{Effort} / \text{TDEV} \\ &= 2,5 / 5 \\ &= 1 \text{ (hasil pembulatan)} \end{aligned}$$

Berikut adalah total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.18**

Tabel 6.18 Hasil perhitungan biaya untuk kebutuhan fungsional Menampilkan Sasaran Program

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-----------------------------|-----------------|---------|--------|--------------------------|------------------|
| Software Development | | | | | |
| <i>Require-ments</i> | Analisis Sistem | 1 | 0,07 | Rp 7.000.000,00 | Rp 455.840,00 |
| <i>Specifica-tions</i> | Analisis Sistem | 1 | 0,31 | Rp 7.000.000,00 | Rp 2.136.750,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,24 | Rp 7.000.000,00 | Rp 1.709.400,00 |
| <i>Implemen-tation</i> | Pro-grammer | 1 | 2,12 | Rp 8.000.000,00 | Rp 16.931.200,00 |

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|------------------------------------|---------|---------|--------|--------------------------|---------------------|
| <i>Acceptance & deployment</i> | Tehnici | 1 | 0,22 | Rp 8.000.000,00 | Rp 1.790.800,00 |
| TOTAL | | | | | Rp23.023.990 |

6.2.1.11. Melakukan Penilaian

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 29
2. KSLOC : 1,537
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 1,537^{1,104} \times 0,56 \\
 &= 4,7 \text{ Person Month}
 \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned}
 \text{Durasi} &= 3 \times 4,7^{1/3} \\
 &= 5,03 \text{ Bulan} \\
 &= 100.6 \text{ Hari}
 \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata rata

yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\text{TDEV} = 3,67 \times 4,7^{0,28 + 0,2 \times (1,104 - 0,91)}$$

$$= 6 \text{ bulan (hasil pembulatan)}$$

$$\text{Average Staff} = \text{Effort} / \text{TDEV}$$

$$= 4,7 / 6$$

$$= 1 \text{ (hasil pembulatan)}$$

Tabel 6.19 Hasil perhitungan biaya untuk kebutuhan fungsional Penilaian

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-------------------------------------|-----------------|---------|--------|--------------------------|---------------------|
| Software Development | | | | | |
| <i>Require-ments</i> | Analisis Sistem | 1 | 0,08 | Rp 7.000.000,00 | Rp 563.360,00 |
| <i>Specifica-tions</i> | Analisis Sistem | 1 | 0,38 | Rp 7.000.000,00 | Rp 2.640.750,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,30 | Rp 7.000.000,00 | Rp 2.112.600,00 |
| <i>Implemen-tation</i> | Pro-grammer | 1 | 2,62 | Rp 8.000.000,00 | Rp 20.924.800,00 |
| <i>Acceptance & deploy-ment</i> | Tehnikisi | 1 | 0,28 | Rp 8.000.000,00 | Rp 2.213.200,00 |
| TOTAL | | | | | Rp28.454.710 |

Total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.19**

6.2.1.12. Generate Rapor

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 22
2. KSLOC : 1,166
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,166^{1,104} \times 0,56 \\ &= 3,5 \text{ Person Month} \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned} \text{Durasi} &= 3 \times 3,5^{1/3} \\ &= 4,55 \text{ Bulan} \\ &= 91 \text{ Hari} \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 3,5^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 5 \text{ bulan (hasil pembulatan)} \end{aligned}$$

$$\begin{aligned}
 \text{Average Staff} &= \text{Effort} / \text{TDEV} \\
 &= 3,5 / 5 \\
 &= 1 \text{ (hasil pembulatan)}
 \end{aligned}$$

Tabel 6.20 Hasil perhitungan biaya untuk kebutuhan fungsional Generate Rapor

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|------------------------------------|-----------------|---------|--------|--------------------------|---------------------|
| Software Development | | | | | |
| <i>Require-ments</i> | Analisis Sistem | 1 | 0,07 | Rp 7.000.000,00 | Rp 509.600,00 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,34 | Rp 7.000.000,00 | Rp 2.388.750,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,27 | Rp 7.000.000,00 | Rp 1.911.000,00 |
| <i>Implementation</i> | Programmer | 1 | 2,37 | Rp 8.000.000,00 | Rp 18.928.000,00 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,25 | Rp 8.000.000,00 | Rp 2.002.000,00 |
| TOTAL | | | | | Rp25.739.350 |

Total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.20**

6.2.1.13. Filter Pencarian

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 34
2. KSLOC : 1,802

3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,802^{1,104} \times 0,56 \\ &= 5,6 \quad \text{Person Month} \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned} \text{Durasi} &= 3 \times 5,6^{1/3} \\ &= 5,33 \quad \text{Bulan} \\ &= 106.6 \quad \text{Hari} \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 5,6^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 6 \text{ bulan (hasil pembulatan)} \end{aligned}$$

$$\begin{aligned} \text{Average Staff} &= \text{Effort} / \text{TDEV} \\ &= 5,6 / 6 \\ &= 1 \text{ (hasil pembulatan)} \end{aligned}$$

Berikut adalah total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.21**

Tabel 6.21 Hasil perhitungan biaya untuk kebutuhan fungsional Pencarian dengan Filter

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-------------------------------------|-----------------|---------|--------|--------------------------|--------------------------|
| Software Development | | | | | |
| <i>Require-ments</i> | Analisis Sistem | 1 | 0,09 | Rp 7.000.000 ,00 | Rp 596.960,00 |
| <i>Specifica-tions</i> | Analisis Sistem | 1 | 0,40 | Rp 7.000.000 ,00 | Rp 2.798.250, 00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,32 | Rp 7.000.000 ,00 | Rp 2.238.600, 00 |
| <i>Implemen-tation</i> | Pro-grammer | 1 | 2,77 | Rp 8.000.000 ,00 | Rp 22.172.800 ,00 |
| <i>Acceptance & deploy-ment</i> | Tehnici | 1 | 0,29 | Rp 8.000.000 ,00 | Rp 2.345.200, 00 |
| TOTAL | | | | | Rp30.151. 810 |

6.2.1.14. Login

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 15
2. KSLOC : 0,795
3. SF : 19,94
4. EM : 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 0,795^{1,104} \times 0,56 \\
 &= 1,3 \text{ Person Month}
 \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned} \text{Durasi} &= 3 \times 1,3^{1/3} \\ &= 3,27 \text{ Bulan} \\ &= 65.4 \text{ Hari} \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3,67 \times 1,3^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 4 \text{ bulan (hasil pembulatan)} \end{aligned}$$

$$\begin{aligned} \text{Average Staff} &= \text{Effort} / \text{TDEV} \\ &= 1,3 / 4 \\ &= 1 \text{ (hasil pembulatan)} \end{aligned}$$

Berikut adalah total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.22**

Tabel 6.22 Hasil perhitungan biaya untuk kebutuhan fungsional Login

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-----------------------------|-----------------|---------|--------|--------------------------|---------------|
| Software Development | | | | | |
| <i>Require-ments</i> | Analisis Sistem | 1 | 0,05 | Rp 7.000.000,00 | Rp 366.240,00 |

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-------------------------------------|-----------------|---------|--------|--------------------------|---------------------|
| <i>Specifica-tions</i> | Analisis Sistem | 1 | 0,25 | Rp 7.000.000,00 | Rp 1.716.750,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,20 | Rp 7.000.000,00 | Rp 1.373.400,00 |
| <i>Implemen-tation</i> | Pro-grammer | 1 | 1,70 | Rp 8.000.000,00 | Rp 13.603.200,00 |
| <i>Acceptance & deploy-ment</i> | Tehnici | 1 | 0,18 | Rp 8.000.000,00 | Rp 1.438.800,00 |
| TOTAL | | | | | Rp18.498.390 |

6.2.1.15. Ganti Password

Seperti perhitungan sebelumnya, perhitungan untuk fungsi menginputkan aktivitas adalah sebagai berikut :

1. UFP : 15
2. KSLOC : 0,795
3. SF : 19,94
4. EM :s 0,56
5. E : 1,1044

Kemudian, dilakukan estimasi dengan COCOMO II untuk menentukan usaha seperti berikut :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 0,795^{1,104} \times 0,56 \\
 &= 1,3 \text{ Person Month}
 \end{aligned}$$

Untuk mengetahui berapa lama waktu yang dibutuhkan dalam menyelesaikan fungsi tersebut, dilakukan perhitungan durasi :

$$\begin{aligned}
 \text{Durasi} &= 3 \times 1,3^{1/3} \\
 &= 3,27 \text{ Bulan} \\
 &= 65.4 \text{ Hari}
 \end{aligned}$$

Kemudian untuk mengetahui orang yang terlibat dalam tahap pengembangan adalah dengan menggunakan formula rata-rata yang didapat dari perhitungan pembagian Effort dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned}
 \text{TDEV} &= 3,67 \times 1,3^{0,28 + 0,2 \times (1,104 - 0,91)} \\
 &= 4 \text{ bulan (hasil pembulatan)}
 \end{aligned}$$

$$\begin{aligned}
 \text{Average Staff} &= \text{Effort} / \text{TDEV} \\
 &= 1,3 / 4 \\
 &= 1 \text{ (hasil pembulatan)}
 \end{aligned}$$

Berikut adalah total biaya yang dibutuhkan untuk memenuhi kebutuhan fungsional setelah dilakukan seperti yang terdapat pada **Tabel 6.23**

Tabel 6.23 Hasil perhitungan biaya untuk kebutuhan fungsional Ganti Password

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-----------------------------|-----------------|---------|--------|--------------------------|-----------------|
| Software Development | | | | | |
| <i>Require-ments</i> | Analisis Sistem | 1 | 0,05 | Rp 7.000.000,00 | Rp 366.240,00 |
| <i>Specifica-tions</i> | Analisis Sistem | 1 | 0,25 | Rp 7.000.000,00 | Rp 1.716.750,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,20 | Rp 7.000.000,00 | Rp 1.373.400,00 |

| Kelompok aktivitas | Jabatan | Pers on | Mon th | Standar Gaji (per-bulan) | Total |
|-------------------------------------|-------------|---------|--------|--------------------------|--------------------------|
| <i>Implemen-tation</i> | Pro-grammer | 1 | 1,70 | Rp 8.000.000 ,00 | Rp 13.603.200 ,00 |
| <i>Acceptance & deploy-ment</i> | Tehnisi | 1 | 0,18 | Rp 8.000.000 ,00 | Rp 1.438.800, 00 |
| TOTAL | | | | | Rp18.498. 390 |

6.2.2. Pemilihan Objek Penelitian

Setelah semua biaya kebutuhan fungsional diketahui, Selanjutnya dilakukan pemilihan objek penelitian dengan mencari bobot tiap kebutuhan fungsional dan akan diambil berdasarkan waktu yang dibutuhkan untuk mengembangkan fitur dengan rentang waktu 600 – 900 jam dan diatas 900 jam (*Complex* , *Very Complex*) dengan persamaan dibawah ini (**Tabel 6.24**):

$$\text{Bobot} = \frac{\text{Biaya fitur}}{\text{Biaya total fitur}} \times 100\% \quad (6.1)$$

Tabel 6.24 Hasil perhitungan Bobot tiap kebutuhan fungsional berdasarkan estimasi biaya yang diperoleh

| No | Nama fungsi | Hours | PM | Estimasi biaya | Bobot | Keterangan |
|----|---------------------------------|-------|-----|----------------|-------|------------|
| 1 | Melakukan update data pegawai | 787.2 | 4,4 | Rp27.832.440 | 7,9% | Diterima |
| 2 | Input aktivi-tas | 728 | 3,5 | R25.739.350 | 7,3% | Diterima |
| 3 | Melakukan persetujuan aktivitas | 692.8 | 3,0 | Rp24.494.810 | 6,9% | Diterima |
| 4 | Melakukan pembatalan | 692.8 | 3,0 | Rp24.494.810 | 6,9% | Diterima |

| | | | | | | |
|----|--|-------|-----|--------------|------|----------|
| | aktivitas | | | | | |
| 5 | Menampilkan capaian aktivitas | 584 | 1,8 | Rp20.648.050 | 5,8% | Ditolak |
| 6 | Menampilkan perangkat daerah | 584 | 1,8 | Rp20.648.050 | 5,8% | Ditolak |
| 7 | Menampilkan capaian kegiatan | 548.8 | 1,5 | Rp19.403.510 | 5,5% | Ditolak |
| 8 | Menampilkan indikator kerja | 584 | 1,8 | Rp20.648.050 | 5,8% | Ditolak |
| 9 | Menampilkan rekap penguasaan | 699.2 | 3,1 | Rp24.721.090 | 7% | Diterima |
| 10 | Menampilkan sasaran program | 651.2 | 2,5 | Rp23.023.990 | 6,5% | Diterima |
| 11 | Melakukan penilaian | 804.8 | 4,7 | Rp28.454.710 | 8,1% | Diterima |
| 12 | Mengenerate Rapor | 728 | 3,5 | Rp25.739.350 | 7,3% | Diterima |
| 13 | Melakukan Pencarian berdasarkan filer tertentu | 852.8 | 5,6 | Rp30.151.810 | 8,5% | Diterima |
| 14 | Login | 523.2 | 1,3 | Rp18.498.390 | 5,2% | Ditolak |
| 15 | Ganti Password | 523.2 | 1,3 | Rp18.498.390 | 5,2% | Ditolak |

Sesuai dengan perhitungan bobot yang telah dilakukan didapat ada **9 fungsi** yang akan dijadikan sebagai objek penelitian. Lebih lanjut, fungsi tersebut akan dibuat kedalam diagram FAST.

6.2.3. Pembuatan FAST Diagram

6.2.3.1. Melakukan *update* data pegawai

Fungsi ini dapat berjalan ketika *user* berhasil melakukan login dengan hak akses Admin SKPD hingga perekaman data terbaru dari pegawai, yang akan digambarkan pada diagram FAST (**Gambar 6.4**). Untuk memudahkan proses pemahaman terkait fungsi dan pembuatan diagram tersebut peneliti menjawab beberapa pertanyaan sebagai berikut :

1. *Bagaimana fungsi tersebut didapat?*
Fungsi *update* data pegawai tersebut termasuk pada kategori *high order function* yang mengakibatkan akses oleh *database* pegawai yang telah dilakukan pembaharuan sebelumnya pada satu atau lebih data pegawai.
2. *Mengapa fungsi tersebut bisa dijalankan ?*
Fungsi ini dapat dijalankan jika *system* mampu menampilkan seluruh pegawai yang terdapat pada suatu SKPD yang dapat diakses oleh Admin SKPD untuk keperluan memperbaharui data pegawai. Kemudian *system* dapat menampilkan data pegawai yang akan diperbaharui baik dari status atau penambahan entri data pegawai baru. Aktivitas ini kemudian disimpan kedalam *database* pegawai untuk setelahnya dapat diakses kembali oleh *system* untuk menampilkan setiap data pegawai yang telah diperbaharui.

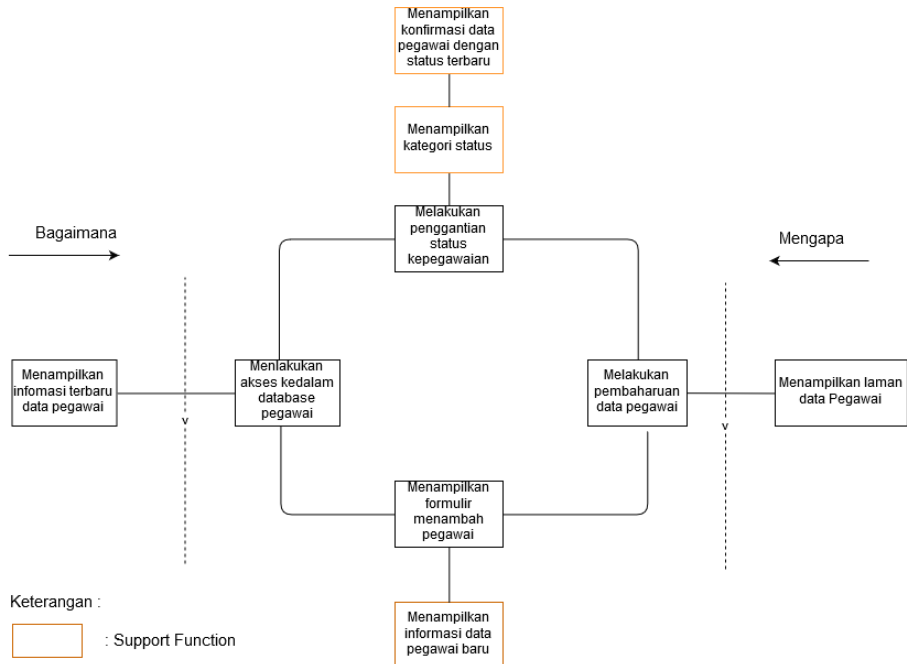
Untuk mendetailkan alur dan objektif fungsi peneliti menjawab pertanyaan dibawah ini :

3. *Kapan fungsi tersebut digunakan ?*
Fungsi pembaharuan data berupa *update* status kepegawaian pada suatu data pegawai dilakukan ketika *system* telah menampilkan detail seorang data pegawai yang akan diubah, kemudian *user* akan memperbarui status sesuai dengan yang

terjadi pada pegawai baik telah dimutasi ke SKPD/ divisi lain, telah pensiun, ganti atasan, atau telah naik jabatan.

Ketika pembaharuan data berupa ganti jabatan, *system* akan menampilkan *input* kepala jabatan/ atasan dari pegawai yang kemudian akan diganti oleh Admin SKPD sesuai dengan yang terjadi.

Selain dengan pembaharuan data, *user* dapat melakukan entri data pegawai baru pada sebuah SKPD ketika *system* menampilkan formulir entri data pegawai baru kemudian hasilnya akan direkam dan disimpan oleh *database* pegawai yang selanjutnya dapat ditampilkan oleh *system* untuk profil pegawai yang baru dimasukkan pada *system*.



Gambar 6.4 Diagram FAST untuk fungsi Update Data Pegawai

6.2.3.2. Melakukan *input* aktivitas

Fungsi ini dapat berjalan ketika *user* berhasil melakukan login dengan hak akses pejabat yang dinilai hingga informasi aktivitas pejabat terkait ditampilkan, yang akan digambarkan pada diagram FAST (**Gambar 6.5**). Untuk memudahkan proses pemahaman terkait fungsi dan pembuatan diagram tersebut peneliti menjawab beberapa pertanyaan sebagai berikut :

1. *Bagaimana fungsi tersebut didapat?*

Aktivitas pegawai didapat ditampilkan jika *system* telah melakukan akses kedalam database aktivitas

2. *Mengapa fungsi tersebut bisa dijalankan ?*

Fungsi yang dimaksud mampu berjalan ketika *system* menampilkan detail informasi formulir aktivitas yang dilaksanakan oleh pegawai. Formulir tersebut berisi mengenai detail aktivitas baik dari tanggal aktivitas yang dilakukan, nama aktivitas, kategori waktu pelaksanaan, dan aktivitas yang dilakukan.

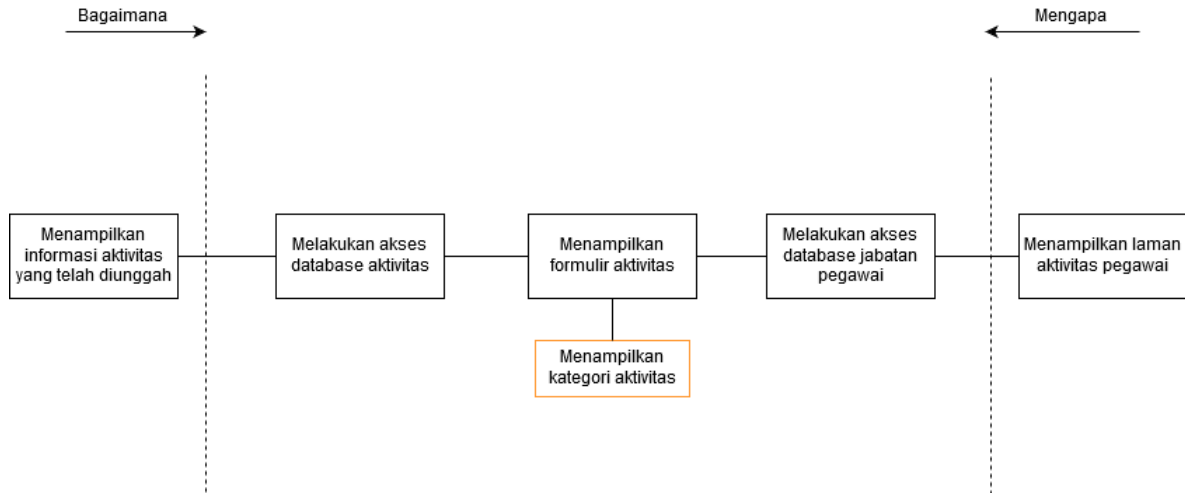
Untuk selanjutnya entri aktivitas pada formulir tersebut direkam dan disimpan oleh database aktivitas pegawai.

Untuk mendetailkan alur dan objektif fungsi peneliti menjawab pertanyaan dibawah ini :

3. *Kapan fungsi tersebut digunakan ?*

Fungsi halaman detail informasi aktivitas ini digunakan ketika *user* yaitu pejabat yang dinilai telah mengisi formulir aktivitas yang ditampilkan oleh *system*.

Ketika aktivitas pejabat yang dinilai telah dientrikan, pejabat penilai akan mendapat pemberitahuan mengenai aktivitas pegawai dalam status menunggu atau sedang antri untuk kemudian dilanjutkan dengan verifikasi aktivitas oleh pejabat penilai.



Keterangan :

: Secondary Function (Required)

Gambar 6.5 Diagram FAST untuk fungsi Input Aktivitas

6.2.3.3. Menyetujui aktivitas pegawai

Fungsi ini dapat berjalan ketika *user* berhasil melakukan login dengan hak akses pejabat penilai hingga informasi aktivitas yang ditampilkan disetujui oleh *user* dan digambarkan pada diagram FAST(**Gambar 6.6**). Untuk memudahkan proses pemahaman terkait fungsi dan pembuatan diagram tersebut peneliti menjawab beberapa pertanyaan sebagai berikut :

1. *Bagaimana fungsi tersebut didapat?*

Fungsi ini didapat ketika *system* menampilkan aktivitas pegawai dengan status aktivitas direvisi atau telah disetujui

2. *Mengapa fungsi tersebut bisa dijalankan ?*

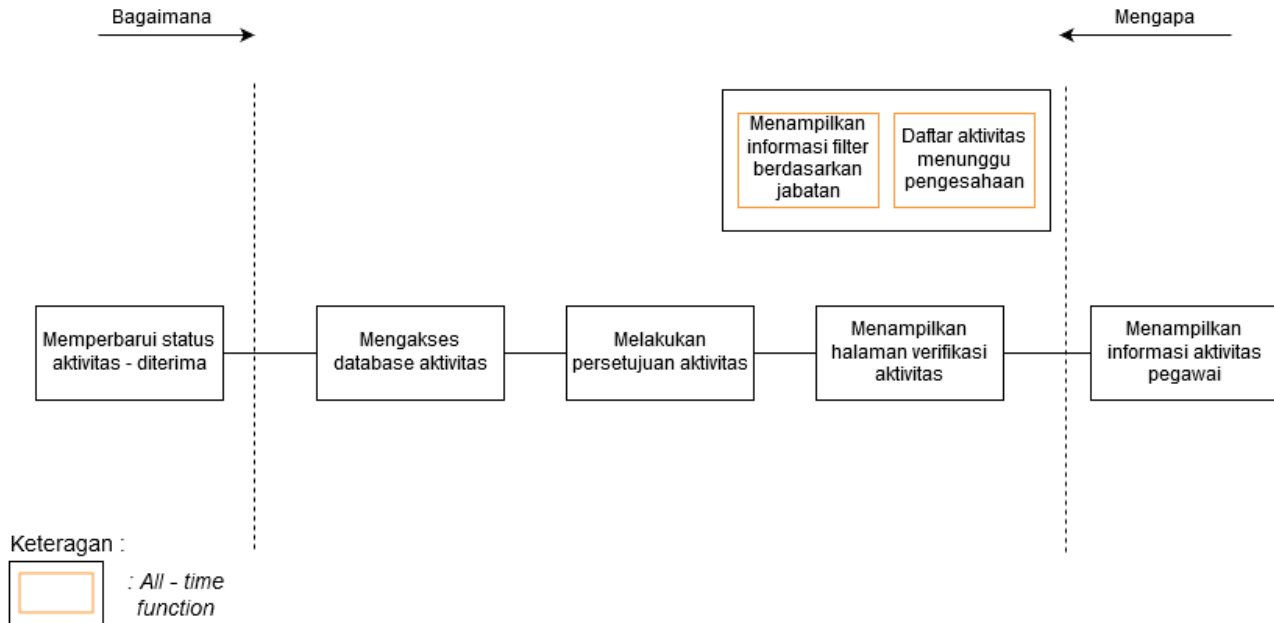
Fungsi ini dapat dijalankan akibat *system* menampilkan informasi hasil aktivitas yang telah di entrikan oleh pejabat yang dinilai dan menunggu untuk dilakukan verifikasi oleh *user* (pejabat penilai)

Untuk mendetailkan alur dan objektif fungsi peneliti menjawab pertanyaan dibawah ini :

3. *Kapan fungsi tersebut digunakan ?*

Fungsi halaman status aktivitas pegawai yang telah disetujui dapat digunakan saat *user* (pejabat penilai) telah melakukan verifikasi aktivitas.

Ketika aktivitas tersebut telah disetujui, pejabat yang dinilai akan mendapat pemberitahuan mengenai aktivitas yang mereka entrikan diterima oleh pejabat penilai.



Gambar 6.6 Diagram FAST untuk fungsi Mengesahkan Aktivitas

6.2.3.4. Membatalkan aktivitas pegawai

Fungsi ini dapat berjalan ketika *user* berhasil login dengan hak akses pejabat penilai hingga informasi aktivitas yang ditampilkan dibatalkan *user* (pejabat penilai) akibat ketidaksesuaian unggahan bukti aktivitas, yang akan digambarkan pada diagram FAST (**Gambar 6.7**). Untuk memudahkan proses pemahaman terkait fungsi dan pembuatan diagram tersebut peneliti menjawab beberapa pertanyaan sebagai berikut :

1. *Bagaimana fungsi tersebut didapat?*

Fungsi ini didapat ketika *system* menampilkan aktivitas pegawai dengan status aktivitas dibatalkan.

2. *Mengapa fungsi tersebut bisa dijalankan ?*

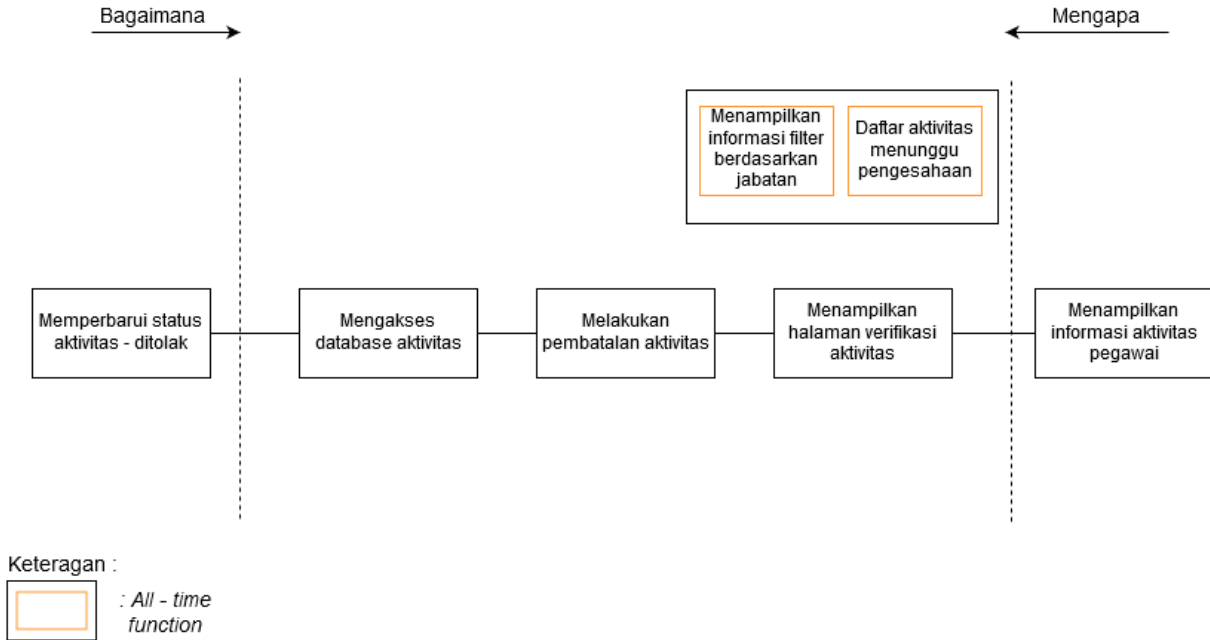
Fungsi ini dapat dijalankan akibat *system* menampilkan informasi hasil aktivitas yang telah di entrikan oleh pejabat yang dinilai dan menunggu untuk dilakukan verifikasi oleh *user* (pejabat penilai).

Untuk mendetailkan alur dan objektif fungsi peneliti menjawab pertanyaan dibawah ini :

3. *Kapan fungsi tersebut digunakan ?*

Fungsi halaman status aktivitas pegawai yang dibatalkan dapat digunakan saat *user* (pejabat penilai) telah melakukan verifikasi aktivitas.

Ketika aktivitas tersebut dibatalkan, pejabat yang dinilai akan mendapat pemberitahuan mengenai aktivitas yang mereka entrikan tidak diterima oleh pejabat penilai akibat tidak sesuai dengan peraturan berlaku.



Gambar 6.7 Diagram FAST untuk fungsi Membatalkan Aktivitas

6.2.3.5. Menampilkan informasi rekap penugasan

Fungsi ini dapat berjalan ketika *user* berhasil melakukan login dengan hak akses Pejabat penilai (merupakan atasan atau pejabat tingkat 1 dan 2) & pejabat yang dinilai (pegawai dan pejabat tingkat 3 & 4) hingga dapat menampilkan hasil seluruh penugasan aktivitas/ kegiatan yang dilakukan oleh pejabat yang dinilai (pejabat setingkat 3 dan 4), yang akan digambarkan pada diagram FAST (**Gambar 6.8**). Untuk memudahkan proses pemahaman terkait fungsi dan pembuatan diagram tersebut peneliti menjawab beberapa pertanyaan sebagai berikut :

1. *Bagaimana fungsi tersebut didapat?*

Fungsi menampilkan informasi rekap penugasan termasuk kedalam *high function order* yang didapat ketika *system* mengakses kedalam database penugasan dengan akses yang telah dijelaskan kemudian menampilkan seluruh informasi terkait penugasan yang diberikan kepada bawahan (pejabat penilai) kepada user (pejabat penilai) yang bersangkutan.

2. *Mengapa fungsi tersebut bisa dijalankan ?*

Fungsi ini didapat karena *system* mampu mengakses database penugasan untuk kedua pihak pejabat (pejabat yang dinilai dan pejabat yang menilai) yang kemudian menampilkan laman penugasan yang berisi informasi detail penugasan, atasan yang terlibat.

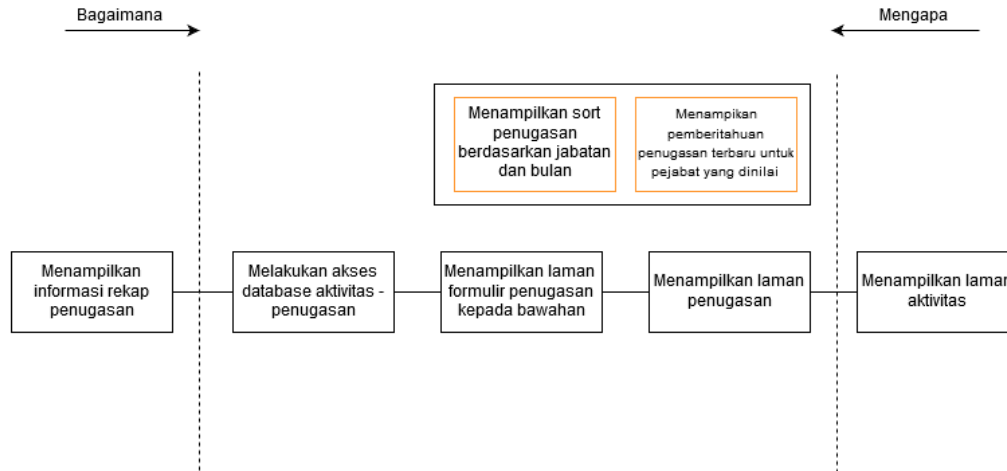
Untuk mendetailkan alur dan objektif fungsi peneliti menjawab pertanyaan dibawah ini :

3. *Kapan fungsi tersebut digunakan ?*


Fungsi ini dapat digunakan ketika *user* (pejabat yang dinilai, dan pejabat penilai) ingin mengetahui informasi lengkap mengenai penugasan.

Untuk tampilan pada laman penugasan dengan hak akses pejabat penilai akan berisi informasi mengenai semua penugasan yang diberikan kepada bawahannya (pejabat yang dinilai).

Untuk tampilan pada laman penugasan dengan hak akses pejabat yang dinilai akan berisi informasi terkait penugasan yang harus diselesaikan dan tidak dapat melihat seluruh detail penugasan pejabat yang dinilai lainnya.



Keterangan :

 : happen all the time function

Gambar 6.8 Diagram FAST untuk Menampilkan Penugasan

6.2.3.6. Menampilkan Sasaran Program

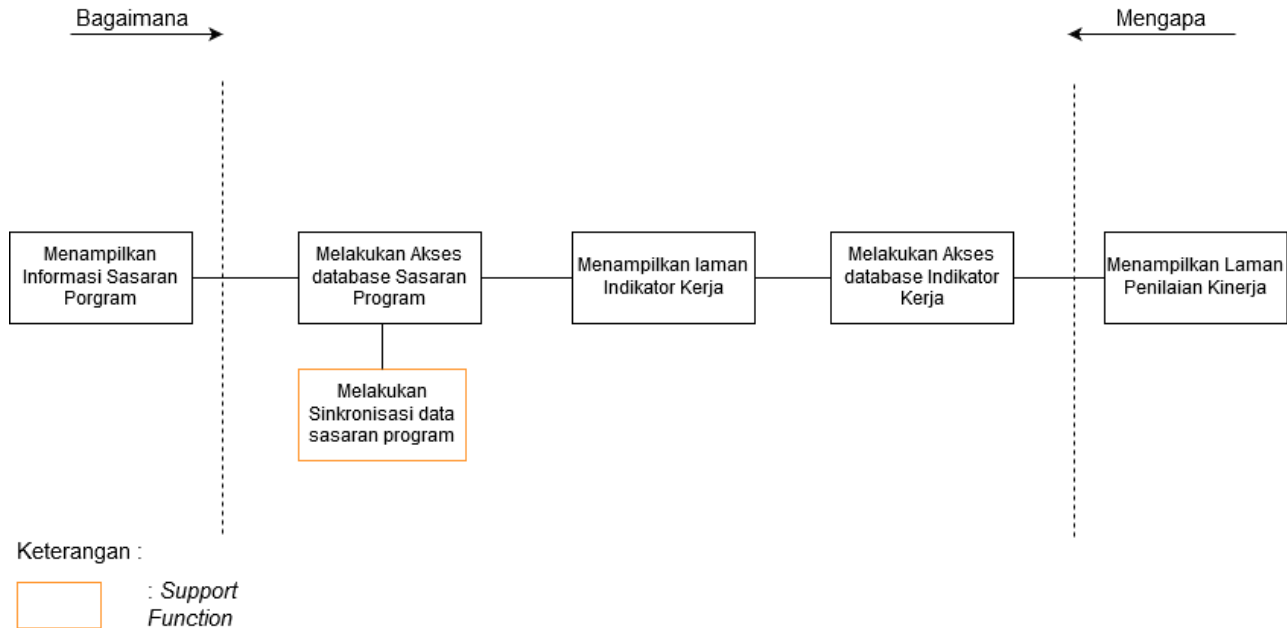
Fungsi ini dapat berjalan ketika *user* berhasil melakukan login dengan hak akses Admin SKPD hingga dapat menampilkan bobot penilaian setiap kegiatan/ aktivitas/ program yang dilakukan setiap pejabat penilai maupun pejabat yang dinilai dan akan digambarkan pada diagram FAST (**Gambar 6.9**). Untuk memudahkan proses pemahaman terkait fungsi dan pembuatan diagram tersebut peneliti menjawab beberapa pertanyaan sebagai berikut :

1. *Bagaimana fungsi tersebut didapat?*
Fungsi menampilkan informasi Sasaran Program termasuk kedalam *high function order* yang didapat ketika *system* mengakses kedalam database penugasan dengan akses yang telah dijelaskan kemudian menampilkan seluruh informasi terkait penugasan yang diberikan kepada bawahan (pejabat penilai) kepada user (pejabat penilai) yang bersangkutan.
2. *Mengapa fungsi tersebut bisa dijalankan ?*
Fungsi ini didapat karena *system* mampu mengakses database dan menampilkan laman Indikator Kinerja kemudian *system* mampu untuk melakukan sinkronisasi atau mengambil data dari aplikasi e-monev untuk akhirnya dapat menampilkan rincian Sasaran Program yang dijalankan setiap pegawai berdasarkan aplikasi yang berhubungan

Untuk mendetailkan alur dan objektif fungsi peneliti menjawab pertanyaan dibawah ini :

3. *Kapan fungsi tersebut digunakan ?*
Fungsi ini dapat digunakan ketika *user* (Admin SKPD) ingin mengetahui informasi terkait tentang bobot setiap program yang ada pada SKPD. User dapat melakukan pembaruan berupa sinkronisasi

bobot penilaian yang bersifat dinamis dan dapat berubah setiap saat (sesuai dengan peraturan Pemerintah Daerah setempat) dari aplikasi yang terintegrasi dengan aplikasi *e-Performance*.



Gambar 6.9 Diagram FAST untuk fitur menampilkan informasi Sasaran Program

6.2.3.7. Penilaian pegawai

Fungsi ini dapat berjalan ketika *user* berhasil melakukan login dengan hak akses Admin SKPD, pejabat penilai dan pejabat yang dinilai hingga *system* melakukan penilaian untuk seluruh pegawai terkait, yang akan digambarkan pada diagram FAST (**Gambar 6.10**). Untuk memudahkan proses pemahaman terkait fungsi dan pembuatan diagram tersebut peneliti menjawab beberapa pertanyaan sebagai berikut :

1. *Bagaimana fungsi tersebut didapat?*

Fungsi ini adalah termasuk *high order function* yang didapat ketika *system* berhasil melakukan perhitungan penilaian triwulan yang dilakukan oleh seluruh pegawai pada suatu SKPD, selain itu *system* menampilkan soal untuk penilaian yang diberikan oleh pejabat penilai yang terkait kepada pejabat yang dinilai.

2. *Mengapa fungsi tersebut bisa dijalankan ?*

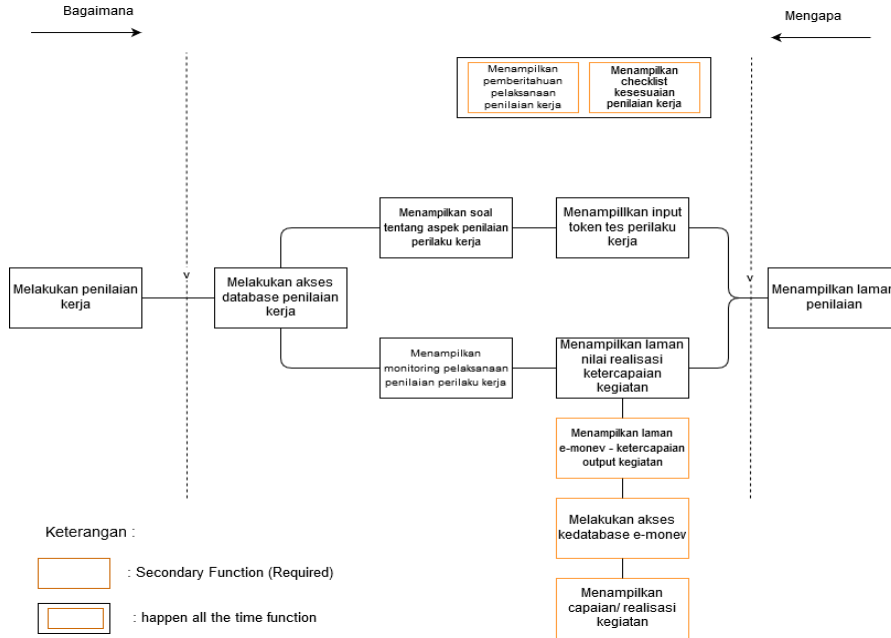
Fungsi ini dijalankan karena *system* mampu membuka penilaian dengan memberikan nilai selain itu, *system* akan memberikan notifikasi untuk mengingatkan pegawai untuk melakukan penilaian.

Untuk mendetailkan alur dan objektif fungsi peneliti menjawab pertanyaan dibawah ini :

3. *Kapan fungsi tersebut digunakan ?*

Fungsi ini digunakan untuk melakukan penilaian pegawai mulai dari performa kinerja Admin SKPD, Pejabat penilai dan pejabat yang dinilai.

Untuk memudahkan *user*, *system* juga menampilkan pemberitahuan pada laman utama aplikasi tentang jadwal pelaksanaan penilaian.



Gambar 6.10 Diagram FAST untuk fungsi Melakukan Penilaian pegawai

6.2.3.8. Melakukan pencarian berdasarkan filter tertentu

Fungsi ini dapat berjalan ketika *user* berhasil melakukan login dengan hak akses Admin SKPD, yang akan digambarkan pada diagram FAST (**Gambar 6.11**). Untuk memudahkan proses pemahaman terkait fungsi dan pembuatan diagram tersebut peneliti menjawab beberapa pertanyaan sebagai berikut :

Bagaimana fungsi tersebut didapat?

Fungsi ini didapat ketika *system* mampu mengakses database sesuai *filter* yang diinginkan *user* dalam melakukan pencarian berkategori dan menampilkan hasil pencarian sesuai yang diinginkan *user*.

Mengapa fungsi tersebut bisa dijalankan ?

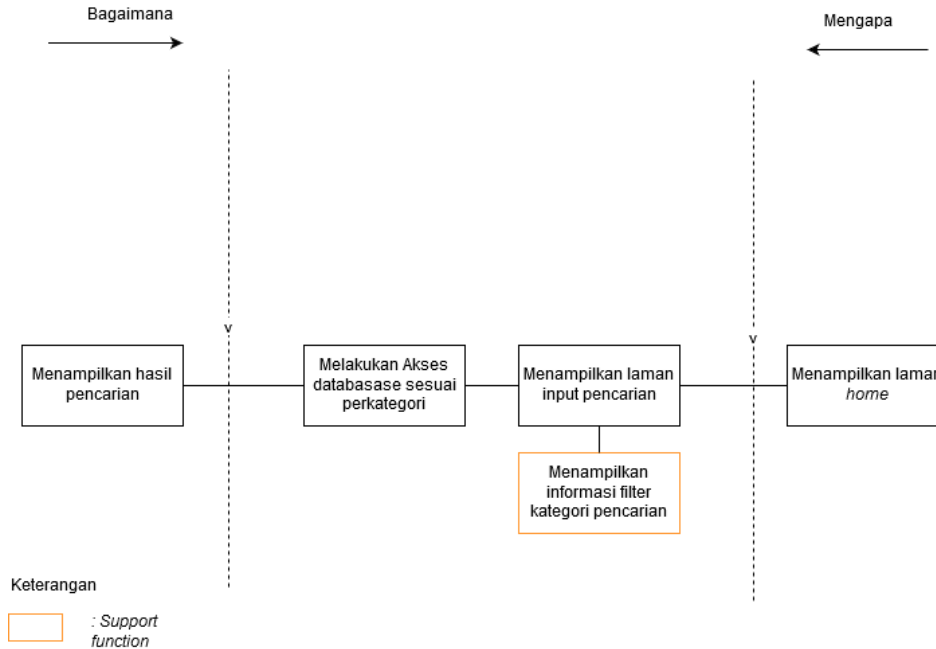
Fungsi ini didapat karena *system* mengakses database sesuai pencarian yang dientrikan oleh *user* kemudian menampilkan hasil pencarian.

System juga dapat menampilkan kategori pencarian berdasarkan filter pegawai, aktivitas, penugasan dan lain lain.

Untuk mendetailkan alur dan objektif fungsi peneliti menjawab pertanyaan dibawah ini :

Kapan fungsi tersebut digunakan ?

Fungsi ini digunakan untuk memudahkan *user* ketika mencari aktivitas, detail pegawai, atau *filter* lain. *Filter* ini bisa berisi kategori pencarian (mencari aktivitas, pegawai, penugasan) berdasarkan tanggal, hari, bulan, nama pegawai saat *user* sedang melakukan pencarian.



Gambar 6.11 Diagram FAST untuk fungsi Filter pada Pencarian

6.2.3.9. Men-generate rapor

Fungsi ini dapat berjalan ketika *user* berhasil melakukan login dengan hak akses Admin SKPD hingga aplikasi mampu menampilkan hasil rapor triwulan setiap pegawai suatu SKPD (baik pejabat yang dinilai maupun pejabat penilai), yang akan digambarkan pada diagram FAST. Untuk memudahkan proses pemahaman terkait fungsi dan pembuatan diagram tersebut peneliti menjawab beberapa pertanyaan sebagai berikut :

1. *Bagaimana fungsi tersebut didapat?*

Fungsi ini diperoleh dengan menampilkan hasil perhitungan penilaian dalam bentuk rapor 3 bulanan untuk setiap pegawai suatu PD.

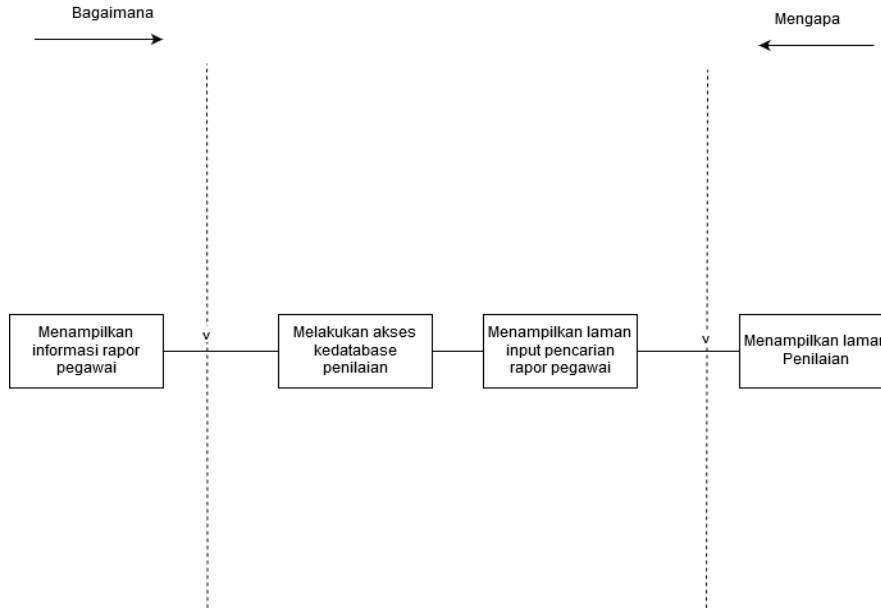
2. *Mengapa fungsi tersebut bisa dijalankan ?*

Fungsi ini dapat dijalankan ketika *system* mengakses database pegawai, penilaian, berserta aktivitas kegiatan yang berisi hasil perhitungan kinerja dari seorang pegawai atau kinerja suatu PD. Dengan mengakses hasil penilaian kinerja pegawai, kemudian *system* mampu menampilkan laporan berupa rapor kinerja pegawai selama 3 bulan.

Untuk mendetailkan alur dan objektif fungsi peneliti menjawab pertanyaan dibawah ini :

3. *Kapan fungsi tersebut digunakan*

Fungsi ini dapat digunakan untuk memudahkan *user* (Admin SKPD) untuk menampilkan hasil laporan penilaian kinerja setiap pegawai pada suatu SKPD yang kemudian akan didistribusikan kepada pegawai tersebut sebagai acuan penerimaan gaji dan bonus bagi pegawai (baik pejabat penilai dan pejabat yang dinilai).



Gambar 6.12 Diagram FAST untuk fungsi Generate Rapor

6.3. Creativity Phase

Tahapan Selanjutnya adalah pembuatan skenario berdasarkan pada 9 fitur yang terpilih dari perhitungan menggunakan metode perhitungan bobot.

Persiapan Perhitungan *Effort Multiplier* untuk alternatif skenario 1

Pada bagian ini adalah dokumentasi hasil dari perubahan *effort multiplier* yang nantinya akan digunakan untuk melakukan perhitungan *effort* pada skenario 1 dan digunakan dalam perhitungan biaya setiap fitur pada objek penelitian.

Hasil perhitungan *effort multiplier* yang baru dapat dilihat pada **Tabel 6.25**

Tabel 6.25 Hasil perubahan Effort Multiplier untuk alternatif 1

| Kategori | Effort Multiplier | Nominal | Skala | Hasil kalkulasi |
|----------|-------------------|---------|------------|-----------------|
| RUSE | RUSE | 1 | nominal | 1 |
| PERS | ACAP | 5 | Extra High | 0,5 |
| | PCAP | 5 | | |
| | PCON | 4 | | |
| RCPX | RELY | 1 | Extra Low | 0,49 |
| | DATA | 1 | | |
| | CPLX | 1 | | |
| | DOCU | 2 | | |
| PDIF | TIME | 3 | High | 1,29 |
| | STOR | 3 | | |
| | PVOL | 3 | | |
| PREX | AEXP | 4 | Extra High | 0,62 |
| | PEXP | 5 | | |
| | LTEX | 5 | | |

| | | | | |
|---------------------------------|------|---|-----------|------|
| FCIL | TOOL | 4 | Very High | 0,73 |
| | SITE | 5 | | |
| | RUSE | 1 | | |
| Total | | | | 0,14 |
| Total Effort Multiplier sebelum | | | | 0,56 |

Hasil Tabel 6.25 yang diberikan pada pengisian setiap *effort multiplier* dilakukan perubahan pada kategori :

- PRES (*Personnel Capability*)
Nilai yang diberikan pada kategori ini dinaikkan hingga mencapai skala *Extra High* dengan hasil kalkulasi sebesar 0,5. Semakin tinggi skala yang diberikan pada kategori ini akan memiliki nilai kalkulasi semakin rendah.
- RCPX (*Product Reliability and Complexity*)
Nilai yang diberikan untuk kategori ini diturunkan hingga mencapai skala *Extra Low* dengan hasil kalkulasi sebesar 0,49. Semakin rendah skala yang diberikan akan berdampak pada hasil kalkulasi yang semakin rendah.
- PREX (*Personnel Experience*)
Nilai yang diberikan untuk kategori ini dinaikkan hingga mencapai skala *Extra High* dengan hasil kalkulasi sebesar 0,62. Semakin tinggi skala berbanding terbalik dengan hasil kalkulasi yang semakin rendah.

Persiapan Perhitungan *Effort Multiplier* untuk alternatif skenario 2

Pada bagian ini adalah dokumentasi hasil dari perubahan *scale factor* yang nantinya akan digunakan untuk melakukan perhitungan *effort* pada skenario 2 dan digunakan dalam perhitungan biaya setiap fitur pada objek penelitian.

Hasil perhitungan *scale factor* yang baru dapat dilihat pada **Tabel 6.26**

Hasil **Tabel 6.26** yang diberikan pada pengisian setiap *scale factor* dilakukan perubahan pada kategori :

- **TEAM** (*Cohesion Team*)
 Nilai yang diberikan pada kategori ini dinaikkan hingga mencapai skala *Very High* dengan hasil kalkulasi sebesar 1,1. Semakin tinggi skala yang diberikan pada kategori ini akan memiliki nilai kalkulasi semakin rendah.

Tabel 6.26 Hasil perubahan perhitungan Scale Factor untuk alternatif 2

| Scale Factor | Skala | Hasil kalkulasi |
|-----------------------------|-----------|-----------------|
| PRE | Nominal | 3,72 |
| FLEX | Very high | 1,01 |
| RISK | Low | 5,65 |
| TEAM | Very high | 1,1 |
| PMAT | Nominal | 6,24 |
| Total | | 16,16 |
| Total Scale Factors Sebelum | | 19,44 |

6.3.1. Melakukan Update Data Pegawai

Dilihat dari total waktu yang diperlukan dalam pembangunan keseluruhan aplikasi yaitu sebesar 140 hari, skenario yang dilakukan akan berkaitan dengan memotong lama kerja dengan memanfaatkan pengurangan *cost driver &* dan *scale factor*.

Alternatif 1. Meminimalkan nilai *Effort Mutplier*

Dengan nilai *Size* dan *scale factor* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,325^{1,104} \times 0,14 \\ &= 0,56 \text{ Person Month} \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned} \text{Durasi} &= 3 \times 0,56^{1/3} \\ &= 2,48 \text{ Bulan} \\ &= 49,5 \text{ Hari} \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV
Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3 \times 3,67 \times 0,56^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 3,05 \text{ Bulan} \\ \text{Average} &= 0,56 \div 3,05 \\ \text{Staff} &= 0,2 \approx 01 \text{ Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.27**

Tabel 6.27 Hasil perhitungan biaya dari perubahan effort alternatif 1

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,04 | Rp 7.000.000,00 | Rp277.218,88 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,19 | Rp 7.000.000,00 | Rp1.299.463,50 |
| <i>Design</i> | Analisis Sistem | 1 | 0,15 | Rp 7.000.000,00 | Rp1.039.570,80 |
| <i>Implementation</i> | Programmer | 1 | 1,29 | Rp 8.000.000,00 | Rp10.296.701,26 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,14 | Rp 8.000.000,00 | Rp1.089.074,17 |
| TOTAL | | | | | Rp14.002.028,61 |

Hasil estimasi biaya dengan alternatif skenario 1 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp14.002.028,61 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

Alternatif 2. Meminimalkan nilai Scale Factor

Dengan nilai *Size* dan *effort multiplier* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 1,325^{1,07} \times 0,56 \\
 &= 2,23 \text{ Person Month}
 \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned}
 \text{Durasi} &= 3 \times 2,3^{1/3} \\
 &= 3,92 \text{ Bulan}
 \end{aligned}$$

$$= 78,3 \quad \text{Hari}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\text{TDEV} = 3 \times 3,67 \times 2,23^{0,28 + 0,2 \times (1,07 - 0,91)}$$

$$= 4,71 \quad \text{Bulan}$$

$$\text{Average} = 2,23 \div 4,71$$

Staff

$$= 0,5 \approx 01 \quad \text{Orang}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.28**

Tabel 6.28 Hasil perhitungan biaya dari perubahan effort alternatif 1

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,06 | Rp 7.000.000,00 | Rp438.705,66 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,29 | Rp 7.000.000,00 | Rp2.056.432,80 |
| <i>Design</i> | Analisis Sistem | 1 | 0,24 | Rp 7.000.000,00 | Rp1.645.146,24 |
| <i>Implementation</i> | Programmer | 1 | 2,04 | Rp 8.000.000,00 | Rp16.294.781,81 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,22 | Rp 8.000.000,00 | Rp1.723.486,54 |
| TOTAL | | | | | Rp22.158.553,00 |

Hasil estimasi biaya dengan alternatif skenario 2 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp22.158.553,05 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

6.3.2. *Input Aktivitas Pegawai*

Dilihat dari total waktu yang diperlukan dalam pembangunan keseluruhan aplikasi yaitu sebesar 140 hari, skenario yang dilakukan akan berkaitan dengan memotong lama kerja dengan memanfaatkan pengurangan *cost driver* & dan *scale factor*.

Alternatif 1. Meminimalkan nilai *Effort Mutplier*

Dengan nilai *Size* dan *scale factor* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,060^{1,104} \times 0,14 \\ &= 0,44 \text{ Person Month} \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned} \text{Durasi} &= 3 \times 0,56^{1/3} \\ &= 2,28 \text{ Bulan} \\ &= 45,6 \text{ Hari} \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV
Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3 \times 3,67 \times 0,56^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 2,28 \text{ Bulan} \\ \text{Average Staff} &= 0,44 \div 2,28 \\ &= 0,2 \approx 01 \text{ Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.29**

Tabel 6.29 Hasil perhitungan biaya dari perubahan effort alternatif 1

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,04 | Rp 7.000.000,00 | Rp255.356,55 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,17 | Rp 7.000.000,00 | Rp1.196.983,82 |
| <i>Design</i> | Analisis Sistem | 1 | 0,14 | Rp 7.000.000,00 | Rp957.587,05 |
| <i>Implementation</i> | Programmer | 1 | 1,19 | Rp 8.000.000,00 | Rp9.484.671,78 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,13 | Rp 8.000.000,00 | Rp1.003.186,44 |
| TOTAL | | | | | Rp12.897.785,64 |

Hasil estimasi biaya dengan alternatif skenario 1 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp12.897.785,64 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

Alternatif 2. Meminimalkan nilai *Scale Factor*

Dengan nilai *Size* dan *effort multiplier* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,060^{1,07} \times 0,56 \\ &= 1,75 \text{ Person Month} \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\text{Durasi} = 3 \times 1,75^{1/3}$$

$$= 3,62 \quad \text{Bulan}$$

$$= 72,3 \quad \text{Hari}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\text{TDEV} = 3 \times 1,75 \times 1,66^{0,28 + 0,2 \times (1,07 - 0,91)}$$

$$= 4,37 \quad \text{Bulan}$$

$$\text{Average} = 1,75 \div 4,37$$

$$\text{Staff} = 0,4 \approx 01 \quad \text{Orang}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.30**

Tabel 6.30 Hasil perhitungan biaya dari perubahan effort alternatif 2

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,06 | Rp 7.000.000,00 | Rp405.095,10 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,27 | Rp 7.000.000,00 | Rp1.898.883,26 |
| <i>Design</i> | Analisis Sistem | 1 | 0,22 | Rp 7.000.000,00 | Rp1.519.106,61 |
| <i>Implementation</i> | Programmer | 1 | 1,88 | Rp 8.000.000,00 | Rp15.046.389,25 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,20 | Rp 8.000.000,00 | Rp1.591.445,02 |
| TOTAL | | | | | Rp20.460.919,23 |

Hasil estimasi biaya dengan alternatif skenario 2 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp20.460.919,23 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

6.3.3. Menyetujui Aktivitas Pegawai

Dilihat dari total waktu yang diperlukan dalam pembangunan keseluruhan aplikasi yaitu sebesar 140 hari, skenario yang dilakukan akan berkaitan dengan memotong lama kerja dengan memanfaatkan pengurangan *cost driver* & dan *scale factor*.

Alternatif 1. Meminimalkan nilai *Effort Mutplier*

Dengan nilai *Size* dan *scale factor* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,007^{1,104} \times 0,14 \\ &= 0,41 \text{ Person Month} \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned} \text{Durasi} &= 3 \times 0,56^{1/3} \\ &= 2,24 \text{ Bulan} \\ &= 44,7 \text{ Hari} \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV
Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3 \times 3,67 \times 0,56^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 2,77 \text{ Bulan} \\ \text{Average} &= 0,44 \div 2,77 \\ \text{Staff} &= 0,1 \approx 01 \text{ Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.31**

Tabel 6.31 Hasil perhitungan biaya dari perubahan effort alternatif 1

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,04 | Rp 7.000.000,00 | Rp250.579,95 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,17 | Rp 7.000.000,00 | Rp1.174.593,51 |
| <i>Design</i> | Analisis Sistem | 1 | 0,13 | Rp 7.000.000,00 | Rp939.674,81 |
| <i>Implementation</i> | Programmer | 1 | 1,16 | Rp 8.000.000,00 | Rp9.307.255,22 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,12 | Rp 8.000.000,00 | Rp984.421,23 |
| TOTAL | | | | | Rp12.656.524,71 |

Hasil estimasi biaya dengan alternatif skenario 1 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp12.656.524,71 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

Alternatif 2. Meminimalkan nilai Scale Factor

Dengan nilai *Size* dan *effort multiplier* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,007^{1,07} \times 0,56 \\ &= 1,66 \text{ Person Month} \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\text{Durasi} = 3 \times 1,66^{1/3}$$

$$= 3,55 \quad \text{Bulan}$$

$$= 71 \quad \text{Hari}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV.

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3 \quad \times \quad 3,67 \times 1,66^{0,28 + 0,2 \times (1,07 - 0,91)} \\ &= 4,30 \quad \text{Bulan} \\ \text{Average} &= 1,66 \quad \div \quad 4,30 \\ \text{Staff} &= 0,4 \quad \approx \quad 1 \quad \text{Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.32**

Tabel 6.32 Hasil perhitungan biaya dari perubahan effort alternatif 2

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,06 | Rp 7.000.000,00 | Rp397.740,54 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,27 | Rp 7.000.000,00 | Rp1.864.408,77 |
| <i>Design</i> | Analisis Sistem | 1 | 0,21 | Rp 7.000.000,00 | Rp1.491.527,02 |
| <i>Implementation</i> | Programmer | 1 | 1,85 | Rp 8.000.000,00 | Rp14.773.219,98 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,20 | Rp 8.000.000,00 | Rp1.562.552,11 |
| TOTAL | | | | | Rp20.089.448,42 |

Hasil estimasi biaya dengan alternatif skenario 2 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp20.089.448,42 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

6.3.4. Melakukan pembatalan Aktivitas Pegawai

Dilihat dari total waktu yang diperlukan dalam pembangunan keseluruhan aplikasi yaitu sebesar 140 hari, skenario yang dilakukan akan berkaitan dengan memotong lama kerja dengan memanfaatkan pengurangan *cost driver* & dan *scale factor*.

Alternatif 1. Meminimalkan nilai *Effort Mutplier*

Dengan nilai *Size* dan *scale factor* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,007^{1,104} \times 0,14 \\ &= 0,41 \text{ Person Month} \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned} \text{Durasi} &= 3 \times 0,56^{1/3} \\ &= 2,24 \text{ Bulan} \\ &= 44,7 \text{ Hari} \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV
Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3 \times 3,67 \times 0,56^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 2,77 \text{ Bulan} \\ \text{Average} &= 0,44 \div 2,77 \\ \text{Staff} &= 0,1 \approx 01 \text{ Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.33**

Tabel 6.33 Hasil perhitungan biaya dari perubahan effort alternatif 1

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,04 | Rp 7.000.000,00 | Rp250.579,95 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,17 | Rp 7.000.000,00 | Rp1.174.593,51 |
| <i>Design</i> | Analisis Sistem | 1 | 0,13 | Rp 7.000.000,00 | Rp939.674,81 |
| <i>Implementation</i> | Programmer | 1 | 1,16 | Rp 8.000.000,00 | Rp9.307.255,22 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,12 | Rp 8.000.000,00 | Rp984.421,23 |
| TOTAL | | | | | Rp12.656.524,71 |

Hasil estimasi biaya dengan alternatif skenario 1 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp12.656.524,71 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

Alternatif 2. Meminimalkan nilai *Scale Factor*

Dengan nilai *Size* dan *effort multiplier* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 1,007^{1,07} \times 0,56 \\
 &= 1,66 \text{ Person Month}
 \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned}
 \text{Durasi} &= 3 \times 1,66^{1/3} \\
 &= 3,55 \text{ Bulan} \\
 &= 71 \text{ Hari}
 \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned}
 \text{TDEV} &= 3 \times 3,67 \times 1,66^{0,28 + 0,2 \times (1,07 - 0,91)} \\
 &= 4,30 \text{ Bulan}
 \end{aligned}$$

$$\text{Average} = 1,66 \div 4,30$$

Staff

$$= 0,4 \approx 1 \text{ Orang}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.34**

Tabel 6.34 Hasil perhitungan biaya dari perubahan effort alternatif 2

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|-----------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,06 | Rp 7.000.000,00 | Rp397.740,54 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,27 | Rp 7.000.000,00 | Rp1.864.408,77 |
| <i>Design</i> | Analisis Sistem | 1 | 0,21 | Rp 7.000.000,00 | Rp1.491.527,02 |
| <i>Implementation</i> | Programmer | 1 | 1,85 | Rp 8.000.000,00 | Rp14.773.219,98 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,20 | Rp 8.000.000,00 | Rp1.562.552,11 |

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|--------------------|---------|--------|-------|-------------------------|------------------------|
| TOTAL | | | | | Rp20.089.448,42 |

Hasil estimasi biaya dengan alternatif skenario 2 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp20.089.448,42 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

6.3.5. Menampilkan Informasi Penugasan Pegawai

Dilihat dari total waktu yang diperlukan dalam pembangunan keseluruhan aplikasi yaitu sebesar 140 hari, skenario yang dilakukan akan berkaitan dengan memotong lama kerja dengan memanfaatkan pengurangan *cost driver* & dan *scale factor*.

Alternatif 1. Meminimalkan nilai *Effort Mutplier*

Dengan nilai *Size* dan *scale factor* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 1,060^{1,104} \times 0,14 \\
 &= 0,44 \text{ Person Month}
 \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned}
 \text{Durasi} &= 3 \times 0,56^{1/3} \\
 &= 2,28 \text{ Bulan} \\
 &= 45,6 \text{ Hari}
 \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned}
 \text{TDEV} &= 3 \quad \times \quad 3,67 \times 0,56^{0,28 + 0,2 \times (1,104 - 0,91)} \\
 &= 2,28 \quad \text{Bulan} \\
 \text{Average} &= 0,44 \quad \div \quad 2,28 \\
 \text{Staff} &= 0,2 \quad \approx \quad 01 \quad \text{Orang}
 \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.45**

Tabel 6.35 Hasil perhitungan biaya dari perubahan effort alternatif 1

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,04 | Rp 7.000.000,00 | Rp255.356,55 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,17 | Rp 7.000.000,00 | Rp1.196.983,82 |
| <i>Design</i> | Analisis Sistem | 1 | 0,14 | Rp 7.000.000,00 | Rp957.587,05 |
| <i>Implementation</i> | Programmer | 1 | 1,19 | Rp 8.000.000,00 | Rp9.484.671,78 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,13 | Rp 8.000.000,00 | Rp1.003.186,44 |
| TOTAL | | | | | Rp12.897.785,64 |

Hasil estimasi biaya dengan alternatif skenario 1 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp12.897.785,64 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

Alternatif 2. Meminimalkan nilai Scale Factor

Dengan nilai *Size* dan *effort multiplier* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,060^{1,07} \times 0,56 \\ &= 1,75 \text{ Person Month} \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned} \text{Durasi} &= 3 \times 1,75^{1/3} \\ &= 3,62 \text{ Bulan} \\ &= 72,3 \text{ Hari} \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3 \times 1,75 \times 1,66^{0,28 + 0,2 \times (1,07 - 0,91)} \\ &= 4,37 \text{ Bulan} \\ \text{Average} &= 1,75 \div 4,37 \\ \text{Staff} &= 0,4 \approx 01 \text{ Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.36**

Tabel 6.36 Hasil perhitungan biaya dari perubahan effort alternatif 2

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|-----------------------------|-----------------|--------|-------|-------------------------|----------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,06 | Rp 7.000.000,00 | Rp405.095,10 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,27 | Rp 7.000.000,00 | Rp1.898.883,26 |

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| <i>Design</i> | Analisis Sistem | 1 | 0,22 | Rp 7.000.000,00 | Rp1.519.106,61 |
| <i>Implementation</i> | Programmer | 1 | 1,88 | Rp 8.000.000,00 | Rp15.046.389,25 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,20 | Rp 8.000.000,00 | Rp1.591.445,02 |
| TOTAL | | | | | Rp20.460.919,23 |

Hasil estimasi biaya dengan alternatif skenario 2 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp20.460.919,23 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

6.3.6. Menampilkan Informasi Sasaran Program

Dilihat dari total waktu yang diperlukan dalam pembangunan keseluruhan aplikasi yaitu sebesar 140 hari, skenario yang dilakukan akan berkaitan dengan memotong lama kerja dengan memanfaatkan pengurangan *cost driver* & dan *scale factor*.

Alternatif 1. Meminimalkan nilai *Effort Mutplier*

Dengan nilai *Size* dan *scale factor* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 0,848^{1,104} \times 0,14 \\
 &= 0,34 \text{ Person Month}
 \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned}
 \text{Durasi} &= 3 \times 0,34^{1/3} \\
 &= 2,1 \text{ Bulan} \\
 &= 42 \text{ Hari}
 \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3 \times 3,67 \times 0,34^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 2,61 \quad \text{Bulan} \\ \text{Average} &= 0,34 \div 2,61 \\ \text{Staff} &= 0,1 \approx 01 \quad \text{Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada .

Tabel 6.37.

Tabel 6.37 Hasil perhitungan biaya dari perubahan effort alternatif 1

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,03 | Rp 7.000.000,00 | Rp235.218,35 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,16 | Rp 7.000.000,00 | Rp1.102.586,00 |
| <i>Design</i> | Analisis Sistem | 1 | 0,13 | Rp 7.000.000,00 | Rp882.068,80 |
| <i>Implementation</i> | Programmer | 1 | 1,09 | Rp 8.000.000,00 | Rp8.736.681,44 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,12 | Rp 8.000.000,00 | Rp924.072,08 |
| TOTAL | | | | | Rp11.880.626,66 |

Hasil estimasi biaya dengan alternatif skenario 1 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp11.880.626,66 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

Alternatif 2. Meminimalkan nilai *Scale Factor*

Dengan nilai *Size* dan *effort multiplier* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned} \text{Effort} &= 2,94 \times 0,848^{1,07} \times 0,56 \\ &= 1,38 \text{ Person Month} \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned} \text{Durasi} &= 3 \times 1,38^{1/3} \\ &= 3,34 \text{ Bulan} \\ &= 66,8 \text{ Hari} \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV

Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3 \times 1,38 \times 1,66^{0,28 + 0,2 \times (1,07 - 0,91)} \\ &= 4,06 \text{ Bulan} \\ \text{Average} &= 1,38 \div 4,306 \\ \text{Staff} &= 0,3 \approx 01 \text{ Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.38**

Tabel 6.38 Hasil perhitungan biaya dari perubahan effort alternatif 2

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,05 | Rp 7.000.000,00 | Rp374.059,53 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,25 | Rp 7.000.000,00 | Rp1.753.404,06 |
| <i>Design</i> | Analisis Sistem | 1 | 0,20 | Rp 7.000.000,00 | Rp1.402.723,25 |
| <i>Implementation</i> | Programmer | 1 | 1,74 | Rp 8.000.000,00 | Rp13.893.639,83 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,18 | Rp 8.000.000,00 | Rp1.469.519,60 |
| TOTAL | | | | | Rp18.893.346,28 |

Hasil estimasi biaya dengan alternatif skenario 2 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp18.893.346,28 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

6.3.7. Melakukan Penilaian

Dilihat dari total waktu yang diperlukan dalam pembangunan keseluruhan aplikasi yaitu sebesar 140 hari, skenario yang dilakukan akan berkaitan dengan memotong lama kerja dengan memanfaatkan pengurangan *cost driver* & dan *scale factor*.

Alternatif 1. Meminimalkan nilai *Effort Multiplier*

Dengan nilai *Size* dan *scale factor* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 1,537^{1,104} \times 0,14 \\
 &= 0,66 \text{ Person Month}
 \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned}
 \text{Durasi} &= 3 \times 0,66^{1/3} \\
 &= 2,61 \text{ Bulan} \\
 &= 52,3 \text{ Hari}
 \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned}
 \text{TDEV} &= 3 \times 3,67 \times 0,66^{0,28 + 0,2 \times (1,104 - 0,91)} \\
 &= 3,22 \text{ Bulan}
 \end{aligned}$$

$$\text{Average} = 0,66 \div 3,22$$

Staff

$$= 0,2 \approx 01 \text{ Orang}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada Tabel 6.39

Tabel 6.39 Hasil perhitungan biaya dari perubahan effort alternatif 1

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,04 | Rp 7.000.000,00 | Rp292.787,10 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,20 | Rp 7.000.000,00 | Rp1.372.439,53 |
| <i>Design</i> | Analisis Sistem | 1 | 0,16 | Rp 7.000.000,00 | Rp1.097.951,63 |
| <i>Implementation</i> | Programmer | 1 | 1,36 | Rp 8.000.000,00 | Rp10.874.949,46 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,14 | Rp 8.000.000,00 | Rp1.150.235,04 |
| TOTAL | | | | | Rp14.788.362,76 |

Hasil estimasi biaya dengan alternatif skenario 1 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp14.788.362,76 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario

Alternatif 2. Meminimalkan nilai *Scale Factor*

Dengan nilai *Size* dan *effort multiplier* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,537^{1,07} \times 0,56 \\ &= 2,61 \text{ Person Month} \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned} \text{Durasi} &= 3 \times 2,61^{1/3} \\ &= 4,13 \text{ Bulan} \\ &= 82,6 \text{ Hari} \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3 \times 3,67 \times 2,61^{0,28 + 0,2 \times (1,07 - 0,91)} \\ &= 4,95 \text{ Bulan} \\ \text{Average Staff} &= 3,61 \div 4,95 \\ &= 0,5 \approx 1 \text{ Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada Tabel 6.40

Tabel 6.40 Hasil perhitungan biaya dari perubahan effort alternatif 2

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,07 | Rp 7.000.000,00 | Rp462.591,49 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,31 | Rp 7.000.000,00 | Rp2.168.397,59 |
| <i>Design</i> | Analisis Sistem | 1 | 0,25 | Rp 7.000.000,00 | Rp1.734.718,07 |
| <i>Implementation</i> | Programmer | 1 | 2,15 | Rp 8.000.000,00 | Rp17.181.969,48 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,23 | Rp 8.000.000,00 | Rp1.817.323,70 |
| TOTAL | | | | | Rp23.365.000,33 |

Hasil estimasi biaya dengan alternatif skenario 2 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp23.365.000,33 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

6.3.8. Melakukan generate rapor

Dilihat dari total waktu yang diperlukan dalam pembangunan keseluruhan aplikasi yaitu sebesar 140 hari, skenario yang dilakukan akan berkaitan dengan memotong lama kerja dengan memanfaatkan pengurangan *cost driver &* dan *scale factor*.

Alternatif 1. Meminimalkan nilai Effort Mutplier

Dengan nilai *Size* dan *scale factor* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,166^{1,104} \times 0,14 \\ &= 0,49 \text{ Person Month} \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned} \text{Durasi} &= 3 \times 0,49^{1/3} \\ &= 2,36 \text{ Bulan} \\ &= 4,72 \text{ Hari} \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3 \times 3,67 \times 0,49^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 2,92 \text{ Bulan} \end{aligned}$$

$$\text{Average} = 0,49 \div 2,92$$

Staff

$$= 0,2 \approx 01 \text{ Orang}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.41**

Tabel 6.41 Hasil perhitungan biaya dari perubahan effort alternatif 1

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|-----------------------------|-----------------|--------|-------|-------------------------|----------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,04 | Rp 7.000.000,00 | Rp264.475,24 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,18 | Rp 7.000.000,00 | Rp1.239.727,70 |

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| <i>Design</i> | Analisis Sistem | 1 | 0,14 | Rp 7.000.000,00 | Rp991.782,16 |
| <i>Implementation</i> | Programmer | 1 | 1,23 | Rp 8.000.000,00 | Rp9.823.366,19 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,13 | Rp 8.000.000,00 | Rp1.039.009,89 |
| TOTAL | | | | | Rp13.358.361,19 |

Hasil estimasi biaya dengan alternatif skenario 1 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp13.358.361,19 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario

Alternatif 2. Meminimalkan nilai *Scale Factor*

Dengan nilai *Size* dan *effort multiplier* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 1,166^{1,07} \times 0,56 \\
 &= 1,94 \text{ Person Month}
 \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned}
 \text{Durasi} &= 3 \times 1,94^{1/3} \\
 &= 3,74 \text{ Bulan} \\
 &= 74,8 \text{ Hari}
 \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3 \times 3,67 \times 1,94^{0,28 + 0,2 \times (1,07 - 0,91)} \\ &= 4,51 \quad \text{Bulan} \\ \text{Average} &= 1,94 \div 4,51 \\ \text{Staff} &= 0,4 \approx 1 \quad \text{Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada

Tabel 6.42

Tabel 6.42 Hasil perhitungan biaya dari perubahan effort alternatif 2

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,06 | Rp 7.000.000,00 | Rp419.123,92 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,28 | Rp 7.000.000,00 | Rp1.964.643,40 |
| <i>Design</i> | Analisis Sistem | 1 | 0,22 | Rp 7.000.000,00 | Rp1.571.714,72 |
| <i>Implementation</i> | Programmer | 1 | 1,95 | Rp 8.000.000,00 | Rp15.567.460,06 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,21 | Rp 8.000.000,00 | Rp1.646.558,28 |
| TOTAL | | | | | Rp21.169.500,37 |

Hasil estimasi biaya dengan alternatif skenario 2 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar

Rp21.169.500,37 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

6.3.9. Melakukan Pencarian berdasarkan Filter

Dilihat dari total waktu yang diperlukan dalam pembangunan keseluruhan aplikasi yaitu sebesar 140 hari, skenario yang dilakukan akan berkaitan dengan memotong lama kerja dengan memanfaatkan pengurangan *cost driver* & dan *scale factor*.

Alternatif 1. Meminimalkan nilai *Effort Mutplier*

Dengan nilai *Size* dan *scale factor* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned} \text{Effort} &= 2,94 \times 1,802^{1,104} \times 0,14 \\ &= 0,79 \text{ Person Month} \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned} \text{Durasi} &= 3 \times 0,79^{1/3} \\ &= 2,77 \text{ Bulan} \\ &= 55,4 \text{ Hari} \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV

Dimana koefisien C,D,B bernilai C = 3,67, D = 0,28, B = 0,91.

$$\begin{aligned} \text{TDEV} &= 3 \times 3,67 \times 0,79^{0,28 + 0,2 \times (1,104 - 0,91)} \\ &= 3,40 \text{ Bulan} \\ \text{Average} &= 0,79 \div 3,40 \\ \text{Staff} &= 0,2 \approx 01 \text{ Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada **Tabel 6.43**

Tabel 6.43 Hasil perhitungan biaya dari perubahan effort alternatif 1

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|-----------------|--------|-------|-------------------------|------------------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,04 | Rp 7.000.000,00 | Rp310.443,76 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,21 | Rp 7.000.000,00 | Rp1.455.205,10 |
| <i>Design</i> | Analisis Sistem | 1 | 0,17 | Rp 7.000.000,00 | Rp1.164.164,08 |
| <i>Implementation</i> | Programmer | 1 | 1,44 | Rp 8.000.000,00 | Rp11.530.768,05 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,15 | Rp 8.000.000,00 | Rp1.219.600,47 |
| TOTAL | | | | | Rp15.680.181,46 |

Hasil estimasi biaya dengan alternatif skenario 1 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp15.680.181,46 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario

Alternatif 2. Meminimalkan nilai *Scale Factor*

Dengan nilai *Size* dan *effort multiplier* yang sama, didapat hasil *effort* dari alternatif skenario 1 adalah :

$$\begin{aligned}
 \text{Effort} &= 2,94 \times 1,802^{1,07} \times 0,56 \\
 &= 3,09 \text{ Person Month}
 \end{aligned}$$

Kemudian dari hasil perhitungan *effort* yang baru dihitung kembali durasi yang dibutuhkan dalam melakukan pengembangan aplikasi seperti berikut :

$$\begin{aligned} \text{Durasi} &= 3 \times 3,09^{1/3} \\ &= 4,37 \text{ Bulan} \\ &= 87,4 \text{ Hari} \end{aligned}$$

Selanjutnya dilakukan perhitungan sumber daya yang terlibat dalam tahap pengembangan fitur dengan menggunakan persamaan rata-rata dari pembagian *effort* dengan TDEV

Dimana koefisien C,D,B bernilai $C = 3,67$, $D = 0,28$, $B = 0,91$.

$$\begin{aligned} \text{TDEV} &= 3 \times 3,67 \times 3,09^{0,28 + 0,2 \times (1,07 - 0,91)} \\ &= 5,22 \text{ Bulan} \\ \text{Average} &= 3,09 \div 5,22 \\ \text{Staff} &= 0,6 \approx 1 \text{ Orang} \end{aligned}$$

Dari hasil perhitungan diatas, kemudian didapat rincian biaya untuk setiap aktivitas pada tahap pengembangan fitur Update data pegawai pada Tabel 6.44

Tabel 6.44 Hasil perhitungan biaya dari perubahan effort alternatif 2

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|-----------------------------|-----------------|--------|-------|-------------------------|----------------|
| Software Development | | | | | |
| <i>Requirements</i> | Analisis Sistem | 1 | 0,07 | Rp 7.000.000,00 | Rp489.636,00 |
| <i>Specifications</i> | Analisis Sistem | 1 | 0,33 | Rp 7.000.000,00 | Rp2.295.168,74 |
| <i>Design</i> | Analisis Sistem | 1 | 0,26 | Rp 7.000.000,00 | Rp1.836.134,99 |

| Kelompok aktivitas | Jabatan | Person | Month | Standar Gaji (perbulan) | Total |
|------------------------------------|----------------|---------------|--------------|--------------------------------|------------------------|
| <i>Implementation</i> | Programmer | 1 | 2,27 | Rp 8.000.000,00 | Rp18.186.479,93 |
| <i>Acceptance & deployment</i> | Teknisi | 1 | 0,24 | Rp 8.000.000,00 | Rp1.923.569,99 |
| TOTAL | | | | | Rp24.730.989,65 |

Hasil estimasi biaya dengan alternatif skenario 2 dapat membuat harga fase pengembangan fitur lebih murah yaitu sebesar Rp24.730.989,65 dibandingkan dengan estimasi biaya sebelum diuji cobakan dengan skenario.

Seluruh skenario yang telah dibuat didokumentasikan seperti yang terlihat pada Tabel 6.45 Hasil rekap skenario Creativity Phase

Tabel 6.45 Hasil rekap skenario Creativity Phase

| No. | Kebutuhan Fungsional | Skenario alternatif kebutuhan fungsional | Kode Skenario |
|------------|-------------------------------|---|----------------------|
| 1 | Melakukan Update data pegawai | Meminimalkan EM | A-01 |
| | | Meminimalkan nilai SC | A-02 |
| | | Menambah SDM | A-03 |
| 2 | Input Aktivitas Pegawai | Meminimalkan EM | B-01 |
| | | Meminimalkan nilai SC | B-02 |
| | | Menambah SDM | B-03 |
| 3 | Mensetujui Aktivitas Pegawai | Meminimalkan EM | C-01 |
| | | Meminimalkan nilai SC | C-02 |
| | | Menambah SDM | C-03 |
| 4 | Membatalkan Aktivitas | Meminimalkan | D-01 |

| No. | Kebutuhan Fungsional | Skenario alternatif kebutuhan fungsional | Kode Skenario |
|-----|---|--|---------------|
| | Pegawai | EM | |
| | | Meminimalkan nilai SC | D-02 |
| | | Menambah SDM | D-03 |
| 5 | Menampilkan Informasi Penugasan Pegawai | Meminimalkan EM | E-01 |
| | | Meminimalkan nilai SC | E-02 |
| | | Menambah SDM | E-03 |
| 6 | Menampilkan Informasi Sasaran Program | Meminimalkan EM | F-01 |
| | | Meminimalkan nilai SC | F-02 |
| | | Menambah SDM | F-03 |
| 7 | Melakukan Penilaian | Meminimalkan EM | G-01 |
| | | Meminimalkan nilai SC | G-02 |
| | | Menambah SDM | G-03 |
| 8 | Melakukan Pencarian berdasar Filter | Meminimalkan EM | H-01 |
| | | Meminimalkan nilai SC | H-02 |
| | | Menambah SDM | H-03 |
| 9 | Mengenerate Rapor | Meminimalkan EM | I-01 |
| | | Meminimalkan nilai SC | I-02 |
| | | Menambah SDM | I-03 |

6.4. *Evaluation Phase*

Tahapan terakhir dari rangkaian tahapan pada penelitian ini adalah tahapan evaluasi. Tahapan evaluasi berisi mengenai analisis dari hasil perhitungan pada tahap sebelumnya dengan menggunakan teknik pencarian indeks nilai untuk pertimbangan apakah layak dilanjutkan pada fase berikutnya. Sebelum itu, peneliti telah merangkum hasil perhitungan setiap

skenario pada tahap sebelumnya untuk mempermudah menunjukkan perubahan biaya yang dihasilkan.

Tabel 6.46 Ringkasan perbandingan biaya dari skenario 1

| No | Kebutuhan Fungsional | Biaya Awal | Biaya Akhir |
|----|---|------------------|------------------|
| 1 | Melakukan Update data pegawai | Rp27.832.440,00 | Rp14.002.028,61 |
| 2 | Input Aktivitas Pegawai | Rp19.799.500,00 | Rp12.897.785,64 |
| 3 | Mensetujui Aktivitas Pegawai | Rp24.494.810,00 | Rp12.656.524,71 |
| 4 | Membatalkan Aktivitas Pegawai | Rp24.494.810,00 | Rp12.656.524,71 |
| 5 | Menampilkan Informasi Penugasan Pegawai | Rp24.721.090,00 | Rp12.897.785,64 |
| 6 | Menampilkan Informasi Sasaran Program | Rp23.023.990,00 | Rp11.880.626,66 |
| 7 | Melakukan Penilaian | Rp28.454.710,00 | Rp14.788.362,76 |
| 8 | Melakukan Pencarian berdasar Filter | Rp30.151.810,00 | Rp13.358.361,19 |
| 9 | Mengenerate Rapor | Rp25.739.350,00 | Rp15.680.181,46 |
| 10 | Menampilkan capaian aktivitas | Rp 20.648.050,00 | Rp 20.648.050,00 |
| 11 | Menampilkan capaian kegiatan | Rp20.648.050,00 | Rp 20.648.050,00 |
| 12 | Menampilkan indikator kinerja | Rp19.403.510,00 | Rp 19.403.510,00 |
| 13 | Login | Rp20.648.050,00 | Rp 20.648.050,00 |

| No | Kebutuhan Fungsional | Biaya Awal | Biaya Akhir |
|--------------|----------------------|-------------------------|-------------------------|
| 14 | Ganti Password | Rp18.498.390,00 | Rp 18.498.390,00 |
| Total | | Rp347.056.950,00 | Rp239.162.621,37 |

Tabel 6.47 Ringkasan perbandingan biaya dari skenario 2

| No | Kebutuhan Fungsional | Biaya Awal | Biaya Akhir |
|----|---|-----------------|-----------------|
| 1 | Melakukan Update data pegawai | Rp27.832.440,00 | Rp22.158.553,05 |
| 2 | Input Aktivitas Pegawai | Rp19.799.500,00 | Rp20.460.919,23 |
| 3 | Mensetujui Aktivitas Pegawai | Rp24.494.810,00 | Rp20.089.448,42 |
| 4 | Membatalkan Aktivitas Pegawai | Rp24.394.810,00 | Rp20.089.448,42 |
| 5 | Menampilkan Informasi Penugasan Pegawai | Rp24.721.090,00 | Rp20.460.919,23 |
| 6 | Menampilkan Informasi Sasaran Program | Rp23.023.990,00 | Rp18.893.346,28 |
| 7 | Melakukan Penilaian | Rp28.454.710,00 | Rp23.365.000,33 |
| 8 | Melakukan Pencarian berdasar Filter | Rp30.151.810,00 | Rp21.169.500,37 |
| 9 | Mengenerate Rapor | Rp25.739.350,00 | Rp24.730.989,65 |
| 10 | Menampilkan capaian aktivitas | Rp20.648.050,00 | Rp20.648.050,00 |

| No | Kebutuhan Fungsional | Biaya Awal | Biaya Akhir |
|--------------|-------------------------------|--------------------------|-------------------------|
| 11 | Menampilkan capaian kegiatan | Rp20.648.050,00 | Rp20.648.050,00 |
| 12 | Menampilkan indikator kinerja | Rp 19.403.510,00 | Rp19.403.510,00 |
| 13 | Login | Rp 20.648.050,00 | Rp20.648.050,00 |
| 14 | Ganti Password | Rp 18.498.390,00 | Rp 18.498.390,00 |
| Total | | Rp 347.056.950,00 | Rp309.762.564,97 |

Sesuai dengan skenario yang telah dilakukan, skenario dengan menggunakan pemanfaatan *Software design and development tool* memiliki penurunan harga yang signifikan dibandingkan dengan skenario penambahan SDM. Skenario dengan pemanfaatan *Software design and development tool* dapat menurunkan biaya sebesar 38% atau sebesar Rp 215.094.440 dari harga awal.

6.4.1. Pencarian Indeks Nilai

Berikut ini adalah teknik yang digunakan peneliti dalam menentukan apakah skenario tersebut layak dipertimbangkan atau tidak untuk dilanjutkan pada tahapan Selanjutnya. Penjelasan terkait persamaan yang digunakan dapat dilihat pada BAB 2 Dasar Teori.

Tabel 6.48 Hasil alternatif kebutuhan fungsional

| No | Alternatif Kebutuhan Fungsional | Hasil Pencarian Indeks | Keterangan |
|----|--|------------------------|-----------------------|
| 1 | Meminimalkan <i>Effort Multiplier</i> | 32% | Layak dipertimbangkan |
| 2 | Meninimalkan nilai <i>Scale Factor</i> | 12% | Layak dipertimbangkan |

6.4.2. Pemilihan Rekomendasi Alternatif

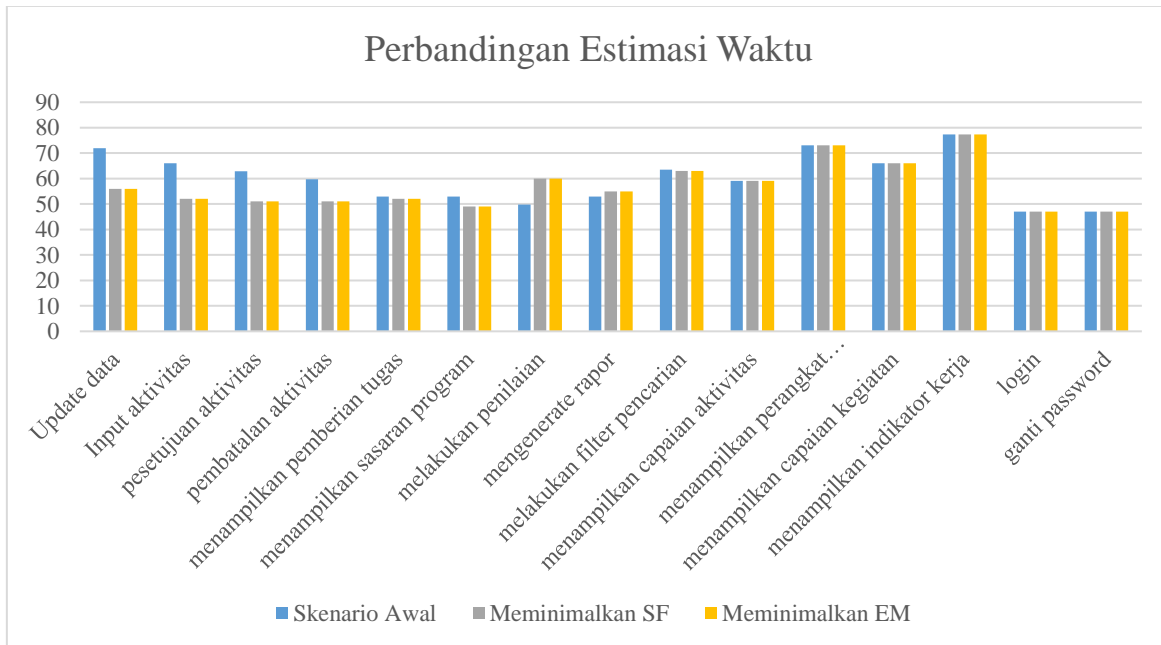
Setelah mengetahui skenario mana yang terpilih untuk dilanjutkan, kemudian dilakukan proses pemilihan rekomendasi dari hasil pencarian indeks. Mengacu pada prinsip pengerjaan proyek yang optimal, peneliti menentukan rekomendasi dari skenario yang menghasilkan waktu dan biaya pengerjaan aplikasi termurah dengan menggunakan bantuan grafik hubungan biaya dan waktu.

1. Perbandingan waktu pengerjaan

Dibawah ini adalah grafik yang menggambarkan perbandingan seluruh kebutuhan fitur dari segi waktu pengembangan setiap skenario yang dijalankan kepada setiap fitur.

Dari grafik tersebut ditemukan bahwa :

- Skenario diterapkan pada fitur diluar objek penelitian tidak mengalami perubahan waktu
- Total waktu pengembangan awal sebanyak 903 hari (terhitung baik fitur untuk objek penelitian atau tidak)
- Skenario Meminimalkan Effort Multiplier dan Scale Factor menghabiskan waktu pengembangan sebanyak 859 hari
- Skenario Menambah orang menghabiskan waktu sebanyak 677 hari



Grafik 6.1 Perubahan hari pengembangan setiap fitur

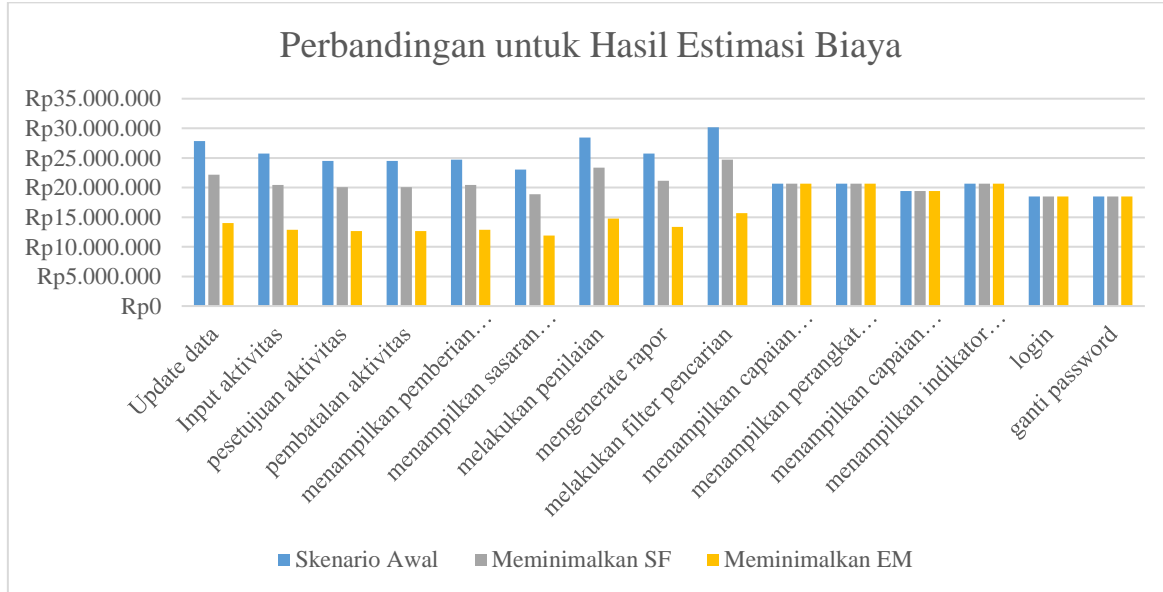
2. **Perbandingan biaya pengembangan**

Terdapat grafik yang menggambarkan perbandingan biaya yang dibutuhkan untuk membangun fitur dari hasil penerapan skenario serta harga normal pada **Grafik 6.2**

Dari grafik tersebut berisi terkait :

- Skenario diterapkan pada fitur diluar objek penelitian tidak mengalami perubahan biaya
- Total biaya untuk pengembangan seluruh fitur adalah sebesar Rp352.996.800
- Untuk estimasi pengembangan biaya dengan hasil paling murah adalah untuk alternatif skenario meminimalkan Effort Multiplier sebesar Rp239.162.621

Grafik 6.2 Perubahan biaya pengembangan setiap fitur



Grafik 6.3 Hubungan waktu dan biaya

Pada **Grafik 6.3** telah menunjukkan hubungan antara kedua skenario terhadap biaya dan waktu pengembangan. Dalam grafik tersebut didapat kesimpulan bahwa penerapan skenario meminimalkan nilai *effort multiplier* memiliki hasil paling rendah dibanding kedua skenario. Hal ini dibuktikan bahwa pada grafik, titik kuning yang menunjukkan skenario tersebut sangat dekat dengan koordinat 0, ketimbang titik lain.

6.4.3. Rekap perbandingan Estimasi Biaya

Pada sub bab ini, peneliti menampilkan semua hasil estimasi biaya berdasarkan dari usulan pihak ketiga, perhitungan keseluruhan dari metode COCOMO II serta perhitungan kebutuhan fungsional berdasarkan skenario terpilih.

Tabel 6.49 Perbandingan biaya yang dihasilkan tiap metode estimasi dan skenario

| <i>Metode</i> | <i>Pengalaman Pihak Ketiga</i> | | <i>COCOMO II</i> | | <i>Value Engineering</i> | |
|---------------|--|---|--|---|---|---|
| Biaya | Rp220.000.000 | | Rp198.741.503,62 | | Rp239.162.621 | |
| Keterangan | Harga berdasarkan Kerangka Acuan Kerja (KAK) | | Menghitung harga pengalaman pihak ketiga dengan <i>COCOMO II</i> | | Menerapkan skenario meminimalkan nilai <i>effort multiplier</i> | |
| Justifikasi | Kelebihan | Kelemahan | Kelebihan | Kelemahan | Kelebihan | Kelemahan |
| | | Aktivitas pengembangan tidak dijelaskan secara rinci pada KAK | Hasil perhitungan lebih murah ketimbang menggunakan value engineering karena hanya mempertimbangkan biaya yang dikeluarkan dari aktivitas proyek saja (tidak mempertimbangkan fitur) | Tidak memperhatikan berdasarkan fitur yang terdapat pada aplikasi | Standar gaji yang digunakan ditetapkan oleh <i>Sallary Guide 2018</i> | Memperhitungkan biaya setiap fitur, mengakibatkan total biaya sangat mahal karena terdapat perhitungan ulang dengan harga yang sama namun dari fitur yang berbeda |
| | | Standar harga yang ditetapkan oleh Tim Pengembang berdasarkan pada pengalaman | Standar gaji yang digunakan ditetapkan oleh <i>Sallary Guide 2018</i> | Pada penggalian data untuk <i>Effort Multiplier</i> dan <i>Scale Factor</i> berdasarkan pada pengalaman tim pengembang yang dapat | Estimasi biaya memperhitungkan kebutuhan fungsi/ fitur dari aplikasi yang akan dikembangkan serta memudahkan dalam mencari harga satu | Spesifikasi tim pengembang akan lebih tinggi dan berpengaruh kepada beban biaya pengembangan secara personnil |

| | | | | | | |
|--|--|--|--|---|---|--|
| | | | | mempengaruhi tingkat kemahalan dari biaya pengembangan aplikasi | fitur saja | |
| | | Pembagian aktivitas untuk setiap aktor yang terlibat dalam proyek tidak dijelaskan | | | Aktivitas pengembangan dijelaskan sesuai pada siklus fase pengembangan seperti <i>software development, on going activity</i> | |

6.4.4. Pembuatan *Value Engineering Change Proposal*

Berdasarkan hasil akhir yang peneliti dapatkan, maka peneliti membuat Value Engineering Change Proposal (VECP) (Lampiran B) sebagai bentuk rangkuman hasil penelitian peneliti untuk diserahkan kepada pihak ketiga, sehingga pihak ketiga dapat mengetahui dan menyetujui rekomendasi tersebut untuk kemudian diimplementasikan dalam pengajuan biaya baru pembangunan aplikasi XYZ.

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

Berikut ini adalah hasil temuan yang didapat peneliti :

1. Hasil estimasi biaya yang dihasilkan sebesar Rp 198.822.137,17 dengan pengerjaan aplikasi membutuhkan 9,61 bulan serta rata-rata pekerja yang dibutuhkan 3 orang.
2. Hasil Perhitungan terkait harga pengembangan/pembangunan setiap fitur pada aplikasi *e-Performance* ini (sesuai dengan beberapa syarat pemilihan objek penelitian seperti pembangunan fitur minimal membutuhkan waktu 600 jam keatas) mendapati 9 fitur yang terpilih sebagai objek penelitian dan sebagai objek yang dilakukan beberapa skenario penghematan. Objek yang dimaksud adalah sebagai berikut

Temuan yang penulis dapat terkait pembangunan fitur tersebut antara lain :

- a. Fitur yang paling banyak membutuhkan biaya adalah fitur Melakukan pencarian dengan filter sebesar Rp30.151.810 atau memakan biaya sekitar 12,2%, untuk fitur dengan harga termurah adalah fitur melakukan input aktivitas pegawai sebanyak R19.799.500 dan memakan banyak biaya sebesar 7.6%.
3. Berikutnya adalah temuan terkait skenario yang dibuat dalam rangka penghematan pembangunan fitur dengan membuat skenario terkait modifikasi nilai *Effort Multiplier* dan *Scale Factor* pada COCOMO II serta penambahan SDM. Dibawah ini adalah rincian terhadap temuan tersebut :

- Terdapat 18 skenario yang diuji coba untuk 9 fitur yang telah lolos FAST pada tahap *Function Analysis Phase*. Fitur yang lolos FAST juga dijadikan sebagai acuan untuk diterapkan skenario pada fase *Creativity Phase*.
- Hasil perhitungan skenario dengan menggunakan *Meminimalkan nilai Effort Multiplier* menghasilkan estimasi biaya dengan penghematan yang signifikan ketimbang skenario lain. Penghematan yang dihasilkan sebesar 32% dari harga awal dengan estimasi biaya sebesar Rp239.162.621
- Dengan meminimalkan nilai *Effort Multiplier* akan berdampak pada kemampuan tim pengembang. Berikut ini adalah rekomendasi yang diberikan kepada tim pengembang untuk memenuhi kriteria yang dirubah pada *Effort Multiplier* :
 - Dapat mencari anggota tim proyek dengan pengalaman pengembangan aplikasi selama minimal 3 tahun untuk meningkatkan nilai non-functional pada *Effort Multiplier* dengan kategori *Application Experience*

7.2. Saran

Berdasarkan atas temuan dan simpulan dari penelitian Tugas Akhir ini, berikut terkait saran yang diberikan khususnya untuk studi kasus Proyek Pembangunan Aplikasi *E-Performance*

1. Dalam perhitungan estimasi biaya untuk penelitian ini masih melibatkan beberapa asumsi karena pada perhitungan COCOMO II tidak dijelaskan secara spesifik antara perbedaan formula perhitungan durasi dengan pembagian pekerja pada tiap durasi. Untuk Selanjutnya dapat menemukan keterkaitan *effort* dan , SDM, durasi proyek pada perhitungan COCOMO II

Penelitian ini melibatkan asumsi :

- a. Penggunaan durasi kerja selama 8 jam tiap hari
 - b. Penggunaan durasi kerja dalam 1 bulan adalah 20 hari
 - c. Hasil perhitungan rerata pekerja dialokasikan sama pada setiap fase atau aktivitas dalam proyek
2. Karena penelitian ini menggunakan *Value Engineering* dengan studi kasus aplikasi *web-based* dan hasil yang tentu berbeda dengan penelitian sebelumnya yaitu aplikasi berbasis android.
 Penelitian Selanjutnya bisa menggunakan studi kasus dengan aplikasi yang mampu diakses di berbagai *device* untuk melihat perbandingan biaya dari studi kasus tersebut.
3. Penerapan *value engineering* berbasis kebutuhan fungsi/ fitur yang diperlukan aplikasi yang akan dibangun. Penelitian ini tidak membahas secara mendetail mengenai penggalian fitur pada fase *Function Analysis* untuk mengetahui secara detail elemen fitur hingga subfitur.
 Untuk penelitian Selanjutnya fase *value engineering* dapat dilakukan pendetailan serta penggalian yang mendalam untuk mengetahui elemen fitur yang dapat dioptimalkan dalam fase *Creativity*.
4. Penelitian ini sangat tergantung dengan dokumen KAK namun pembagian jobdesk tidak detail untuk setiap jabatan, contohnya untuk dibutuhkan 3 orang programmer namun tasknya tidak dijelaskan untuk setiap programmer yang dibutuhkan seperti apa.
 Penelitian Selanjutnya dapat membagi task yang ada pada jabatan untuk setiap orang yang terlibat pada jabatan tersebut untuk pengoptimalan biaya aktivitas proyek.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] A. F. Bilgies, “KEUNGGULAN KOMPETITIF YANG DALAM MENCIPTAKAN INOVASI UNTUK KEWIRAUSAHAAN STRATEGIS,” *An-Nisbah: Jurnal Ekonomi Syariah*, vol. 3, no. 2, Apr. 2017.
- [2] Supriyanto, “PEMBERDAYAAN TEKNOLOGI INFORMASI UNTUK KEUNGGULAN BISNIS,” *Jurnal Ekonomi & Pendidikan*, vol. 2, no. 1, p. 99, Aug. 2004.
- [3] T. S. Group, “The Standish Group Report Chaos,” p. 16, 2014.
- [4] F. Indriani, “STUDI MENGENAI ORIENTASI INOVASI, PENGEMBANGAN PRODUK DAN EFEKTIFITAS PROMOSI SEBAGAI SEBUAH STRATEGI UNTUK MENINGKATKAN KINERJA PRODUK,” p. 11.
- [5] F. S. L. Panjaitan, “PREPARATION OF ALTERNATIVE SOLUTIONS OF OPTIMAL FUNCTIONS IN ESTABLISHMENT PLANNING STAGE OF APPLICATION XYZ PROJECT WITH VALUE ENGINEERING,” p. 319, 2018.
- [6] A. A. Sholih, “AN OWNER ESTIMATE COST MODEL OF SOFTWARE PROJECT USING COCOMO II AS PERSONNEL DIRECT COST,” 2018.
- [7] O. Tailor, J. Saini, and M. P. Rijwani, “COMPARATIVE ANALYSIS OF SOFTWARE COST AND EFFORT ESTIMATION METHODS: A REVIEW,” p. 11, 2014.
- [8] “e-PERFORMANCE SISTEM AKUNTABILITAS KINERJA INSTANSI PEMERINTAH.” [Online]. Available: <http://eperformance.kemenpar.go.id/>. [Accessed: 04-Jan-2019].
- [9] “A review of value engineering as an effective system for planning building projects.” [Online]. Available: <https://www.pmi.org/learning/library/review-value-engineering-effective-system-5410>. [Accessed: 17-Dec-2018].

- [10] K. R. Thorson and R. Snidar, "IMPROVING PROFITS THROUGH PRIME/SUBCONTRACTOR VALUE ENGINEERING," 2011, p. 31.
- [11] B. Boehm, "COCOMO II Model Definition Manual," *University of South California*, vol. 4, no. 1, pp. 6–6.
- [12] Z. T. Abdulmehdi, M. S. S. Basha, M. Jameel, and P. Dhavachelvan, "A Variant of COCOMO II for Improved Software Effort Estimation," *International Journal of Computer and Electrical Engineering*, vol. 6, no. 4, pp. 346–350, 2014.
- [13] O. Moravcik, D. Petrik, T. Skripcak, and P. Schreiber, "Elements of the Modern Application Software Development," *International Journal of Computer Theory and Engineering*, pp. 891–896, 2012.
- [14] Tim Evaluasi Penyusunan dan Pengembangan Manajemen Kinerja Terpadu, "GAMBARAN UMUM APLIKASI EPERFORMANCE." .
- [15] "Function Analysis system Technique (FAST) - Canadian Society of Value Analysis." [Online]. Available: <http://www.valueanalysis.ca/fast.php>. [Accessed: 03-Jan-2019].
- [16] SAVE International, "SAVE International Value Standard and Body Knowledge," *SAVE International*.
- [17] "Value Analysis and Function Analysis System Technique," *Value Analysis and Function Analysis System Technique*, 14-Aug-2018. [Online]. Available: <http://www.npd-solutions.com/va.html>. [Accessed: 16-Jan-2019].
- [18] "What's the Real Value of Project Management?," *ProjectManager.com*, 29-Oct-2018. [Online]. Available: <https://www.projectmanager.com/training/value-of-project-management>. [Accessed: 16-Jan-2019].
- [19] D. Linman, "Value Management in Projects – Definition and Goals," *Value Management in Projects – Definition and Goals*, 11-Sep-2018. [Online]. Available: <https://mymanagementguide.com/value-management-in-projects-definition-and-goals/>. [Accessed: 31-Jan-2019].

- [20] Project Management Institute, Ed., *A guide to the project management body of knowledge (PMBOK guide)*, Fifth edition. Newtown Square, Pennsylvania: Project Management Institute, Inc, 2013.
- [21] “Project Manager,” *Project Manager*. .
- [22] D. M. Brandon, *Project Management for Modern Information Systems*: IGI Global, 2006.
- [23] J. Cadle and D. Yeates, *Project management for information systems*. Harlow: Financial Times, 2008.
- [24] B. Kho, “Pengertian Benchmarking (Tolok Ukur) dan Jenis-jenisnya,” *Ilmu Manajemen Industri*, 08-Jul-2016. .
- [25] J. Sodhi and P. Sodhi, “It Project Management Handbook,” *Carcinogenesis*, vol. 34, no. 10, p. 350, 2001.
- [26] D. B. Boehm, R. Valerdi, J. A. Lane, and A. W. Brown, “COCOMO Suite Methodology and Evolution,” p. 6, 2005.
- [27] B. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy, and R. Selby, “Cost models for future software life cycle processes: COCOMO 2.0,” *Annals of Software Engineering*, vol. 1, no. 1, pp. 57–94, Dec. 1995.
- [28] H. Tohidi, “Review the benefits of using value engineering in information technology project management,” *Procedia Computer Science*, vol. 3, pp. 917–924, 2011.
- [29] “FPA - Function Point Analysis.” [Online]. Available: <https://www.functionpoints.org/function-point-analysis.html>. [Accessed: 11-Jan-2019].
- [30] Liming Wu, “The Comparison of the Software Cost Estimating Methods,” *Computing Decue*, 04-Mar-1997. [Online]. Available: <https://www.computing.dcu.ie/~renaat/ca421/LWu1.htm> l. [Accessed: 01-Oct-2019].
- [31] “Disadvantages of function point analysis - Project management (PMP).” [Online]. Available: <https://www.careerride.com/pmp-disadvantages-of-function-point-analysis.aspx>. [Accessed: 11-Jan-2019].
- [32] “Value Analysis: Meaning, Types and Procedure,” *Learn Accounting: Notes, Procedures, Problems and Solutions*,

- 18-Jun-2016. [Online]. Available: <http://www.accountingnotes.net/cost-accounting/value-analysis/value-analysis-meaning-types-and-procedure/6427>. [Accessed: 16-Jan-2019].
- [33] R. E. Cole and Y. Nakata, "The Japanese Software Industry: What Went Wrong and What Can We Learn from it?," *California Management Review*, vol. 57, no. 1, pp. 16–43, Nov. 2014.
- [34] H. Sabrang, "Value Engineering," *Universitas Atma Jaya Yogyakarta*, 1998.
- [35] A. F. Osborn, *Applied imagination principles and procedures of creative thinking*, Rev. ed. New York Scribner, 1957.
- [36] R. Rachwan, I. Abotaleb, and M. Elgazouli, "The Influence of Value Engineering and Sustainability Considerations on the Project Value," *Procedia Environmental Sciences*, vol. 34, pp. 431–438, 2016.
- [37] K. R. Thorson and R. Snidar, "Improving Profits Through Prime / Subcontractor Value Engineering," no. June 1984, 2011.
- [38] SAVE International, "Value methodology standard and body of knowledge," *The Value Society*, no. June, pp. 1–33, 2007.
- [39] D. Chua, "Value Improvement Methods," in *The Civil Engineering Handbook, Second Edition*, vol. 20021369, W. Chen and J. Richard Liew, Eds. CRC Press, 2002.
- [40] A. E.W, "Value Engineering," *Value Engineering Department of the Army Corps of Engineers USA*, 1995.
- [41] IFPUG, "Function Point Modeler Workbench," *Function Point Modeler Workbench*. [Online]. Available: <http://www.functionpointmodeler.com/fpm-infocenter/index.jsp?topic=%2Fcom.functionpointmodeler.fpm.help%2Fditafiles%2Fconcepts%2Fcon-115.html>. [Accessed: 30-Apr-2019].
- [42] F. P. Laksamana, A. R. Perdanakusuma, and M. C. Saputra, "Evaluasi Biaya Pengembangan Sistem Informasi

- Pengelolaan Arsip Surat (SIPAS) Menggunakan Function Point dan Object Point (Studi Kasus : PT Sekawan Media Informatika),” p. 10.
- [43] M. N. Khoiro, A. D. Herlambang, and M. C. Saputra, “Evaluasi Biaya Pengembangan Perangkat Lunak Dengan Menggunakan Metode Cocomo II (Studi Kasus: PT DOT Indonesia),” p. 10.
- [44] P. L. P. A. Sholih, “Effort Distribution to Estimate Cost in Small to Medium Software Development Project with Use Case Points,” *Procedia Computer Science*, vol. 72, pp. 78–85.
- [45] “How long does it take to build a website?,” *Bill Erickson*, 22-Feb-2016. [Online]. Available: <https://www.billerickson.net/how-long-does-it-take-to-build-a-website/>. [Accessed: 24-Jun-2019].
- [46] L. L. s Herdiman, E. Liquiddanu, and D. Paramita, “PERBAIKAN RANCANGAN PADA DESAIN KNEE ANKLE FOOT ORTHOSIS (KAFO) DENGAN PENDEKATAN METODE FUNCTION ANALYSIS SYSTEM TECHNIQUE,” *J@ti Undip : Jurnal Teknik Industri*, vol. 6, no. 3, pp. 189–198, 2011.
- [47] A. L, “Function Point Analysis FPA on A Team Planning Website Based on PHP and MYSQL,” *Journal of Information Technology & Software Engineering*, vol. 08, no. 03, 2018.

Halaman ini sengaja dikosongkan

BIODATA PENELITI



Penulis lahir di kota kecil Tulungagung pada 26 Maret 1997. Terlahir menjadi anak pertama dari dua bersaudara. Penulis telah menempuh pendidikan formal SDN Kampung Dalem 1, SMP Negeri 1 Tulungagung, SMA Negegeri 1 Kedungwaru.

Tahun 2015, penulis melanjutkan pendidikan kejenjang perguruan tinggi, diterima melalui jalur SNMPTN di Institut Teknologi Sepuluh Nopember, Surabaya mengambil prodi S1 Sistem In-

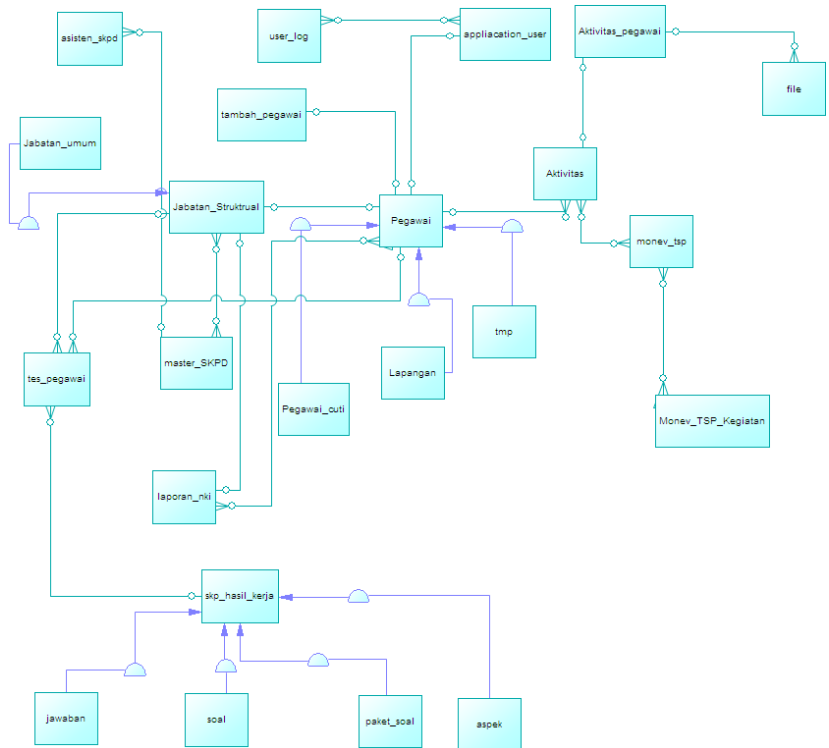
formasi. Selama proses perkuliahaan, Penulis aktif dalam organisasi mahasiswa seperti menjadi Kepala Divisi Akademik departemen *Student Welfare* pada kepengurusan HMSI Evolve 2017-2018, selain itu Penulis aktif dalam kegiatan kepanitian seperti menjadi Staff ahli pada kegiatan ISE pada divisi *Creative Decoration* tahun 2017, terlibat dalam pelaksanaan *Job Fair ITS 37* sebagai staff Administrasi, dan mengikuti kegiatan di luar kampus sebagai *Junior Designer* pada sebuah agensi desain Positive Thirteen.

Selama tahun keempat perkuliahaan, Penulis berfokus pada penelitian terkait manajemen proyek dan *Value Engineering* yang membuat penulis memilih mengambil bidang minat Manajemen Sistem Informasi dan terdaftar sebagai mahasiswa tugas akhir pada laboratorion *Information System Management*, Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember. Terkait pertanyaan atau kritik dan masukan untuk penelitian, penulis dapat dihubungi melalui *email* : safiralism54@gmail.com .

Halaman ini sengaja dikosongkan

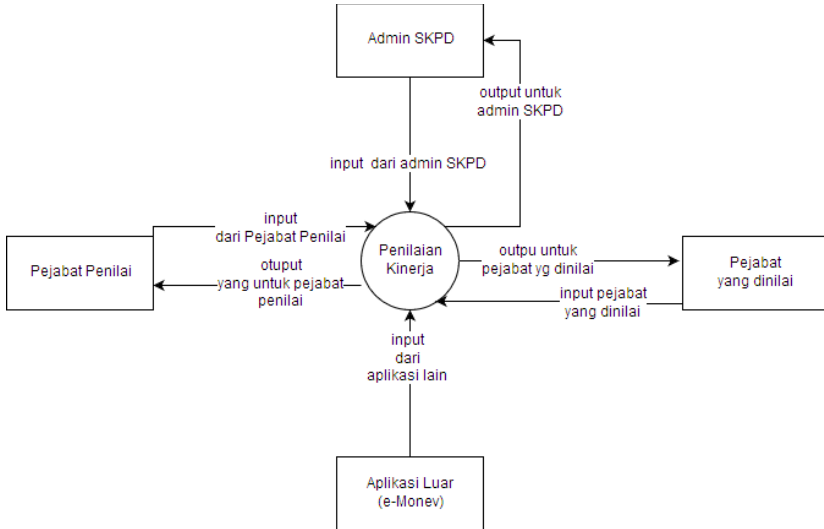
LAMPIRAN A

Lampiran ini berisi *Data Flow Diagram* & Konseptual Data Model untuk aplikasi *E-Performance*. Diagram tersebut dibuat untuk menjelaskan keterkaitan masing-masing komponen (data, proses) untuk fungsi yang ada pada aplikasi serta untuk menghitung ukuran (*size*) pada aplikasi.



Gambar A. 1 Konseptual Data Model untuk Aplikasi e-Performance

Dibawah ini adalah hasil *Data Flow Diagram* pada *context level* dan Level 1 mengenai aliran data serta proses yang terdapat pada aplikasi *e-Performance*



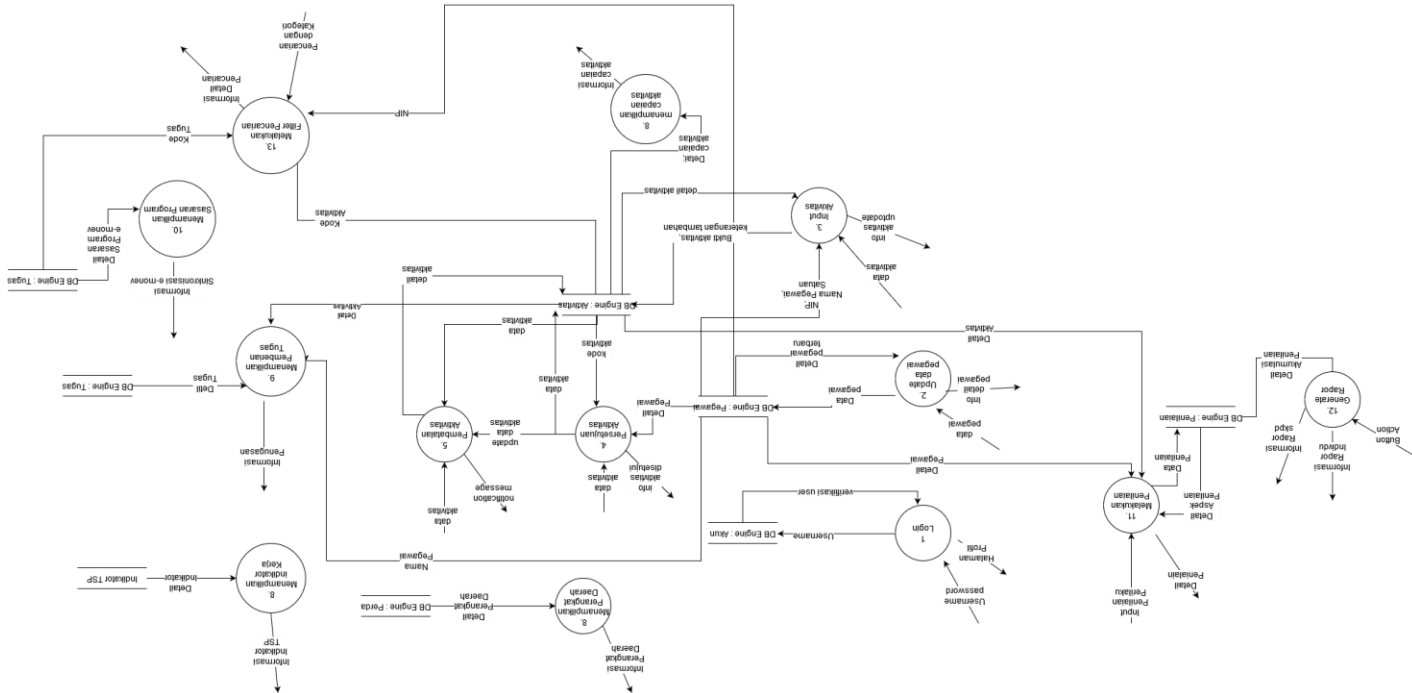
Gambar A. 2 Data Flow Diagram Context Level aplikasi e-Performance

Pada tabel berikut dilampirkan aliran data apa saja yang menjadi *input* dan *output* dari masing masing entitas terhadap proses penilaian kinerja

Tabel A. 1 Daftar data flow input dan output pada DFD level konteks e-Performance

| <i>Data Flow</i> pada level konteks | |
|-------------------------------------|---|
| <i>Input</i> dari Pejabat Penilai | |
| 1. | Pemberian tugas kepada pejabat yang dinilai |
| 2. | Memasukkan aktivitas |
| 3. | Melakukan penilaian |
| 4. | Verifikas aktivitas |
| 5. | Pembatalan Aktivitas |
| 6. | Penilaian pejabat |
| 7. | Pencarian berdasarkan filter |

| <i>Data Flow</i> pada level konteks | |
|---|---|
| <i>Ouput</i> dari Pejabat Penilai | |
| 1. | Informasi detail tugas setiap jabatan |
| 2. | Informasi aktivitas |
| 3. | Informasi capaian aktivitas |
| 4. | Notifikasi verifikasi aktivitas |
| 5. | Notifikasi pembatalan aktivitas |
| 6. | Laporan penilaian pejabat |
| 7. | Informasi checklist penilaian |
| 8. | Hasil pencarian berdasarkan filter |
| <i>Input</i> dari pejabat yang dinilai | |
| 1. | Memasukkan Aktivitas |
| 2. | Pencarian berdasarkan filter |
| <i>Output</i> dari pejabat yang dinilai | |
| 1. | Informasi detail aktivitas |
| 2. | Hasil pencarian berdasarkan filter |
| 3. | Informasi checklist penilaian |
| <i>Input</i> admin SKPD | |
| 1. | Memasukkan data pegawai |
| 2. | Memperbarui data pegawai |
| 3. | Menginputkan nilai pegawai |
| 4. | Melakukan generate nilai pegawai |
| 5. | Melakukan penarikan data dari aplikasi lain |
| <i>Output</i> admin SKPD | |
| 1. | Informasi detail data pegawai |
| 2. | Hasil nilai/ laporan pegawai |
| 3. | Capaian aktivitas |
| 4. | Sasaran Program dan Indikator Penilaian |
| <i>Input</i> Aplikasi lain | |
| 1. | Sasaran Program |
| 2. | Indikator Penilaian |
| 3. | Sasaran Kegiatan |



Gambar A. 3 Data Flow Diagram Level 1 e-Performance

LAMPIRAN B

Lampiran ini berisi hasil akhir dari implementasi *Value Engineering* yaitu *Value Engineering Change Proposal*


PROJECT INFORMATION :

| | | | |
|-------------|--|------------------------|---------------------|
| COMPANY Do | : - | LETING DATE | : /06/2019 |
| DESCRIPTION | : Replacement with Effort Multiplier value minimalize | ORIGINAL PROJECT COSTS | : Rp 220.000.000,00 |
| COMPANY | : PT.XYZ | COUNTRY | : Indonesia |

VECP INFORMATION :

SHORT DESCRIPTION OF VECP : Replacement of programming tools with Software design and development tool 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? : | |  | HOW MANY? : 1 |
| | | YES | NO |
| (Note : Describe items A. though D in further detail in >COMMENTS= as appropriate) | | | |
| (E) PROJECT COST ORIGINAL | Rp220.000.000,00 | (Note :This is the original costs estimation by PT.XYZ) | |
| (F) PROJECT COST W/O VECP | Rp352.996.800,00 | (Note : This costs are totally count from 15 functions which wants to build on this app) | |
| (G) PROJECT COST W VECP | Rp239.162.621,00 | (Note : Count by implementing the fourth scenario – using Software design and development tool) | |
| (H) VECP PROJECT SAVINGS | Rp136.452.360,00 | (Note : Equals item F. minus item G.) | |
| (I) VECP DESIGN COST | - | | |
| (J) DIRECT COST SAVINGS | Rp113.834.179,00 | (Note : Equals item H. minus item I.) | |

COMMENT :

1. For the submission of the VECP document is left because the author's scope only communicates the document to a third party
 2. The original project cost has been predetermined by third parties. These cost estimates focus directly on developing one application. On project costs without VECP determined by the author focusing on application features, this has resulted in higher costs than the project cost from third parties
-

-
3. Estimated project cost with VECP are obtained from the results of using minimalize effort multiplier value, which it can be lost more effort because of the team capability more than complicated beferor uses that scenario
-
4. Cost savings resulting from estimated costs before VECP are reduced by estimated costs of using VECP.
-

VECP – ITEMIZED PROJECT SAVINGS

VECP DESCRIPTION : Software Design and Development tools Scenario

| NO. | FUNGSIONAL REQUIREMENT | INITIAL COST (s) | FINAL COST (s) | CHANGE IN FUND (S) | |
|-----|---|------------------|----------------|--------------------|----------|
| | | | | DECREASE | INCREASE |
| 1. | Melakukan Update data pegawai | Rp 27.832.440 | Rp 14.002.029 | Rp13.830.411 | - |
| 2. | Input Aktivitas Pegawai | Rp 19.799.500 | Rp 12.897.786 | Rp12.841.564 | - |
| 3. | Mensetujui Aktivitas Pegawai | Rp 24.494.810 | Rp 12.656.525 | Rp11.838.285 | - |
| 4. | Membatalkan Aktivitas Pegawai | Rp 24.494.810 | Rp 12.656.525 | Rp11.838.285 | - |
| 5. | Menampilkan Informasi Penugasan Pegawai | Rp 24.721.090 | Rp 12.897.786 | Rp11.823.304 | - |

| NO. | FUNGSIONAL REQUIREMENT | INITIAL COST (s) | FINAL COST (s) | CHANGE IN FUND (S) | |
|-----|---------------------------------------|------------------|----------------|--------------------|----------|
| | | | | DECREASE | INCREASE |
| 6. | Menampilkan Informasi Sasaran Program | Rp 23.023.990 | Rp 11.880.627 | Rp11.143.363 | - |
| 7. | Melakukan Penilaian | Rp 28.454.710 | Rp 14.788.363 | Rp13.666.347 | - |
| 8. | Melakukan Pencarian berdasar Filter | Rp 30.151.810 | Rp 13.358.361 | Rp12.380.989 | - |
| 9. | Mengenerate Rapor | Rp 25.739.350 | Rp 15.680.181 | Rp14.471.629 | - |
| 10. | Menampilkan capaian aktivitas | Rp 20.648.050 | Rp 20.648.050 | - | - |
| 11. | Menampilkan capaian kegiatan | Rp 20.648.050 | Rp 20.648.050 | - | - |
| 12. | Menampilkan indikator kinerja | Rp 19.403.510 | Rp 19.403.510 | - | - |
| 13. | Login | Rp 20.648.050 | Rp 20.648.050 | - | - |
| 14. | Ganti Password | Rp 18.498.390 | Rp 18.498.390 | - | - |
| 15. | Menampilkan infor- | Rp 20.648.050 | Rp 18.498.390 | | |

| NO. | FUNGSIONAL REQUIREMENT | INITIAL COST (s) | FINAL COST (s) | CHANGE IN FUND (S) | |
|-----|------------------------|-----------------------|-----------------------|-----------------------|----------|
| | | | | DECREASE | INCREASE |
| | masi perangkat daerah | | | | |
| | TOTAL | Rp 347.056.950 | Rp 239.162.621 | Rp 113.834.179 | - |

halaman ini sengaja dikosongkan

LAMPIRAN C

Dibawah ini adalah jenis pertanyaan yang terdapat pada koesioner penilaian *Cost driver* untuk mengetahui nilai *Scale Factor*, dan *Effort Multiplier* pada pembangunan aplikasi

Lembar Penilaian – *Effort Multiplier* COCOMO II untuk proyek pembangunan Aplikasi E-Performance

Lembar penilaian ini adalah salah satu alat yang digunakan untuk peneliti mendapatkan beberapa faktor yang digunakan dalam perhitungan estimasi biaya, waktu, dan sumber daya pengerjaan pengerjaan Sistem Informasi E-Performance yang dilakukan dalam tahap perencanaan proyek. Data yang dihasilkan lembar penilaian ini akan digunakan sebagai bahan penelitian Tugas Akhir. Kami sangat mengharapkan kerjasama Bapak/ Ibu/ Saudara/I untuk mengisi lembar penilaian dengan jujur.

IDENTITAS KORESPONDEN :

Nama :

Peran dalam proyek : **Project Manager / Sistem Analis / Programmer / Lainnya :**

Alamat :

Nomor Telepon :

Pendidikan :

Jenis Kelamin :

Email :

Pengalaman Proyek : **> 1 / < 1 / Lainnya :**

Tabel C. 1 Daftar pertanyaan penilaian Cost Driver COCOMO II

| No | Kategori Scale Factor | Pertanyaan | Skala Jawaban | | | | |
|----|------------------------|---|------------------|------------|---------------|-------------|-------------------|
| | | | VL (Very Low) | L (Low) | N (Nomina) | H (High) | VH (Very High) |
| 1 | <i>Precedentedness</i> | <p>Seberapa sering tim pengembang menyelesaikan proyek pembangunan aplikasi dalam kurun waktu tertentu</p> <p>Ketentuan Penilaian :</p> <ul style="list-style-type: none"> • Very Low : Jika tim pengembang belum pernah sama sekali melakukan pemabngunan perangkat lunak aplikasi • Low : Jika tim pengembang telah melaksanakan proyek tersebut sebanyak 1 - 2 kali dalam kurun waktu tertentu • Nominal : Jika tim | | | | | |

| No | Kategori <i>Scale Factor</i> | Pertanyaan | Skala Jawaban | | | | |
|----|------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>pengembang telah melaksanakan proyek tersebut sebanyak 3-4 kali dalam kurun waktu tertentu</p> <ul style="list-style-type: none"> • High : Jika tim pengembang sangat familiar dalam penyelesaian proyek tersebut dan dengan melaksanakan lebih dari 5 kali dalam kurun waktu tertentu • Very High : Jika tim pengembang sangat sering melaksanakan proyek tersebut dengan intensitas melakukan diatas 10 kali dalam | | | | | |

| No | Kategori <i>Scale Factor</i> | Pertanyaan | Skala Jawaban | | | | |
|----|---------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | kurun waktu tertentu | | | | | |
| 2 | <i>Development Flexsibility</i> | <p>Kualitas keleluasaan/ fleksibilitas klien/ calon user dalam menentukann kebutuhan perangkat lunak pada tim pengembang</p> <p>Ketentuan Penilaian :</p> <ul style="list-style-type: none"> • Low : Jika klien telah mendefinisikan semua kebutuhan software sebelumnya • Nominal - High : Jika klien telah mendefinisikan hanya sebagian dari kebutuhan software sebelumnya • Very high : Jika klien | | | | | |

| No | Kategori <i>Scale Factor</i> | Pertanyaan | Skala Jawaban | | | | |
|----|------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | hanya mampu mendefinisikan fungsi dasar software sebelumnya | | | | | |
| 3 | <i>Risk Resolution</i> | <p>Tingkat perencanaan dan pelaksanaan manajemen risiko yang dialami selama proses penyelesaian perangkat lunak</p> <p>Ketentuan Penilaian :</p> <ul style="list-style-type: none"> Very Low : Tim pengembang TIDAK melakukan perencanaan Manajemen Risiko untuk pengembangan perangkat lunak SERTA TIDAK | | | | | |

| No | Kategori <i>Scale Factor</i> | Pertanyaan | Skala Jawaban | | | | |
|----|------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>menentukann milestone sesuai Life Cycle Architecture (LCA) dan Product Design Review (PDR)</p> <ul style="list-style-type: none"> • Low : Jika Tim Pengembang melakukan SEDIKIT perencanaan manajemen risiko (sebesar 10%) • Nominal : Jika perencanaan manajemen risiko dilakukan SEBAGIAN (sebesar 50%) • High : Jika perencanaan risiko dilakukan secara | | | | | |

| No | Kategori <i>Scale Factor</i> | Pertanyaan | Skala Jawaban | | | | |
|----|------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>umum</p> <ul style="list-style-type: none"> • Very High : Jika perencanaan manajemen risiko dilakukan secara penuh dan spesifik untuk semua item yang berisiko | | | | | |
| 4 | <i>Team Cohesion</i> | <p>Kemampuan, kemauan Tim Pengembang dalam melakukan akomodasi dan kerjasama terhadap Stakeholder terkait</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Very Low : kemampuan Tim pengembang dalam akomodasi dan kerjasama SANGAT | | | | | |

| No | Kategori <i>Scale Factor</i> | Pertanyaan | Skala Jawaban | | | | |
|----|------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>RENDAH</p> <ul style="list-style-type: none"> • Low : kemampuan Tim pengembang dalam akomodasi dan kerjasama RENDAH • Nominal : kemampuan Tim pengembang dalam akomodasi dan kerjasama NORMAL/ RATA RATA • High : kemampuan Tim pengembang dalam akomodasi dan kerjasama TINGGI • Very High : kemampuan Tim pengembang dalam akomodasi dan kerjasama SANGAT | | | | | |

| No | Kategori <i>Scale Factor</i> | Pertanyaan | Skala Jawaban | | | | |
|----|------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | TINGGI | | | | | |
| 5 | <i>Process Maturity</i> | <p>Penggunaan dokumen Software Requirement untuk mengontrol pembuatan perangkat lunak sesuai dengan baseline/ jadwal yang ditentukan</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Very Low : Pengembangan aplikasi TIDAK berdasarkan Software Requirement • Low : Pengembangan aplikasi hanya melaksanakan 10% dari Software Requirement • Nominal : | | | | | |

| No | Kategori <i>Scale Factor</i> | Pertanyaan | Skala Jawaban | | | | |
|----|------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>Pengembangan aplikasi hanya melaksanakan 40% dari Software Requirement</p> <ul style="list-style-type: none"> • High : Pengembangan aplikasi hanya melaksanakan 40-60% dari Software Requirement • Very High : Pengembangan aplikasi melaksanakan 60-90% dari Software Requirement | | | | | |
| 6 | | <p>Kegiatan dan komitmen proyek perangkat lunak direncanakan dan didokumentasikan</p> <p>Kriteria Penilaian :</p> | | | | | |

| No | Kategori <i>Scale Factor</i> | Pertanyaan | Skala Jawaban | | | | |
|----|------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <ul style="list-style-type: none"> • Very Low : Tim Pengembang TIDAK PERNAH melakukan dokumentasi terkait kegiatan proyek • Low : Tim Pengembang JARANG melakukan dokumentasi terkait kegiatan proyek (prosentase dokumentasi : dibawah 40%) • Nominal : Tim Pengembang RATA-RATA melakukan dokumentasi terkait kegiatan proyek (prosentase | | | | | |

| No | Kategori <i>Scale Factor</i> | Pertanyaan | Skala Jawaban | | | | |
|----|------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>dokumentasi : 40-60%)</p> <ul style="list-style-type: none"> • High : Tim Pengembang SERING melakukan dokumentasi terkait kegiatan proyek (prosentase dokumentasi : 60%) • Very High : Tim Pengembang SELALU melakukan dokumentasi terkait kegiatan proyek (prosentase dokumentasi : 60-90%) | | | | | |

Lembar Penilaian – *Scale Factori* COCOMO II untuk proyek pembangunan Aplikasi E-Performance

Lembar penilaian ini adalah salah satu alat yang digunakan untuk peneliti mendapatkan beberapa faktor yang digunakan dalam perhitungan estimasi biaya, waktu, dan sumber daya pengerjaan pengerjaan Sistem Informasi E-Performance yang dilakukan dalam tahap perencanaan proyek. Data yang dihasilkan lembar penilaian ini akan digunakan sebagai bahan penelitian Tugas Akhir. Kami sangat mengharapkan kerjasama Bapak/ Ibu/ Saudara/I untuk mengisi lembar penilaian dengan jujur.

IDENTITAS KORESPONDEN :

- Nama** :
- Peran dalam proyek** : **Project Manager / Sistem Analis / Programmer / Lainnya :**
- Alamat** :
- Nomor Telepon** :
- Pendidikan** :
- Jenis Kelamin** :
- Email** :
- Pengalaman Proyek** : **> 1 / < 1 / Lainnya :**

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|--|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| 1 | <i>Developed for Re-usability (RUSE)</i> | <p>Peran dokumentasi selama pengembangan proyek digunakan kembali untuk proyek serupa dimasa depan</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Low : dokumentasi TIDAK digunakan dalam proyek, program, dan produk lain • Nominal - High : dokumentasi AKAN digunakan dalam proyek lain (1-2 Proyek serupa Selanjutnya) • Very High : | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | dokumentasi SELALU digunakan dalam produk lain yang serupa | | | | | |
| 2 | <i>Personnel Capability</i> | <p>Kemampuan personel dalam analisis dan desain, efisiensi kemampuan komunikasi dan kerjasama, serta banyak proyek pengembangan aplikasi yang telah diikuti</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Very Low : 2 kali mengikuti kegiatan analisis selama pengerjaan 10 proyek terakhir • Low : 4 kali | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>mengikuti kegiatan analisis selama pengerjaan 10</p> <ul style="list-style-type: none"> • Nominal : 6 kali mengikuti kegiatan analisis selama pengerjaan 10 • High : 8 kali mengikuti kegiatan analisis selama pengerjaan 10 • Very High : 9 kali mengikuti kegiatan analisis selama pengerjaan 10 | | | | | |
| 3 | | Berapa kali Tim Pengembang (Programmer) mengikuti proyek pengembangan | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>perangkat lunak</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Very Low : 2 kali berpartisipasi dalam 10 proyek terakhir • Low : 4 kali berpartisipasi dalam 10 kali proyek terakhir • Nominal : 6 kali berpartisipasi dalam 10 proyek terakhir | | | | | |
| 4 | | <p>Tingkat pergantian personel tiap tahun (annual personel turnover)</p> <p>Kriteria Penilaian :</p> | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <ul style="list-style-type: none"> • Very Low : 48%/ tahun • Low : 24%/ tahun • Nominal : 12%/ tahun • High : 6%/ tahun • Very High : 3%/ tahun | | | | | |
| 5 | <i>Product Reliability(RCPX)</i> | <p>Pada aplikasi serupa, seberapa besar kemampuan aplikasi dalam menjalankan fungsinya pada periode waktu tertentu</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Very Low : kegagalan aplikasi hanya berdampak | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>pada kenyamanan pengguna</p> <ul style="list-style-type: none"> • Low : kegagalan aplikasi berdampak RENDAH mengakibatkan SEDIKIT kerugian finansial • Nominal : kegagalan aplikasi berdampak SEDANG pada kerugian finansial • High : kegagalan aplikasi berdampak BERAT dan mengakibatkan kerugian fatal pada finansial | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <ul style="list-style-type: none"> • Very High : kegagalan aplikasi berdampak pada nyawa/ kehidupan pengguna | | | | | |
| 6 | | <p>Untuk aplikasi serupa , ukuran database yang digunakan. Dapat menggunakan perhitungan D/P (database(bytes)/ Program)</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Low : D/P SLOC < 10 • Nominal : $10 \leq D/P \leq 100$ • High : $100 \leq D/P \leq 1000$ | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <ul style="list-style-type: none"> • Very High : D/P \geq 1000 | | | | | |
| 7 | | <p>Kesesuaian platform, OS, bahasa pemrograman, manajemen database dalam menjalankan aplikasi</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Very Low : ketidaksesuaian menyebabkan aplikasi tidak bisa berjalan sama sekali (down service) • Low : ketidaksesuaian menyebabkan aplikasi mengalami eror | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <ul style="list-style-type: none"> Nominal : aplikasi berjalan sesuai fungsi (bug : 10%) High : aplikasi berjalan tanpa mendapat eror/ bug Very High : pada saat pengujian aplikasi tidak ditemukan eror | | | | | |
| | | <p>Kesesuaian dokumentasi pada proyek sesuai dengan siklus hidup pengembangan perangkat lunak</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> Very Low : sedikit dokumentasi yang | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>dilakukan (hanya mencakup pada 1 fase siklus hidup)</p> <ul style="list-style-type: none"> • Low : dokumentasi dilakukan pada beberapa proses siklus hidup pengembangan aplikasi • Nominal : dokumentasi dilakukan diseluruh siklus hidup pengembangan aplikasi • High : dokumentasi pengembangan aplikasi dilakukan | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | secara berlebihan | | | | | |
| 8 | <i>Platform Difficulty (PDIF)</i> | <p>Prosentasi kendala / hambatan terkait penyelesaian proyek pengembangan perangkat lunak yang diharapkan :</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Very Low : 5% dari waktu yang diharapkan • Low : 25% dari waktu yang diharapkan • Nominal : 50% dari waktu yang diharapkan | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <ul style="list-style-type: none"> • High : 80% dari waktu yang diharapkan • Very High : 95% dari waktu yang diharapkan | | | | | |
| | | <p>Presentase KETERSEDIAAN TEMPAT DALAM proses penyimpanan data selama pengembangan perangkat lunak berlangsung :</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Nominal : $\leq 50\%$ ketersediaan memori • High : 70% ketersediaan memori | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <ul style="list-style-type: none"> • Very High : 85% ketersediaan memori | | | | | |
| | | <p>Perubahan pada hardware dan software (OS, DBMS) pada kurun waktu tertentu</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Low : perubahan besar 12 bulan, perubahan kecil 1 bulan • Nominal : perubahan besar 6 bulan, perubahan kecil 2 minggu • High : perubahan besar 2 bulan, perubahan kecil | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>dialami 1 minggu</p> <ul style="list-style-type: none"> • Very High : perubahan besar 2 minggu, perubahan kecil dialami setiap hari | | | | | |
| 9 | <i>Personnel Experience(PREX)</i> | <p>Pengalaman durasi/ waktu yang didapat tim dalam menyelesaikan proyek pengembangan aplikasi serupa</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Very Low : 2 bulan • Low : 6 bulan • Nominal : 1 tahun • High : 3 tahun • Very High : 6 tahun | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>Pemahaman tim dalam penggunaan platform, data-base jaringan & middleware</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Very Low : 2 bulan • Low : 6 bulan • Nominal 1 tahun • High : 3 tahun • Very High : 6 tahun | | | | | |
| | | <p>Pengalaman tim pengembang dalam bahasa pemrograman yang digunakan, termasuk alat- alat yang digunakan selama proses siklus hidup pengembangan perangkat lunak</p> | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | Kriteria Penilaian : <ul style="list-style-type: none"> • Very Low : 2 bulan • Low : 6 bulan • Nominal : 1 tahun • High : 3 tahun • Very High : 6 tahun | | | | | |
| 10 | <i>Facilities(FCIL)</i> | Pemanfaatan Tools untuk mengatasi permasalahan dalam pengembangan perangkat lunak Kriteria Penilaian : <ul style="list-style-type: none"> • Low : Penggunaan Tools sangat minimal. Seperti dukungan mengedit kode, memperbaiki bug menggunakan | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|---|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>tools sederhana, berkomunikasi via telepon/ email saja</p> <ul style="list-style-type: none"> • Nominal : penggunaan tools selama siklus hidup pengembangan perangkat lunak berlangsung • High : Tools memberi dukungan secara kuat pada M/S Development, tools teriegrasi dengan baik, komunikasi dilakukan melalu wide band communcation | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | (smartphone dll) <ul style="list-style-type: none"> • Very High : Tools memberi dukungan secara kuat pada simple M/S Development, tools teriegrasi dengan baik, komunikasi dilakukan melalu wide band communcation, confrence | | | | | |
| | | Mengenai cara distribusi yang digunakan dalam pengembangan perangkat lunak. Apakah komunikasi dilakukan masih menggunakan telepon, email, | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | fax, teleconference Kriteria Penilaian : <ul style="list-style-type: none"> • Very Low : komunikasi via telepon, email • Low : Komunikasi telepon, fax • Nominal : melalui email pribadi • High : komunikasi melalui wide band communciation, video conferencee • Very High : komunikasi dengan Interactive Media | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|--|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| 11 | <i>Required Development Schedule(SCED)</i> | <p>Mengenai kendala jadwal yang dikenakan pada tim proyek pengembangan perangkat lunak (pada proyek pengembangan sebelumnya)</p> <p>Kriteria Penilaian :</p> <ul style="list-style-type: none"> • Very Low : kompresi 75% dari jadwal yang telah ditentukan • Low : kompresi 85% dari jadwal yang telah ditentukan • Nominal : kompresi 100% dari jadwal yang telah ditentukan • High : kompresi 130% dari jadwal | | | | | |

| No | Kategori <i>Effort Multiplier</i> | Pertanyaan | Skala Jawaban | | | | |
|----|-----------------------------------|--|---------------------------|---------------------|------------------------|----------------------|----------------------------|
| | | | VL (<i>Very Low</i>) | L (<i>Low</i>) | N (<i>Nomina</i>) | H (<i>High</i>) | VH (<i>Very High</i>) |
| | | <p>yang telah ditentukan</p> <ul style="list-style-type: none"> • Very High : kompresi 160% dari jadwal yang telah ditentukan | | | | | |

