



**TUGAS AKHIR - KS141501**

**RANCANG BANGUN APLIKASI ESTIMASI BIAYA  
PERANGKAT LUNAK MENGGUNAKAN METODE  
*FUNCTION POINT***

***DESIGN AND IMPLEMENTATION OF SOFTWARE  
COST ESTIMATION APPLICATION USING  
FUNCTION POINT METHOD***

**FANDHI AKHMAD  
NRP 05211440000171**

**Dosen Pembimbing  
Dr. Apol Pribadi S., S.T., M.T.  
Sholiq, S.T ,M.Kom, M.SA**

**DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018**



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - KS 141501**

**RANCANG BANGUN APLIKASI ESTIMASI  
BIAYA PERANGKAT LUNAK MENGGUNAKAN  
METODE *FUNCTION POINT***

**FANDHI AKHMAD**  
NRP 05211440000171

Dosen Pembimbing  
Dr. Apol Pribadi S., S.T., M.T.  
Sholiq, S.T, M.Kom, M.SA

**DEPARTEMEN SISTEM INFORMASI**  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - KS 141501**

**DESIGN AND IMPLEMENTATION OF  
SOFTWARE COST ESTIMATION  
APPLICATION USING FUNCTION POINT  
METHOD**

**FANDHI AKHMAD**  
NRP 05211440000171

**Supervisor**

**Dr. Apol Pribadi S., S.T., M.T.**  
**Sholiq, S.T, M.Kom, M.SA**

**DEPARTMENT OF INFORMATION SYSTEM**  
**Faculty of Information and Communication**  
**Technology**  
**Institute of Technology Sepuluh Nopember**  
**Surabaya 2018**

## LEMBAR PENGESAHAN

# RANCANG BANGUN APLIKASI ESTIMASI BIAYA PERANGKAT LUNAK MENGGUNAKAN METODE *FUNCTION POINT*

### 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:

Fandhi Akhmad  
05211440000171

Surabaya, Juli 2017

**KEPALA  
DEPARTEMEN SISTEM INFORMASI**

Dr.Ir. Aris Tianyanto, M.Kom  
NIP 196503101991021001



**RANCANG BANGUN APLIKASI ESTIMASI  
BIAYA PERANGKAT LUNAK MENGGUNAKAN  
METODE *FUNCTION POINT***

**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 :

**Fandhi Akhmad**  
**05211440000171**

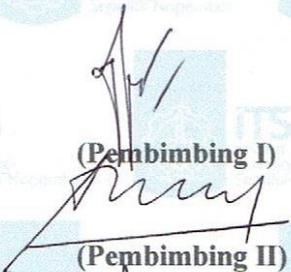
Disetujui Tim Penguji : Tanggal Ujian : 4 Juli 2018  
Periode Wisuda : September 2018

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

**Sholiq, S.T ,M.Kom, M.SA**

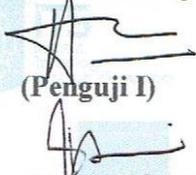
**Ir. Khakim Ghozali, M.MT**

**Hanim Maria Astuti, S.Kom, M.Sc**



**(Pembimbing I)**

**(Pembimbing II)**



**(Penguji I)**

**(Penguji II)**

# **RANCANG BANGUN APLIKASI ESTIMASI BIAYA PERANGKAT LUNAK MENGGUNAKAN METODE *FUNCTION POINT***

**Nama Mahasiswa : Fandhi Akhmad**  
**NRP : 05211440000171**  
**Departemen : Sistem Informasi FTIF-ITS**  
**Dosen Pembimbing I : Dr. Apol Pribadi S., S.T., M.T.**  
**Dosen Pembimbing II : Sholih, S.T ,M.Kom, M.SA**

## **ABSTRAK**

*Dalam rangka melakukan estimasi harga perangkat lunak, suatu tim biasanya melakukannya dengan cara membandingkan ukuran dan kompleksitas perangkat lunak. Ada dua permasalahan yang biasa dihadapi saat melakukan estimasi harga yaitu over-estimates dan under estimates. Kedua hal tersebut membuktikan bahwa estimasi harga perangkat lunak harus dilakukan. Salah satu cara untuk memperkirakan harga perangkat lunak yang akan dikembangkan adalah dengan menggunakan pendekatan Function Point. Function Point mengukur perangkat lunak dengan mengkuantifikasi fungsionalitas perangkat lunak. Salah satu tahapannya adalah menghitung nilai Relative Complexity Adjustment Factor yang merupakan 14 faktor kompleksitas. Perangkat lunak semakin berkembang kompleksitasnya. Faktor kompleksitas saat ini dirasa kurang relevan sehingga diperlukan modifikasi. Kompleksitas proses bisnis menghasilkan kebutuhan dalam pengembangan perangkat lunak. Perbedaan kompleksitas proses bisnis mempengaruhi ukuran dari perangkat lunak. Hal yang dilakukan adalah dengan menambahkan kompleksitas proses bisnis dalam 14 faktor kompleksitas. Dengan adanya penambahan kompleksitas proses bisnis, Function Point memiliki langkah perhitungan yang cukup kompleks dan membutuhkan pemahaman lebih tentang estimasi biaya perangkat lunak sehingga dibutuhkan tools bantuan untuk*

*mempermudah proses estimasi bagi orang-orang yang tidak mendalami estimasi harga perangkat lunak.*

*Oleh karena itu, aplikasi ini sangat dibutuhkan untuk estimasi biaya perangkat lunak berdasarkan metode Function Point. Pembuatan aplikasi menggunakan bahasa pemrograman php dan framework codeigniter(CI). Metode pengembangan perangkat lunak untuk membuat aplikasi estimasi biaya perangkat lunak ini adalah metode waterfall. Metode waterfall dipilih karena dapat menghasilkan sistem yang baik dengan pengerjaannya yang bertahap.*

*Hasil dari penelitian ini berupa aplikasi estimasi biaya perangkat lunak menggunakan metode function point dimana penelitian dimulai dari tahap analisis sampai testing.*

***Kata kunci: Rancang Bangun; Function Point; Proses Bisnis; Estimasi Harga***

**DESIGN SOFTWARE COST ESTIMATION  
APPLICATION USING FUNCTION POINT METHOD**

**Name** : Fandhi Akhmad  
**NRP** : 05211440000171  
**Department** : Information Systems FTIF -ITS  
**Supervisors I** : Dr. Apol Pribadi S., S.T., M.T.  
**Supervisors II** : Sholiq, S.T ,M.Kom, M.SA

**ABSTRACT**

*In order to estimate software price, a team usually does it by comparing the size and complexity of the software. There are two common problems encountered when estimating price are over estimates and under estimates. Both of these things prove that the estimated price of the software should be done. One way to estimate the price of software that will be developed is to use function point method. Function point measures software by quantifying software functionality. One step is to calculate the relative complexity adjustment factor value which is 14 factor complexity. The complexity of software increased. The current complexity factor is less relevant so modifications are required. The complexity of business processes generates a need in software development. Differences in the complexity of business processes affect the size of the software. The thing to do is to add the complexity of business processes in 14 factors of complexity. The addition of business process complexity, the function point has a fairly complex calculation step and requires more understanding of the software price estimate so that it takes tool to simplify the estimation process for people who don't deepen the software price estimate.*

*Therefore, this application is needed to estimate software cost based on function point method. In the making application using PHP programming language and codeigniter framework. Software development method to make cost estimation application is waterfall method. Waterfall method is chosen*

*because it can produce a good system with gradual workmanship.*

*The result of this research is cost estimation application using function point method where this research start from analysis phase until testing.*

***Keywords: Design; Function Point; Business Processes; Price Estimation***

## KATA PENGANTAR

Alhamdulillah atas karunia, rahmat, barakah, dan jalan yang telah diberikan Allah SWT selama ini sehingga penulis mendapatkan kelancaran dalam menyelesaikan tugas akhir dengan judul:

### **RANCANG BANGUN APLIKASI ESTIMASI BIAYA PERANGKAT LUNAK MENGGUNAKAN METODE *FUNCTION POINT***

Terima kasih atas pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik materi maupun spiritual demi tercapainya tujuan pembuatan tugas akhir ini. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada :

1. Allah S.W.T yang telah memberikan nikmat iman dan islam sehingga penulis dapat menyelesaikan tugas akhir ini.
2. Orangtua dan saudara yang selalu memberikan segala bentuk dukungan serta doa sehingga penulis dapat menyelesaikan tugas akhir ini.
3. Dosen Pembimbing I, Bapak Dr. Apol Pribadi Subriadi, S.T., M.T. terimakasih atas segala bimbingan, arahan, dukungan, ilmu, serta motivasi yang sangat bermanfaat bagi penulis.
4. Dosen Pembimbing II, Bapak Sholiq, S.T, M.Kom, M.SA terima kasih atas segala bimbingan, arahan, dukungan, ilmu, serta motivasi yang sangat bermanfaat bagi penulis.
5. Bapak Ir. Khakim Ghozali, M.Kom dan Ibu Hanim Maria Astuti, S.Kom, M.Sc, sebagai dosen penguji, terima kasih atas kritikan dan masukan yang bersifat membangun untuk peningkatan kualitas tugas akhir ini.
6. Bapak Ir. Aris Tjahyanto. M.Kom., selaku Ketua Departemen Sistem Informasi ITS, yang telah menyediakan fasilitas terbaik untuk kebutuhan penelitian mahasiswa.
7. Bapak Sholiq, S.T, M.Kom, M.SA selaku ketua laboratorium Manajemen Sistem Informasi dan Bapak

Hermono selaku laboran, terimakasih atas segala arahan, masukkan serta fasilitas yang telah diberikan di laboratorium.

8. Keluarga besar Osiris yang telah memberi dukungan kepada penulis untuk menyelesaikan tugas akhir ini.
9. Teman-teman E-Home, khususnya Naufal, Umar, Faiz, Bang In, Imad, Hendro, Joni, Tatus, Sigit, Dewa, Oman, Arep dan Bintang yang telah menjadi teman seperjuangan selama perkuliahan dan telah menjadi teman 1 atap selama tiga tahun sebagai tempat berkeluh kesah serta berbagi canda dan tawa.
10. Dewangga yang telah mengajari serta membantu penulis untuk menyelesaikan tugas akhir ini.
11. Adrian yang telah meminjamkan laptopnya kepada penulis untuk menyelesaikan tugas akhir ini.
12. Mas Faiz Fanani yang telah memberikan penulis referensi dalam membuat aplikasi.
13. Mbak Anggi yang telah membantu penulis dalam menjelaskan metode yang dikerjakan.
14. Fufu yang telah membantu penulis dalam mengatur format penulisan buku.

Penulis menyadari bahwa dalam penyusunan tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, penulis meminta maaf atas kesalahan yang dibuat dalam penulisan tugas akhir ini. Penulis membuka pintu selebar-lebarnya bagi pihak yang ingin memberikan kritik dan saran, dan penelitian selanjutnya yang ingin menyempurnakan karya dari tugas akhir ini. Akhir kata, semoga buku tugas akhir ini dapat bermanfaat bagi seluruh pembaca.

Surabaya, Juli 2018

Fandhi Akhmad

## DAFTAR ISI

ABSTRAK .....	v
ABSTRACT .....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	xi
DAFTAR TABEL .....	xv
DAFTAR GAMBAR .....	xix
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	5
1.3 Batasan Masalah .....	5
1.4 Tujuan Tugas Akhir .....	6
1.5 Manfaat Tugas Akhir .....	6
1.6 Relevansi .....	7
1.7 Sistematika Penulisan .....	7
BAB II TINJAUAN PUSTAKA .....	11
2.1 Studi Sebelumnya .....	11
2.2 Dasar Teori .....	15
2.2.1 Perkiraan Biaya .....	16
2.2.2 Proyek Perangkat Lunak .....	16
2.2.3 Kategori Ukuran Harga Pengembangan Perangkat Lunak .....	17
2.2.4 Rancang Bangun .....	17
2.2.5 Website .....	18
2.2.6 Metode Pengembangan Perangkat Lunak .....	19
2.2.7 Metode Waterfall .....	19
2.2.8 Unified Modelling Language (UML) .....	21
2.2.9 PHP ( <i>Hypertext Preprocessor</i> ) .....	22
2.2.10 Use Case Diagram .....	23
2.2.11 Sequence Diagram .....	24
2.2.12 Class Diagram .....	25
2.2.13 Activity Diagram .....	25

2.2.14	Framework Code Igniter .....	26
2.2.15	Pengujian.....	28
2.2.16	Unit Testing.....	29
2.2.17	Function Point .....	29
2.2.18	Function Point Modified .....	35
2.2.19	Estimasi Effort .....	39
2.2.20	Estimasi Biaya.....	40
<b>BAB III METODOLOGI PENELITIAN .....</b>		<b>45</b>
3.1	Tahap Analisis.....	46
3.2	Tahap Desain.....	46
3.3	Tahap Coding .....	46
3.4	Tahap Testing.....	47
3.5	Tahap Dokumentasi.....	47
<b>BAB IV PERANCANGAN .....</b>		<b>49</b>
4.1	Tahapan Analisis.....	49
4.1.1	Analisis Kebutuhan Pengguna .....	50
4.1.2	Analisis Kebutuhan Aplikasi.....	61
4.1.3	Use Case Diagram.....	70
4.1.4	Use Case Description .....	77
4.2	Tahap Desain.....	83
4.2.1	Desain Sequence Diagram .....	84
4.2.2	Desain Activity Diagram.....	87
4.2.3	Desain Class Diagram .....	91
4.2.4	Desain Antarmuka Pengguna .....	93
4.3	Tahapan Testing .....	103
4.3.1	Rencana Pengujian Aplikasi.....	104
4.3.2	Prosedur Pengujian.....	105
<b>BAB V IMPLEMENTASI .....</b>		<b>116</b>
5.1	Pembuatan Database .....	117
5.2	Pembuatan Kode Program.....	119
5.3	Pembuatan Kode Pengujian.....	121
5.4	Sampel Implementasi Sistem .....	123
<b>BAB VI HASIL DAN PEMBAHASAN.....</b>		<b>127</b>
6.1	Aplikasi Estimasi Biaya Perangkat Lunak .....	127
6.2	Hasil Pengujian Aplikasi.....	136

6.3 Validasi Aplikasi .....	145
6.4 Data Percobaan .....	149
6.5 Perhitungan Biaya dengan Aplikasi Estimasi Biaya Perangkat Lunak .....	153
6.6 Pembahasan.....	157
<b>BAB VII KESIMPULAN DAN SARAN .....</b>	<b>161</b>
7.1 Kesimpulan .....	161
7.2 Saran.....	162
<b>DAFTAR PUSTAKA .....</b>	<b>165</b>
<b>BIODATA PENULIS .....</b>	<b>169</b>
<b>LAMPIRAN A .....</b>	<b>- 1 -</b>
A.1 Requirement Analysis .....	- 1 -
A.2 Use Case Description .....	- 2 -
<b>LAMPIRAN B .....</b>	<b>- 1 -</b>
B.1. Use Case Diagram Aplikasi Estimasi Biaya Perangkat Lunak .....	- 1 -
B.2. Sequence Diagram Aplikasi Estimasi Biaya Perangkat Lunak .....	- 7 -
B.3. Activity Diagram Aplikasi Estimasi Biaya Perangkat Lunak .....	- 18 -
<b>LAMPIRAN C .....</b>	<b>- 1 -</b>
C.1 Hasil Aplikasi Estimasi Biaya Perangkat Lunak.....	- 1 -
<b>LAMPIRAN D .....</b>	<b>- 1 -</b>
D.1 Penjelasan Penilaian Indikator RCAF.....	- 1 -
<b>LAMPIRAN E .....</b>	<b>- 1 -</b>
E.1 Penjelasan Aktivitas .....	- 1 -

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 2. 1 Studi literatur 1.....	11
Tabel 2. 2 Studi literatur 2.....	12
Tabel 2. 3 Studi literatur 3.....	13
Tabel 2. 4 Studi litaratur 4.....	14
Tabel 2. 5 Studi litaratur 5.....	14
Tabel 2. 6 Bobot kompleksitas crude function point.....	31
Tabel 2. 7 Perhitungan bobot kompleksitas .....	31
Tabel 2. 8 Karakteristik perangkat lunak .....	34
Tabel 2. 9 Nilai kompleksitas proses bisnis .....	37
Tabel 2. 10 Konversi nilai kompleksitas kedalam indeks .....	37
Tabel 2. 11 Modifikasi karaktersitik perangkat lunak.....	38
Tabel 2. 12 Distribusi usaha dari masing-masing aktivitas....	41
Tabel 2. 13 Standar gaji tenaga IT .....	41
Tabel 2. 14 Tim pengembang perangkat lunak .....	42
Tabel 2. 15 Kategori biaya operasional.....	43
Tabel 4. 1 Translasi kebutuhan .....	50
Tabel 4. 2 Deskripsi Pengguna.....	57
Tabel 4. 3 Hak akses pengguna.....	61
Tabel 4. 4 Daftar kebutuhan sistem aplikasi estimasi biaya perangkat lunak .....	62
Tabel 4. 5 Daftar kebutuhan fungsional aplikasi estimasi biaya perangkat lunak .....	67
Tabel 4. 6 Use case description memasukkan informasi client .....	77
Tabel 4. 7 Use case description memasukkan deskripsi aplikasi .....	78
Tabel 4. 8 Use case description menilai indikator CFP .....	79
Tabel 4. 9 Use case description menilai indikator RCAF .....	80
Tabel 4. 10 Use case description mendapat nilai FP.....	82
Tabel 4. 11 Use case description mendapat nilai estimasi ....	82
Tabel 4. 12 Daftar tabel data aplikasi function point .....	91
Tabel 4. 13 Daftar halaman pengguna.....	94
Tabel 4. 14 Rencana pengujian aplikasi estimasi biaya perangkat lunak .....	105

Tabel 4. 15 Prosedur pengujian aplikasi estimasi biaya perangkat lunak .....	106
Tabel 6. 1 Hasil pengujian aplikasi estimasi biaya perangkat lunak .....	137
Tabel 6. 2 Proyek perusahaan DTS .....	149
Tabel 6. 3 Penilaian CFP .....	151
Tabel 6. 4 Penilaian RCAF.....	151
Tabel A. 1 Tabel Perhitungan Crude Function Point .....	- 1 -
Tabel A. 2 Tabel Perhitungan Relative Complexity Adjustment Factor .....	- 1 -
Tabel A. 3 Tabel Formulasi Function Point .....	- 2 -
Tabel A. 4 UC Description Login .....	- 2 -
Tabel A. 5 UC Description Logout .....	- 4 -
Tabel A. 6 UC Description Membuka Halaman Awal.....	- 5 -
Tabel A. 7 UC Description Memasukkan Fitur Aplikasi ...	- 6 -
Tabel A. 8 UC Description Melihat Daftar Fitur Aplikasi .	- 7 -
Tabel A. 9 UC Description Mengubah Biaya Tiap Aktivitas-	7 -
Tabel A. 10 UC Description Menambah Biaya Operasional-	9 -
Tabel A. 11 UC Description Melihat Biaya Operasional ..	- 10 -
Tabel A. 12 UC Description Mengirim Hasil Estimasi.....	- 11 -
Tabel A. 13 UC Description Validasi Hasil Estimasi.....	- 12 -
Tabel A. 14 UC Description Mencetak Dokumen Penawaran .	- 13 -
Tabel A. 15 UC Description Melihat Log Estimasi .....	- 14 -
Tabel A. 16 UC Description Menambah Daftar Pengguna-	15 -
Tabel A. 17 UC Description Melihat Daftar Pengguna.....	- 16 -
Tabel A. 18 UC Description Menambah Daftar Tim Anggota -	17 -
Tabel A. 19 UC Description Melihat Daftar Tim Anggota-	19 -
Tabel A. 20 UC Description Melihat Daftar Aktivitas.....	- 20 -
Tabel A. 21 UC Description Menambah Daftar Profesi....	- 21 -
Tabel A. 22 UC Description Melihat Daftar Profesi .....	- 22 -
Tabel D. 1 Penjelasan bobot tingkat kompleksitas komunikasi data .....	- 1 -

Tabel D. 2 Penjelasan bobot tingkat kompleksitas pemrosesan terdistribusi.....	- 2 -
Tabel D. 3 Penjelasan bobot tingkat kompleksitas performance .....	- 2 -
Tabel D. 4 Penjelasan bobot tingkat kompleksitas konfigurasi - 3 -	
Tabel D. 5 Penjelasan bobot tingkat frekuensi pengguna perangkat lunak .....	- 4 -
Tabel D. 6 Penjelasan bobot tingkat frekuensi input data ...	- 4 -
Tabel D. 7 Penjelasan bobot tingkat kemudahan penggunaan bagi user .....	- 5 -
Tabel D. 8 Penjelasan bobot tingkat frekuensi update data	- 6 -
Tabel D. 9 Penjelasan bobot tingkat kompleksitas processing data .....	- 7 -
Tabel D. 10 Penjelasan bobot tingkat kemungkinan penggunaan kembali .....	- 8 -
Tabel D. 11 Penjelasan bobot tingkat kemudahan dalam instalasi.....	- 8 -
Tabel D. 12 Penjelasan bobot tingkat kemudahan operasional perangkat lunak .....	- 9 -
Tabel D. 13 Penjelasan bobot tingkat perangkat lunak dibuat untuk multi-organisasi.....	- 10 -
Tabel D. 14 Penjelasan bobot tingkat kompleksitas dalam mengikuti perubahan.....	- 11 -
Tabel E. 1 Penjelasan aktivitas.....	- 1 -

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 1 Statistik proyek yang dikerjakan dengan dan tanpa function point .....	3
Gambar 2. 1 Metode Waterfall.....	20
Gambar 2. 2 Contoh Use Case Diagram .....	23
Gambar 2. 3 Contoh Sequence Diagram.....	24
Gambar 2. 4 Contoh Class diagram (sumber: Buku UML Bible) .....	25
Gambar 2. 5 Contoh Activity diagram .....	26
Gambar 2. 6 Prinsip kerja framework codeigniter .....	28
Gambar 2. 7 Modifikasi function point .....	35
Gambar 3. 1 Metodologi .....	45
Gambar 4. 1 Alur estimasi perhitungan usaha metode function point.....	69
Gambar 4. 2 Alur estimasi harga perangkat lunak .....	70
Gambar 4. 3 Desain use case pada aktor project manager .....	72
Gambar 4. 4 Desain use case pada aktor sekretaris.....	73
Gambar 4. 5 Desain use case pada aktor analis.....	74
Gambar 4. 6 Desain use case fitur autentifikasi .....	75
Gambar 4. 7 Desain use case fitur informasi client.....	75
Gambar 4. 8 Desain use case fitur memasukkan deskripsi aplikasi .....	76
Gambar 4. 9 Desain use case fitur perhitungan function point .....	76
Gambar 4. 10 Sequence diagram fitur memasukkan informasi client.....	84
Gambar 4. 11 Sequence diagram fitur memasukkan nama fitur .....	85
Gambar 4. 12 Sequence diagram fitur menilai indikator CFP85 .....	86
Gambar 4. 13 Sequence diagram fitur menilai indikator RCAF .....	86
Gambar 4. 14 Sequence diagram fitur mendapat nilai FP.....	86
Gambar 4. 15 Sequence diagram fitur mendapat nilai estimasi .....	87

Gambar 4. 16 Activity diagram memasukkan informasi client .....	88
Gambar 4. 17 Activity diagram memasukkan deskripsi aplikasi .....	88
Gambar 4. 18 Activity diagram menilai indikator CFP .....	89
Gambar 4. 19 Activity diagram menilai indikator RCAF .....	89
Gambar 4. 20 Activity diagram mendapat nilai FP .....	90
Gambar 4. 21 Activity diagram mendapat nilai estimasi .....	90
Gambar 4. 22 Class Diagram aplikasi estimasi biaya perangkat lunak .....	92
Gambar 4. 23 GUI Login.....	94
Gambar 4. 24 GUI Halaman Beranda .....	95
Gambar 4. 25 GUI Halaman informasi client.....	96
Gambar 4. 26 GUI Halaman deskripsi aplikasi .....	96
Gambar 4. 27 GUI Halaman CFP.....	97
Gambar 4. 28 GUI Halaman RCAF .....	98
Gambar 4. 29 GUI Halaman hasil function point.....	98
Gambar 4. 30 GUI Halaman hasil harga aplikasi .....	99
Gambar 4. 31 GUI Halaman <i>log</i> estimasi.....	99
Gambar 4. 32 GUI Halaman tambah pengguna.....	100
Gambar 4. 33 GUI Halaman daftar pengguna .....	100
Gambar 4. 34 GUI Halaman tambah profesi .....	101
Gambar 4. 35 GUI Halaman daftar profesi .....	101
Gambar 4. 36 GUI Halaman tambah anggota tim .....	102
Gambar 4. 37 GUI Halaman daftar anggota tim.....	102
Gambar 4. 38 GUI Halaman daftar aktivitas .....	103
Gambar 5. 1 Database aplikasi estimasi biaya perangkat lunak .....	118
Gambar 5. 2 Source code controller estimasi .....	119
Gambar 5. 3 Source code view halaman untuk menampilkan create estimation.....	120
Gambar 5. 4 Source code model untuk mengambil data estimasi berdasarkan id aplikasi .....	121
Gambar 5. 5 Source code pengujian aplikasi .....	122
Gambar 5. 6 Tampilan antarmuka homepage aplikasi estimasi biaya perangkat lunak.....	123

Gambar 5. 7 Tampilan antarmuka create estimation aplikasi estimasi biaya perangkat lunak .....	124
Gambar 5. 8 Tampilan antarmuka hasil estimasi aplikasi estimasi biaya perangkat lunak .....	125
Gambar 5. 9 Tampilan antarmuka log estimasi aplikasi estimasi biaya perangkat lunak.....	126
Gambar 6. 1 Halaman homepage .....	127
Gambar 6. 2 Form informasi client .....	128
Gambar 6. 3 Form deskripsi aplikasi .....	128
Gambar 6. 4 Form penilaian CFP.....	129
Gambar 6. 5 Form penilaian RCAF .....	130
Gambar 6. 6 Halaman hasil estimasi .....	130
Gambar 6. 7 Pop up berhasil kirim estimasi .....	131
Gambar 6. 8 Notifikasi email .....	131
Gambar 6. 9 Isi pesan email .....	131
Gambar 6. 10 Halaman request validasi.....	132
Gambar 6. 11 Halaman validasi hasil estimasi.....	132
Gambar 6. 12 Hasil cetak dokumen penawaran halaman 1	133
Gambar 6. 13 Hasil cetak dokumen penawaran halaman 2	134
Gambar 6. 14 Halaman hasil akhir estimasi.....	134
Gambar 6. 15 Halaman log estimasi .....	135
Gambar 6. 16 Hasil running kode pengujian.....	136
Gambar 6. 17 Hasil perhitungan CFP pada kertas kerja .....	145
Gambar 6. 18 Hasil perhitungan RCAF pada kertas kerja ..	146
Gambar 6. 19 Hasil nilai function point pada kertas kerja..	146
Gambar 6. 20 Perhitungan CFP pada aplikasi estimasi biaya perangkat lunak .....	147
Gambar 6. 21 Perhitungan RCAF pada aplikasi estimasi biaya perangkat lunak .....	148
Gambar 6. 22 Hasil nilai function point pada aplikasi function point.....	148
Gambar 6. 23 Hasil perhitungan CFP .....	153
Gambar 6. 24 Hasil perhitungan RCAF .....	154
Gambar 6. 25 Hasil perhitungan Function Point.....	154
Gambar 6. 26 Hasil perhitungan Effort .....	155
Gambar 6. 27 Hasil perhitungan rekap biaya .....	156
Gambar 6. 28 Hasil biaya keseluruhan.....	157

Gambar B. 1 Use case memasukkan nama fitur aplikasi.....	1 -
Gambar B. 2 Use case melihat hasil perhitungan estimasi ..	1 -
Gambar B. 3 Use case mengubah biaya aktivitas.....	2 -
Gambar B. 4 Use case biaya operasional .....	2 -
Gambar B. 5 Use case log estimasi aplikasi .....	3 -
Gambar B. 6 Use case mengirim hasil estimasi .....	3 -
Gambar B. 7 Use case validasi estimasi .....	4 -
Gambar B. 8 Use case cetak penawaran.....	4 -
Gambar B. 9 Use case pengelola pengguna.....	5 -
Gambar B. 10 Use case pengelola aktivitas .....	5 -
Gambar B. 11 Use case pengelola profesi .....	6 -
Gambar B. 12 Use case pengelola anggota pengembang....	6 -
Gambar B. 13 Sequence Login.....	7 -
Gambar B. 14 Sequence Logout.....	8 -
Gambar B. 15 Sequence menambah fitur aplikasi.....	8 -
Gambar B. 16 Sequence melihat daftar fitur aplikasi .....	9 -
Gambar B. 17 Sequence mengubah biaya tiap aktivitas....	10 -
Gambar B. 18 Sequence menambah biaya operasional.....	11 -
Gambar B. 19 Sequence melihat biaya operasional .....	12 -
Gambar B. 20 Sequence mengirim hasil estimasi .....	12 -
Gambar B. 21 Sequence validasi hasil estimasi .....	13 -
Gambar B. 22 Sequence mencetak dokumen penawaran ..	13 -
Gambar B. 23 Sequence melihat log estimasi .....	14 -
Gambar B. 24 Sequence menambah daftar pengguna .....	14 -
Gambar B. 25 Sequence melihat daftar pengguna.....	15 -
Gambar B. 26 Sequence menambah anggota tim pengembang	15 -
Gambar B. 27 Sequence melihat daftar anggota tim pengembang.....	16 -
Gambar B. 28 Sequence melihat daftar aktivitas.....	16 -
Gambar B. 29 Sequence menambah daftar profesi .....	17 -
Gambar B. 30 Sequence melihat daftar profesi .....	17 -
Gambar B. 31 Activity Login .....	18 -
Gambar B. 32 Activity Logout.....	18 -
Gambar B. 33 Activity menambah fitur aplikasi.....	19 -
Gambar B. 34 Activity melihat daftar fitur aplikasi .....	19 -

Gambar B. 35 Activity mengubah biaya tiap aktivitas .....	19 -
Gambar B. 36 Activity menambah biaya operasional.....	20 -
Gambar B. 37 Activity melihat biaya operasional .....	20 -
Gambar B. 38 Activity mengirim hasil estimasi .....	21 -
Gambar B. 39 Activity validasi hasil estimasi .....	21 -
Gambar B. 40 Activity mencetak dokumen penawaran....	22 -
Gambar B. 41 Activity melihat log estimasi .....	22 -
Gambar B. 42 Activity menambah daftar pengguna .....	22 -
Gambar B. 43 Activity melihat daftar pengguna .....	23 -
Gambar B. 44 Activity menambah anggota tim pengembang ..	23 -
Gambar B. 45 Activity melihat daftar anggota tim pengembang .....	24 -
Gambar B. 46 Activity melihat daftar aktivitas .....	24 -
Gambar B. 47 Activity menambah daftar profesi .....	25 -
Gambar B. 48 Activity melihat daftar profesi.....	25 -
Gambar C. 1 Halaman biaya estimasi .....	1 -
Gambar C. 2 Halaman biaya operasional.....	2 -
Gambar C. 3 Halaman tambah pengguna.....	2 -
Gambar C. 4 Halaman daftar pengguna .....	2 -
Gambar C. 5 Halaman tambah anggota tim .....	3 -
Gambar C. 6 Halaman daftar tim pengembang.....	3 -
Gambar C. 7 Halaman tambah profesi .....	3 -
Gambar C. 8 Halaman daftar profesi .....	4 -
Gambar C. 9 Halaman daftar aktivitas .....	4 -

*Halaman ini sengaja dikosongkan*

# **BAB I**

## **PENDAHULUAN**

Pada bab ini akan dibahas mengenai pendahuluan tugas akhir yang berisi latar belakang, perumusan masalah, batasan pengerjaan tugas akhir, tujuan dan manfaat dari pengerjaan tugas akhir serta sistematika penulisan buku tugas akhir.

### **1.1 Latar Belakang**

Perkembangan teknologi informasi semakin lama semakin pesat. Organisasi, baik itu organisasi profit maupun non-profit mengimplementasikan teknologi informasi untuk mempercepat proses bisnis yang ada[1]. Salah satu bentuk dari teknologi informasi tersebut adalah sebuah perangkat lunak. Dalam pembuatan perangkat lunak, perusahaan/organisasi kebanyakan menggunakan pihak ketiga dalam bentuk tender atau proyek. Adapun yang menjadi alasan mereka menggunakan pihak ketiga karena keterbatasan sumber daya yang ada dan juga waktu dalam membangun perangkat lunak[2].

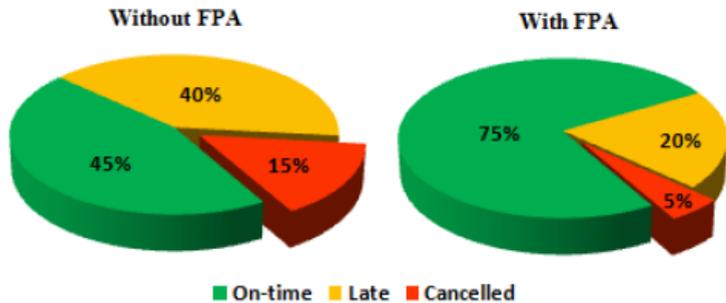
Pembuatan perangkat lunak saat ini telah menjadi suatu hal yang penting. Desain perangkat lunak atau pengembangan dilakukan berdasarkan masa depan atau kondisi tertentu suatu perusahaan. Penting untuk memahami kondisi tersebut untuk menghitung biaya dan durasi proyek perangkat lunak. Sebuah proyek dikatakan berhasil apabila proyek tersebut bisa diserahkan kepada *client* dengan tepat waktu sesuai dengan biaya dan kualitas yang diinginkan. Dalam melakukan estimasi dari waktu pengerjaan dan estimasi harga ada beberapa kendala yang mungkin akan terjadi.

Dalam rangka melakukan estimasi biaya perangkat lunak, suatu tim biasanya melakukannya dengan cara membandingkan ukuran dan kompleksitas perangkat lunak. Ada dua permasalahan yang biasa dihadapi saat melakukan estimasi biaya yaitu *over estimates* dan *under estimates*. *Over estimates* akan menimbulkan penambahan alokasi sumberdaya dari yang dibutuhkan sehingga meningkatkan penanganan manajerial

sedangkan *under estimates* secara tidak langsung akan mengurangi kualitas produk karena untuk menekan biaya maka perangkat lunak dibuat biasa saja dan tidak sesuai dengan standar. Kedua hal tersebut membuktikan bahwa perkiraan biaya perangkat lunak harus dilakukan dengan hati-hati dan terukur dengan jelas sehingga nantinya dapat dicapai keberhasilan proyek yang tepat waktu sesuai *budget* dan terpenuhinya standar kualitas produk[3].

Estimasi atau perkiraan ukuran perangkat lunak dalam proses pengembangan perangkat lunak memegang peranan penting untuk menjaga proses pengembangan tetap dalam kendali pengembang. Estimasi merupakan salah tugas paling menantang dalam manajemen proyek[4]. Proyek akan sulit dikendalikan jika sebelumnya tidak memiliki estimasi dan perencanaan. Estimasi dan perencanaan dalam proyek perangkat lunak bisa diawali dengan pengukuran besar dari perangkat lunak yang akan dibuat atau dikembangkan[5]. Salah satu cara untuk memperkirakan ukuran perangkat lunak yang akan dikembangkan adalah dengan menggunakan pendekatan *function point*.

Metode *function point* memiliki nilai akurasi yang tinggi dan cukup populer digunakan dikalangan pengembang. *International Function Point User Group* (IFPUG) secara resmi mendeklarasikan bahwa metode *function point* sesuai untuk semua jenis perangkat lunak[6]. Terdapat penelitian yang menggunakan *function point* menghasilkan *cost estimation* dengan deviasi 3.26 persen[7]. Hal ini berarti metode *function point* hampir mendekati *actual cost* dalam proyek pengembangan perangkat lunak. Berdasarkan *software productivity research*, pengukuran perangkat lunak dengan *function point* secara signifikan dapat meningkatkan kemungkinan keberhasilan proyek perangkat lunak secara tepat waktu dan tepat biaya. Gambar 1.1 menunjukkan statistik proyek yang dikerjakan dengan dan tanpa *function point*.



**Gambar 1** Statistik proyek yang dikerjakan dengan dan tanpa function point

(Sumber : *Software Productivity Research*)(George & Vogt, 2008)

*Function point* mengukur perangkat lunak dengan mengkuantifikasi fungsionalitas perangkat lunak yang disediakan untuk pengguna berdasarkan *design logic*. Salah satu tahapannya adalah menghitung nilai *relative complexity adjustment factor* dengan cara mengevaluasi *General System Characteristic* (GSC) perangkat lunak yang terdiri dari 14 faktor kompleksitas teknis yang memberikan gambaran jelas dari kompleksitas internal perangkat lunak[8]. Nilai *relative complexity adjustment factor* ini ditentukan berdasarkan pengalaman estimator dengan beberapa aturan yang terbatas dan tidak ada aturan spesifik yang lengkap atau standar proses menentukan faktor-faktor tersebut sehingga perlu adanya investigasi cara yang sesuai dalam menentukan nilai *relative complexity adjustment factor* dengan suatu standar untuk mengelola proses ini.

Banyak penelitian mengalami dilema terhadap estimasi biaya yaitu perangkat lunak yang semakin berkembang kompleksitasnya sehingga sangat sulit memprediksi secara akurat ukuran perangkat lunak yang dikembangkan[9]. Selain itu faktor kompleksitas saat ini dirasa kurang relevan sehingga modifikasi perlu dilakukan berdasarkan kebutuhan perangkat

lunak[7]. Oleh karena itu, *function point* harus didukung dengan data tambahan untuk memperkuat pengukuran perangkat lunak yang dikembangkan[10].

Perangkat lunak seperti semua sistem kompleks yang lain dapat berkembang dari satu periode waktu ke periode waktu lainnya. Kebutuhan dan proses bisnis dapat berubah seiring dengan laju perkembangan penggunanya[11]. Kompleksitas dari suatu proses bisnis menghasilkan kebutuhan yang kompleks dalam pengembangan perangkat lunak[12]. Perbedaan kompleksitas proses bisnis mempengaruhi ukuran dari perangkat lunak yang dikembangkan. Hal yang dilakukan untuk mengatasi masalah tersebut adalah memodifikasi *general system characteristics* (GSC) metode *function point* dengan menambahkan kompleksitas proses bisnis dalam 14 faktor kompleksitas.

Dengan adanya modifikasi metode *function point* membuat metode tersebut memiliki langkah perhitungan yang cukup kompleks dan membutuhkan pemahaman lebih tentang estimasi ukuran perangkat lunak sehingga dibutuhkan *tools* untuk mempermudah proses estimasi bagi orang-orang yang tidak mendalami estimasi ukuran perangkat lunak[13].

Berdasarkan latar belakang diatas, peneliti berupaya untuk memecahkan permasalahan tersebut yaitu dengan membuat aplikasi estimasi biaya perangkat lunak berdasarkan metode *function point*. Metode *function point* adalah metode untuk mengukur estimasi biaya suatu perangkat lunak berdasarkan kompleksitasnya[14]. Pelaksanannya untuk menyelesaikan permasalahan tersebut, peneliti akan merancang suatu sistem yakni berupa sebuah *website* untuk melakukan estimasi biaya suatu pembuatan perangkat lunak. Aplikasi yang dibangun berbentuk *website* sehingga membuat pengguna yang menggunakannya dapat membuat estimasi biaya proyek pembuatan suatu perangkat lunak secara *online*.

Metode pengembangan perangkat lunak yang digunakan untuk merancang aplikasi ini adalah metode *waterfall*. Metode *waterfall* dirasa cocok dan baik digunakan untuk penelitian ini

dikarenakan dokumentasi perancangan sistem sangat terorganisir dimana setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Tidak hanya itu saja, kualitas dari sistem yang dihasilkan juga akan baik dikarenakan proses pengerjaannya secara bertahap sehingga tidak berfokus pada tahapan tertentu. Penggunaan metode *waterfall* juga mudah untuk dikontrol sehingga meminimalisir kesalahan di setiap fasenya.

Tahapan yang peneliti lakukan dalam pembuatan aplikasi ini sesuai dengan metode *waterfall* yaitu merancang spesifikasi kebutuhan perangkat lunak, merancang perangkat lunak, membangun perangkat lunak dan melakukan pengujian terhadap perangkat lunak yang sudah dibuat. Selain itu, peneliti juga melakukan dokumentasi dari setiap tahapan pengembangan perangkat lunak yang peneliti lakukan.

## **1.2 Perumusan Masalah**

Berdasarkan uraian latar belakang di atas, maka rumusan permasalahan yang menjadi fokus dan akan diselesaikan dalam Tugas Akhir ini sebagai berikut :

1. Apa saja spesifikasi kebutuhan perangkat lunak yang diperlukan sebagai pedoman perancangan dan pembangunan aplikasi estimasi biaya perangkat lunak berdasarkan metode *function point*?
2. Bagaimana hasil perancangan aplikasi estimasi biaya perangkat lunak berdasarkan metode *function point*?
3. Bagaimana hasil pengembangan aplikasi estimasi biaya perangkat lunak berdasarkan metode *function point* dengan menggunakan bahasa pemrograman PHP dan *framework* Codeigniter?
4. Bagaimana hasil uji coba fitur-fitur yang ada pada aplikasi estimasi biaya perangkat lunak berdasarkan metode *function point* dengan menggunakan *white-box testing*?

## **1.3 Batasan Masalah**

Batasan permasalahan yang membatasi ruang lingkup pengerjaan tugas akhir ini adalah :

1. Metode estimasi yang digunakan dalam estimasi biaya yaitu metode *function point*.
2. Modifikasi metode *function point* hanya sebatas penambahan tingkat kompleksitas proses bisnis.
3. Aplikasi yang dibuat hanya menghasilkan sampai mendapatkan biaya perangkat lunak menggunakan metode *function point*.
4. Tahapan pengerjaan tugas akhir ini dimulai dari tahap analisis, pembuatan kode program dan hanya sampai tahap pengujian.
5. Pada tahap analisis kebutuhan, hanya menganalisa kebutuhan pengguna dan kebutuhan fungsional aplikasi hingga tahap pengujian.
6. Produk dari tugas akhir ini berbentuk aplikasi hanya berbasis *website*.
7. Bahasa pembuatan kode program yang digunakan hanya php dan framework yang digunakan hanya *codeigniter*.
8. Pengujian yang dilakukan hanya menggunakan *white-box testing*.

#### **1.4 Tujuan Tugas Akhir**

Dari perumusan masalah yang disebutkan sebelumnya, tujuan yang akan dicapai melalui tugas akhir ini adalah:

Menghasilkan aplikasi untuk mengestimasi biaya suatu perangkat lunak berdasarkan metode *function point*.

#### **1.5 Manfaat Tugas Akhir**

Melalui tugas akhir ini diharapkan dapat memberi manfaat yaitu:

1. Membantu suatu tim dalam proyek untuk menentukan biaya perkiraan sementara atau biaya estimasi awal menggunakan metode *function point*.
2. Memberikan kemudahan dan kejelasan tim dalam proyek untuk menentukan nilai proyek pengembangan perangkat lunak.

3. Membuat visualisasi dari modifikasi metode *function point* dengan adanya penambahan kompleksitas proses bisnis pada faktor kompleksitas.
4. Membuat *tools* untuk nantinya sebagai tempat mendapat data mengenai metode *function point* khususnya dengan adanya modifikasi penambahan tingkat kompleksitas proses bisnis.
5. Menambah referensi bagaimana cara membuat aplikasi untuk estimasi biaya perangkat lunak.
6. Menambah referensi mengenai cara penggunaan metode *waterfall* dalam pengembangan perangkat lunak.

## 1.6 Relevansi

Dalam ruang lingkup laboratorium Manajemen Sistem Informasi, penelitian ini termasuk ke dalam topik Pengembangan Perangkat Lunak yang menghasilkan sistem atau perangkat lunak yaitu aplikasi estimasi biaya perangkat lunak menggunakan metode *function point*. Penelitian ini juga mempunyai relevansi dengan mata kuliah wajib Analisis dan Desain Perangkat Lunak dan Konstruksi Pengujian Perangkat Lunak. Dibawah ini merupakan *roadmap* dari laboratorium Manajemen Sistem Informasi yang menunjukkan bahwa penelitian ini masuk ke dalam bidang pengembangan aplikasi yaitu perancangan atau desain aplikasi.

Berdasarkan relevansinya terhadap topik dan mata kuliah tersebut, maka dapat dikatakan bahwa penelitian tugas akhir ini telah mempunyai relevansi terhadap laboratorium Manajemen Sistem Informasi.

## 1.7 Sistematika Penulisan

Sistematika penulisan buku tugas akhir dibagi menjadi 7 bab, diantaranya adalah sebagai berikut.

## **BAB I PENDAHULUAN**

Bab I merupakan bagian pendahuluan tugas akhir yang berisi latar belakang, perumusan masalah, batasan pengerjaan tugas akhir, tujuan dan manfaat dari pengerjaan tugas akhir, serta sistematika penulisan tugas akhir.

## **BAB II TINJAUAN PUSTAKA**

Bab II merupakan bagian yang berisi tinjauan pustaka, yakni mengenai penelitian sebelumnya yang digunakan serta uraian dari istilah-istilah yang digunakan pada penulisan buku tugas akhir ini serta dasar teori yang digunakan pada tugas akhir ini.

## **BAB III METODOLOGI**

Pada bab III akan dijelaskan mengenai metode yang digunakan dalam penyelesaian tugas akhir. Metode dalam tugas akhir bertujuan sebagai pedoman dalam pengerjaan tugas akhir, sehingga proses pengerjaan menjadi lebih terarah dan sistematis. Tahapan dan proses dari metode ini dirangkum dalam sebuah diagram alur yang dapat memudahkan untuk memahami metode keseluruhan.

## **BAB IV PERANCANGAN**

Bab IV atau perancangan berisi mengenai proses perumusan kebutuhan, mulai dari identifikasi kebutuhan, alur kerja sistem, pemodelan proses bisnis, kebutuhan fungsional yang dirumuskan, bagaimana detail usecase dari sistem, *sequence diagram*, *class diagram*, *activity diagram*, *statechart diagram* serta perancangan pengujian dari sistem.

## **BAB V IMPLEMENTASI**

Pada bab V atau Implementasi akan berisi mengenai pembuatan database sistem dan menjelaskan hasil pembuatan kode menggunakan framework yang telah ditentukan.

## **BAB VI HASIL DAN PEMBAHASAN**

Bagian Hasil dan Pembahasan akan menjelaskan bagaimana hasil dari sistem yang telah dibuat, keluaran dari sistem tersebut dan hasil pengujian sistem. Pada hasil dan pembahasan dibuat

juga analisis mengenai metode yang sedang dikembangkan dengan melakukan eksperimen terhadap aplikasi yang telah dibuat.

## **BAB VII PENUTUP**

Bab VII merupakan bagian yang berisi kesimpulan dari keseluruhan pengerjaan tugas akhir ini serta saran untuk proses pengembangan aplikasi *Function Point*.

*Halaman ini sengaja dikosongkan*

## BAB II TINJAUAN PUSTAKA

Pada bab ini akan dibahas mengenai tinjauan pustaka dari tugas akhir. Bab ini berisi dasar teori yang mendukung tugas akhir. Adapun hal yang ada dalam tinjauan pustaka adalah sebagai berikut.

### 2.1 Studi Sebelumnya

Beberapa penelitian sebelumnya yang dijadikan sebagai acuan dalam pengerjaan tugas akhir ini disajikan dalam tabel dibawah ini.

**Tabel 2. 1 Studi literatur 1**

<b>Judul Penelitian</b>	Analisis Perbandingan Estimasi Effort Proyek Perangkat Lunak Menggunakan Metode Function Point Dengan Masukan Dokumen Use Case, Data Flow Diagram Dan Natural Language
<b>Penulis</b>	Rochma Permatasari, S1 Sistem Informasi, Institut Teknologi Sepuluh Nopember.
<b>Tujuan Penelitian</b>	<ol style="list-style-type: none"><li>1. Untuk mengetahui manakah hasil estimasi <i>effort</i> yang paling mendekati akurat dalam pengembangan proyek perangkat lunak dari ketiga masukan, yaitu dokumen <i>use case</i>, <i>data flow diagram</i> dan <i>natural language</i></li><li>2. Untuk mengetahui perbedaan yang timbul antara acuan menggunakan</li></ol>

	estimasi <i>effort</i> awal dan acuan menggunakan <i>actual effort</i> proyek perangkat lunak
<b>Hasil Penelitian</b>	Dari ketiga dokumen masukan, dokumen yang menghasilkan hasil <i>function point</i> mendekati nilai <i>actual effort</i> adalah dokumen <i>data flow diagram</i> dikarenakan dokumen <i>data flow diagram</i> menggambarkan secara detail dan mampu mengefisiensikan sumber daya teknologi dengan mengintegrasikan kebutuhan data dari level atas hingga level yang paling bawah sehingga poin pada komponen-komponen <i>function point</i> menjadi lebih banyak.
<b>Keterkaitan dengan Penelitian yang dilakukan</b>	Penelitian yang dilakukan berkaitan dengan estimasi harga suatu perangkat lunak.

Tabel 2. 2 Studi literatur 2

<b>Judul Penelitian</b>	Development of Software Size Estimation Application using Function Point Analysis (FPA) Approach with Rapid Application Development
<b>Penulis</b>	Hamzah, Ristu Saptono, Rini Anngrainingsih, S1 Informatika, Universitas Sebelas Maret, Surakarta.
<b>Tujuan Penelitian</b>	Untuk melakukan pembangunan aplikasi estimasi ukuran perangkat lunak dengan metode <i>function point analysis</i> .
<b>Hasil Penelitian</b>	Didapatkan aplikasi yang dapat memperkirakan ukuran perangkat

	<p>lunak dalam <i>Line of Code</i> (LOC). Untuk memperkirakan ukuran <i>software</i>, harus dimasukkan nilai TDIdan juga <i>function</i> yang meliputi tabel dan <i>function</i> dari <i>software</i> yang dibangun agar didapatkan perkiraan ukurannya. <i>Testing</i> yang dilakukan pada aplikasi memberikan hasil perkiraan ukuran <i>software</i> memiliki nilai rata-rata <i>relative error</i> sebesar 3,73% sehingga dapat disimpulkan tingkat kepercayaan atau akurasi hasil perhitungan yang dilakukan oleh aplikasi sebesar 96,27%</p>
<b>Keterkaitan dengan Penelitian yang dilakukan</b>	<p>Penelitian yang dilakukan merupakan pembangunan satu aplikasi dengan menggunakan SDLC yang berbeda/</p>

Tabel 2. 3 Studi literatur 3

<b>Judul Penelitian</b>	Rancang Bangun Aplikasi Use Case Point Untuk Estimasi Harga Perangkat Lunak
<b>Penulis</b>	Mukhamad Faiz Fanani, S1 Sistem Informasi, Institut Teknologi Sepuluh Nopember.
<b>Tujuan Penelitian</b>	Menghasilkan aplikasi <i>use case point</i> untuk mengestimasi harga perangkat lunak pada studi kasus perusahaan DTS.
<b>Hasil Penelitian</b>	Hasil dari penelitian yaitu menghasilkan aplikasi <i>use case point</i> untuk estimasi harga perangkat lunak yang sudah teruji melalui tiga tahapan <i>increment</i> .

<b>Keterkaitan dengan Penelitian yang dilakukan</b>	Penelitian yang dilakukan membangun <i>website</i> yang digunakan untuk mengukur estimasi harga perangkat lunak namun metode estimasi dan SDLC yang digunakan berbeda.
---	--

Tabel 2. 4 Studi literatur 4

<b>Judul Penelitian</b>	Modifikasi Metode Function Point Dengan Menambahkan Kompleksitas Proses Bisnis Pada General System Characteristics Untuk Estimasi Biaya Perangkat Lunak
<b>Penulis</b>	Anggi Yhurinda Perdana Putri, S2 Sistem Informasi, Institut Teknologi Sepuluh Nopember.
<b>Tujuan Penelitian</b>	Untuk mengetahui bagaimana pengaruh modifikasi faktor kompleksitas proses bisnis terhadap hasil pengukuran perangkat lunak/
<b>Perbedaan Penelitian</b>	Penelitian yang dilakukan hanya terbatas teori mengenai adanya penambahan kompleksitas proses bisnis tanpa adanya visualisasi.
<b>Keterkaitan dengan Penelitian yang dilakukan</b>	Penelitian ini digunakan dalam hal visualisasi perangkat lunak.

Tabel 2. 5 Studi literatur 5

<b>Judul Penelitian</b>	A Model of Owner Estimate Cost for Software Development Project in Indonesia
<b>Penulis</b>	Sholiq, Apol Pribadi Subriadi, Feby Artwodini Muqtadiroh, Renny Sari Dewi, Sistem Informasi, Institut Teknologi Sepuluh Nopember.

<b>Tujuan Penelitian</b>	Untuk menghasilkan konseptual model yang terdiri dari <i>personnel direct cost, non-personnel direct costs</i> dan <i>taxes</i> .
<b>Hasil Penelitian</b>	Menghasilkan konseptual model dari biaya perangkat lunak menggunakan metode <i>use case point</i> dengan menggunakan estimasi usaha dan distribusi usaha.
<b>Keterkaitan dengan Penelitian yang dilakukan</b>	Penelitian yang dilakukan menggunakan estimasi usaha dan distribusi usaha dalam menentukan harga proyek pengembangan perangkat lunak.

Berdasarkan penelitian sebelumnya, dapat dirangkum menjadi beberapa hal yang menjadi pendukung dalam penelitian ini. Rangkuman dari penelitian sebelumnya adalah sebagai berikut:

1. *Function point* merupakan salah satu metode yang digunakan untuk mengestimasi harga suatu perangkat lunak melalui kompleksitas yang dimiliki suatu *software* namun memiliki akurasi yang cukup tinggi.
2. Penelitian yang dilakukan yaitu visualisasi dalam bentuk *website* mengenai penelitian penambahan kompleksitas proses bisnis.
3. Harga perangkat lunak dalam penelitian ini didapatkan menggunakan konseptual model menggunakan estimasi usaha dan distribusi usaha.
4. Rancang bangun aplikasi yang dilakukan menggunakan metode *waterfall* yang dimulai dari tahap analisis dan hanya sampai tahap *testing* tanpa melakukan tahapan *maintenance*.

## 2.2 Dasar Teori

Pada bagian ini akan dibahas mengenai dasar teori yang digunakan sebagai dasar informasi untuk pengerjaan tugas akhir.

### **2.2.1 Perkiraan Biaya**

Estimasi biaya dan usaha proyek merupakan suatu kegiatan pengaturan sumber daya dalam mencapai tujuan dan sasaran dari target yang dikehendaki. Permasalahan yang biasa dihadapi saat melakukan estimasi biaya yaitu *over estimates* dan *under estimates*. *Over estimates* akan menimbulkan penambahan alokasi sumber daya dari yang dibutuhkan sehingga akan meningkatkan penanganan manajerial sedangkan *Under estimates* secara tidak langsung akan mengurangi kualitas produk karena untuk menekan biaya maka perangkat lunak dibuat biasa saja tidak sesuai dengan standar. Kedua hal tersebut membuktikan bahwa perkiraan biaya perangkat lunak harus dilakukan dengan hati-hati dan terukur dengan jelas sehingga nantinya dapat dicapai keberhasilan proyek yang tepat waktu sesuai *budget* dan terpenuhinya standar kualitas produk[15].

### **2.2.2 Proyek Perangkat Lunak**

Proyek perangkat lunak adalah suatu kegiatan pengembangan perangkat lunak untuk mencapai tujuan tertentu dengan syarat-syarat tertentu seperti dibatasi waktu yang jelas, melibatkan sumberdaya yang bervariasi dan interkorelasi antara satu sumber dengan sumber lainnya, tersedia dan jelas biaya atau modal terdapat unsur resiko (proyek, bisnis, sosial, politik dan sebagainya). Tujuan dari perancangan perangkat lunak adalah untuk memperbaiki kualitas produk perangkat lunak, meningkatkan produktivitas dan memuaskan teknisi perangkat lunak[16].

Dalam pembangunan proyek perangkat lunak pastinya ada hal-hal yang menentukan apakah proyek perangkat lunak akan berjalan dengan yang diharapkan sehingga menghasilkan kualitas yang baik. Bagian penting tersebut adalah Manajemen Proyek Perangkat Lunak. Definisi Manajemen Proyek Perangkat Lunak itu sendiri adalah aktifitas perancangan dan pengontrolan membuat perangkat lunak dalam jangka waktu tertentu untuk memenuhi kebutuhan.

Manajemen Proyek Perangkat Lunak mengatur empat hal penting yaitu personel, produk, proses dan proyek. Berdasarkan empat hal tersebut, personel merupakan hal yang terpenting karena personel adalah aktor yang mengisi pembangunan perangkat lunak. Dengan adanya personel atau orang-orang yang mempunyai kemampuan yang baik maka produk yang dihasilkan, proses yang berjalan serta proyek yang dikerjakan akan baik pula.

### **2.2.3 Kategori Ukuran Harga Pengembangan Perangkat Lunak**

Menurut Dr. John Flacket, untuk mendapatkan pemahaman yang lebih baik tentang kompleksitas pengembangan perangkat lunak, sebuah perangkat lunak tidak akan bekerja sampai setiap fungsi telah dirancang, dibangun, diuji dan dimasa depan, dipertahankan. Setiap pengambilan, menyimpan dan menampilkan data harus dipikirkan. Setiap rumus dan perhitungan harus diketahui, program dan hasil dari dua kali lipat diperiksa. Teknologi yang digunakan untuk memecahkan masalah harus dipilih, dibentuk dan diterapkan. Persyaratan harus diselidiki dan disusun, antarmuka pengguna perlu dirancang dan uji kasus harus ditulis. Akhirnya, sebuah perangkat lunak yang dihasilkan harus dikerahkan dan dipelihara.

Dr. John Flacket juga memberitahu *range* dari biaya perangkat lunak[15], yaitu :

1. *Low range* : \$10,000 - \$25,000
2. *Medium range* : \$25,000 - \$100,000
3. *Average range* : \$100,000 - \$250,000
4. *High range* : \$250,000 - \$800,000
5. *Unique range* : \$800,000 - \$unlimited

### **2.2.4 Rancang Bangun**

Menurut R. Pressman dalam bukunya “*Software Engineering A Practitioner’s Approach 7<sup>th</sup>*” pengertian rancang adalah proses menganalisa kebutuhan dan mendeskripsikan dengan detail

komponen-komponen yang akan diimplementasi. Dalam buku tersebut juga dijelaskan pengertian dari bangun yaitu menciptakan sistem baru atau memperbaiki sistem yang sudah ada.

George M. Scott mengemukakan bahwa proses rancang sistem adalah bagaimana suatu sistem akan menyelesaikan apa yang harus diselesaikan. Kegiatan rancang sistem juga termasuk mengkonfigurasi komponen perangkat lunak dan perangkat keras dari suatu sistem.

Dengan demikian dapat disimpulkan bahwa rancang – bangun adalah kegiatan menganalisa kebutuhan dan mendeskripsikan semua komponen-komponen yang terlibat dengan sistem kemudian menciptakan sistem berdasarkan kebutuhan dan komponen-komponen yang sudah dideskripsikan sebelumnya. Tujuan akhir dari kegiatan rancang – bangun adalah menjawab sebuah permasalahan dengan memberikan solusi yang sesuai dengan kebutuhan[17].

### 2.2.5 Website

*Website* adalah rangkaian atau sejumlah halaman di internet yang memiliki topik saling terkait untuk mempresentasikan suatu informasi. *Website* juga dapat diartikan dengan miniatur dan representasi dari perorangan, lembaga, organisasi ataupun perusahaan yang bersangkutan. *Website* dapat memberikan informasi dalam bentuk gambar serta visualisasi orang atau lembaga yang membuatnya. *Website* dapat dibuat dengan tujuan apa saja, tergantung pemilikinya karena semua hal dapat dituangkan dalam bentuk *website*[18]. Berikut adalah contoh *website* yang sering dijumpai :

1. *Website* untuk mencari informasi (*search engine*) misalnya google.com, yahoo.com, dll.
2. *Website* jejaring sosial (*social media*) misalnya facebook.com, twitter.com, ask.fm, dll.
3. *Website* toko *online* (*online shop*) misalnya zalora.com, lazada.com, dll.

4. *Website* perusahaan (*company profile*) misalnya pertamina.com, pln.co.id, dll.
5. *Website* portal berita (*news*) misalnya detik.com, kapanlagi.com, republik.co.id, dll.

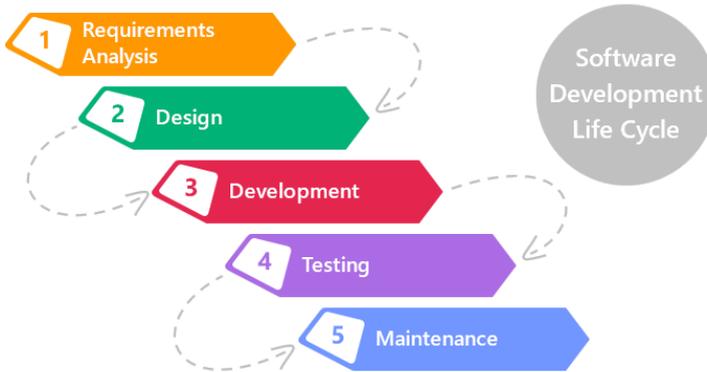
### 2.2.6 Metode Pengembangan Perangkat Lunak

Metodologi pengembangan perangkat lunak merupakan metode atau acuan kerja yang digunakan untuk menstrukturisasi, perencanaan dan kontrol dalam proses pengembangan sebuah sistem informasi. Dalam metodologi juga termasuk melakukan pendefinisian awal luaran dari setiap proses yang ada dalam metodologi tersebut dan juga *tools* yang diperlukan dalam menyelesaikan sebuah proyek pengembangan perangkat lunak. Berikut adalah beberapa metodologi yang saat ini populer dalam pengembangan perangkat lunak[1].

1. Waterfall Model
2. Prototyping Model
3. Agile Software Development
4. Rapid Application Development
5. Spiral Model
6. Extreme Programming

### 2.2.7 Metode Waterfall

SDLC (*software Development Life Cycle*) adalah proses perancangan aplikasi serta metodologi yang digunakan untuk mengembangkan aplikasi. Menurut Pressman metode *waterfall* merupakan model pendekatan secara sistematis dan urut mulai dari level kebutuhan, aplikasi kemudian tahap analisis, desain, *coding*, *testing/verification* dan *maintenance*. Metode ini disebut dengan *waterfall* karena pengerjaannya tahap demi tahap, untuk melanjutkan ke tahap selanjutnya harus menunggu selesai tahap sebelumnya[18].



**Gambar 2. 1 Metode Waterfall**

1. *Requirements Analysis*

Langkah ini merupakan analisis terhadap kebutuhan aplikasi dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan *customer* maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artiker maupun dari internet.

2. *Design*

Langkah ini merupakan tahapan dalam merancang *use case diagram*, *sequence diagram*, *class diagram* dan desain *interface* yang nantinya akan digunakan acuan dalam proses pembuatan kode program.

3. *Coding*

Langkah ini merupakan tahapan proses pembuatan kode program, *coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *customer*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu aplikasi yang berarti penggunaan komputer akan dimaksimalkan dalam tahapan ini.

4. *Testing*

Langkah ini dilakukan setelah tahapan pembuatan kode program diselesaikan maka akan dilakukan *testing* terhadap sistem yang telah dibuat. Adapun tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem untuk kemudian bisa diperbaiki.

## **2.2.8 Unified Modelling Language (UML)**

*Unified Modelling Language (UML)* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *Unified Modelling Language* menawarkan sebuah standar untuk merancang model sebuah sistem[19].

### **2.2.8.1 Diagram Pada UML**

Berikut ini beberapa diagram standar yang digunakan dalam UML :

1. *Use case diagram*
2. *Sequence diagram*
3. *Class diagram*
4. *Activity diagram*

### **2.2.8.2 Tahapan Desain UML**

Menurut Sri, 2013[20], terdapat langkah-langkah dalam pengembangan perangkat lunak dengan menggunakan UML sebagai berikut :

1. Membuat daftar proses bisnis mulai level tinggi dengan tujuan untuk mendefinisikan aktivitas serta proses yang mungkin muncul.
2. Melakukan pemetaan *use case* untuk setiap proses bisnis yang sudah dibuat.
3. Membuat *deployment* diagram untuk mendefinisikan arsitektur fisik dari sistem.
4. Melakukan pendefinisian kebutuhan lain seperti kebutuhan non-fungsional yang harus ada pada sistem yang akan dibuat.

5. Membuat aktivitas diagram dari setiap desain *use case* yang ada.
6. Membuat *sequence* diagram dari *use case* atau tiap alur pekerjaan.
7. Membuat rancangan *user interface* model yang menyediakan antarmuka bagi pengguna untuk menjalankan skenario *use case*.
8. Berdasarkan model yang ada, buatlah *class* diagram. Setiap *package* dipecah sehingga menjadi hirarki kelas yang lengkap dengan *method* dan atributnya.
9. Melakukan pengelompokan kelas menjadi komponen-komponen.
10. Memperhalus diagram *deployment* yang sudah dibuat. Misalnya dengan penjelasan kebutuhan perangkat lunak, sistem operasi dan sebagainya. Lakukan pemetaan kedalam bentuk node.
11. Melakukan pembangunan sistem berdasarkan hasil model yang sudah dibuat.

### 2.2.9 PHP (*Hypertext Preprocessor*)

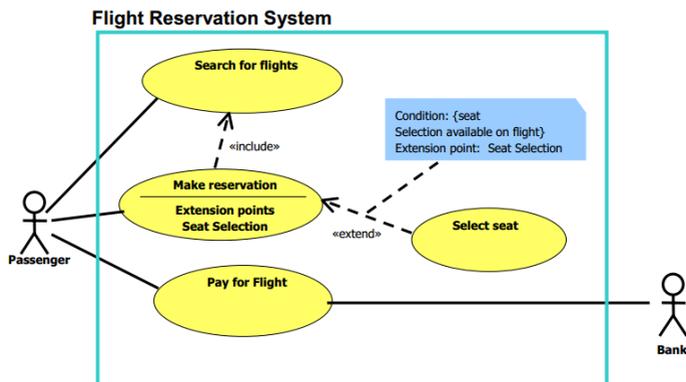
PHP merupakan bahasa pemrograman berbasis web yang diciptakan oleh Rasmus Lerdorf pada tahun 1995. Pada awalnya PHP mempunyai kepanjangan kata yaitu *Personal Home Page*[21]. Pemrograman ini sudah banyak digunakan untuk pembuatan aplikasi atau sistem yang berbasis *website*. Hal ini dibuktikan dari sebuah penelitian yang dilakukan pada tahun 1997 yang menyatakan bahwa bahasa pemrograman PHP sudah digunakan hampir 500.000 lebih *website* di dunia. Keuntungan dari bahasa pemrograman PHP ini yaitu sebagai berikut :

1. Bahasa pemrograman PHP memberikan kemudahan kepada *developer* dalam hal melakukan proses rilis dan proses pembaharuan perangkat lunak yang lebih cepat.
2. Bahasa pemrograman PHP memiliki *library* yang sudah tersedia didalamnya.

- Bahasa pemrograman PHP merupakan bahasa pemrograman yang *open source* sehingga memberikan fitur-fitur yang luas dan lengkap tanpa ada pembatasan.

### 2.2.10 Use Case Diagram

Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behaviour*) sistem yang akan dibuat[22]. Dalam *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, *use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Terdapat beberapa simbol dalam menggambarkan diagram *use case* yaitu *use case*, *aktor* dan *relasi*. Hal yang perlu diingat mengenai diagram *use case* bukan menggambarkan tampilan antarmuka (*user interface*), arsitektur dari sistem kebutuhan non-fungsional dan tujuan performansi sedangkan untuk penamaan *use case* adalah nama didefinisikan sesederhana mungkin, dapat dipahami dan menggunakan kata kerja[18].

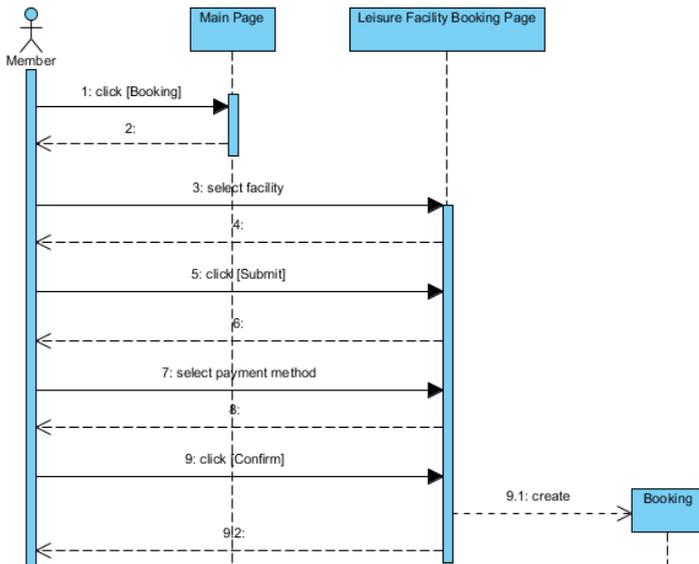


Gambar 2. 2 Contoh Use Case Diagram

### 2.2.11 Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan hubungan antara objek dalam waktu yang berurutan (Pender, 2003). Manfaat dari pembuatan *sequence diagram* ini yaitu untuk mengidentifikasi pesan apa saja yang dipertukarkan antara objek satu dengan objek lainnya dalam sebuah skenario *use case*[1]. Sequence diagram digunakan dalam tahapan analisa dan desain. Adapun kegunaan dari *sequence diagram* ini yaitu untuk memodelkan :

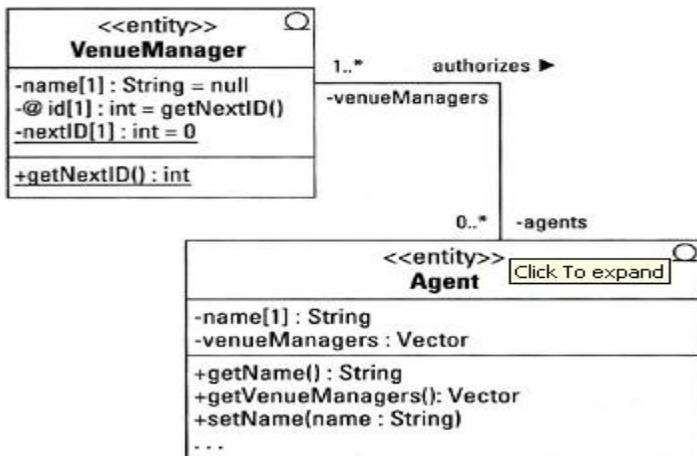
1. Deskripsi dari sebuah sistem yang digambarkan dalam diagram perilaku (*use case diagram*) kedalam bentuk hubungan antar aktor dan *use case diagram*.
2. Logika dari sebuah *method* yang ada dalam suatu fungsi, baik dalam bentuk prosedur, operasi atau fungsi.
3. Logika dari sebuah layanan.



Gambar 2. 3 Contoh Sequence Diagram

### 2.2.12 Class Diagram

*Class diagram* merupakan grafik yang menggambarkan koleksi dari deklaratif model dalam sebuah elemen. *Class diagram* menyediakan ringkasan dari sistem target dengan mendeskripsikan objek dan kelas kedalam sistem dan hubungannya diantara keduanya[1]. *Class diagram* memodelkan sumber daya masing-masing dalam hal struktur, hubungan dan perilaku, 3 standar kompartemen ditemukan di hampir setiap kelas : nama, atribut dan operasi.



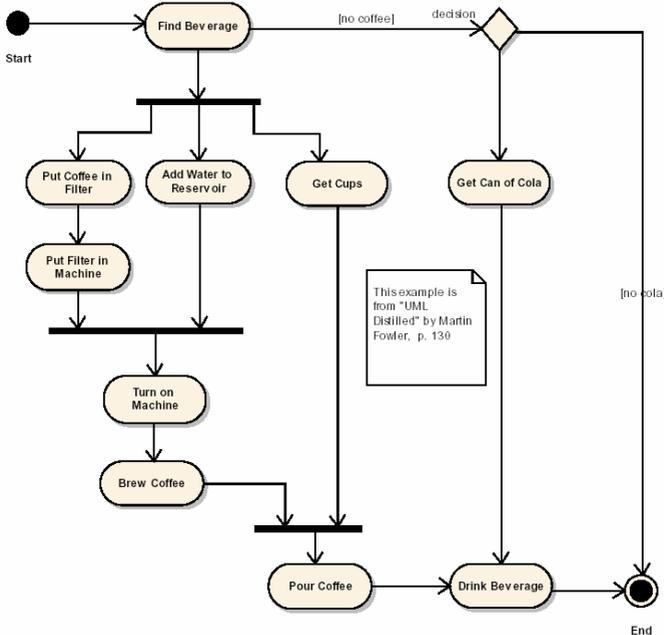
Gambar 2. 4 Contoh Class diagram (sumber: Buku UML Bible)

### 2.2.13 Activity Diagram

*Activity diagram* menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, keputusan yang mungkin terjadi dan bagaimana alur tersebut berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi[23].

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan,

sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggungjawab untuk aktivitas tertentu seperti pada Gambar 2.5.



Gambar 2. 5 Contoh Activity diagram

### 2.2.14 Framework Code Igniter

*Framework* adalah abstraksi di dalam sebuah perangkat lunak yang menyediakan fungsi *generic* sehingga dapat dirubah oleh kode yang dibuat oleh *user* yang dapat menyediakan perangkat lunak untuk aplikasi tertentu[24].

Metode MVC (Model, View, Control) adalah sebuah arsitektur yang dapat diimplementasikan secara bebas dengan atau tanpa bahasa pemrograman berorientasi objek. Dengan demikian metode MVC dapat diimplementasikan dalam sebuah *framework*.

Codeigniter merupakan sebuah *framework* pemrograman web dengan menggunakan bahasa php. *Framework* ini ditulis dengan menggunakan bahasa php versi 4 dan versi 5 oleh Rick Ellislab yang menjadi CEO Ellislab, Inc dan dipublikasikan dengan lisensi dibawah Apache/BSD *open source*.

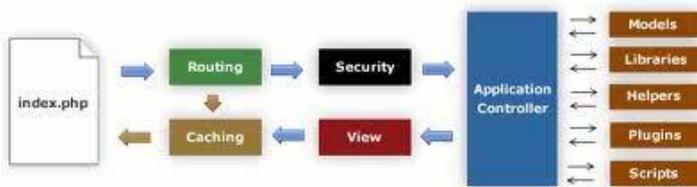
Sebagai *framework* yang menganut metode MVC, CodeIgniter juga terbagi menjadi 3 bagian utama yaitu :

1. Model, bagian ini berisi kode yang digunakan untuk terkoneksi dengan mengakses database.
2. View, berisi kode HTML dan PHP yang digunakan untuk menampilkan informasi ke layar *browser* yang biasanya berisi kode yang berhubungan dengan format tampilan, misalnya huruf, font, warna dan lain-lain.
3. Controller, berisi kode *script* yang menjalankan fungsi aturan bisnis aplikasi dan menjadi perantara antara model dan view serta seluruh sumber daya yang dibutuhkan untuk memproses permintaan layanan HTTP untuk ditampilkan dalam halaman web.

Prinsip kerja utama CodeIgniter terletak pada *file* index.php yang diletakkan pada direktori *root* aplikasi. File ini akan memicu dan mengarahkan permintaan layanan halaman web ke dalam tubuh CodeIgniter. Mekanismenya adalah sebagai berikut :

1. *File* index.php bertindak sebagai pengedali utama yang berfungsi memuat kode *script* utama yang berfungsi menjalankan CodeIgniter.
2. Modul *Routing* berfungsi menerima permintaan layanan HTTP untuk menentukan arah eksekusi *script* yang akan dilaksanakan.
3. Jika konfigurasi *cache* tersedia, maka sistem langsung mengeksekusi untuk ditampilkan dihalaman web.
4. Permintaan layanan HTTP dan data-data dari form yang dikirimkan ke *server*, akan disaring dan diamankan oleh modul *security*.

5. Data dikirimkan ke modul *controller* yang akan mengendalikan akses ke modul model, *library*, *plugin* dan modul-modul sumber daya lainnya.
6. Kemudian *controller* akan mengirimkan data ke modul *view* untuk ditampilkan ke halaman web. Jika konfigurasi *caching* diaktifkan maka sebelum tampilan ini dikirimkan ke web untuk ditampilkan ke *browser* maka tampilan ini akan disimpan ke dalam *cache* sehingga permintaan yang sama dapat dilakukan dengan lebih cepat.



**Gambar 2. 6 Prinsip kerja framework codeigniter**

Pada prinsipnya, penggunaan metode MVC adalah berupa kelas. Di dalam *folder* sistem terdapat definisi dari kelas *controller* dan model sedangkan *view* adalah murni *file* yang disisipkan ke dalam fungsi anggota di dalam pewarisan kelas *controller*.

Pada pembuatan aplikasi menggunakan *framework CodeIgniter* maka objek pengendali berupa aturan bisnis didefinisikan sebagai objek *controller* dengan cara melakukan pewarisan kelas (*inheritance*) ke objek *controller* sedangkan fungsi akses datanya didefinisikan dalam suatu kelas yang mewarisi (*inheritance*) kelas model.

### 2.2.15 Pengujian

Menurut Hetzel 1983, pengujian merupakan suatu aktivitas yang digunakan untuk melakukan evaluasi suatu atribut atau fitur suatu program atau sistem untuk memastikan apakah sistem atau program telah memenuhi kebutuhan atau hasil yang diharapkan[25]. Adapun tiga aktivitas utama dalam pengujian yaitu sebagai berikut[26] :

1. *Verification*

Verifikasi merupakan pengujian untuk memastikan bahwa hasil perancangan sesuai dengan kebutuhan pengguna.

2. *Validation*

Validasi merupakan pengujian yang dilakukan untuk memastikan bahwa sistem dibuat sesuai dengan perancangan atau desain yang sudah dibuat sebelumnya.

3. *Bug Error Finding*

Merupakan pengujian yang dilakukan untuk mengetahui *error* dari aplikasi.

### 2.2.16 Unit Testing

Pengujian Unit (*Unit Testing*) adalah pengujian yang difokuskan pada unit terkecil dari program. Pengujian ini didasarkan pada informasi deskripsi perancangan detail perangkat lunak. Pada umumnya pengujian ini dilakukan secara *white-box* dan *source code based testing* dengan melakukan pengecekan jalur khusus pada struktur kendali modul untuk meyakinkan kelengkapan cakupan dan deteksi maksimum kesalahan[27].

Pengujian ini dilakukan dengan metode *white-box* yang mempunyai kriteria formal dalam pembangkitan data ujinya. Pengujian ini juga telah mempunyai banyak dukungan bantuan untuk pembangkitan data uji secara otomatis.

### 2.2.17 Function Point

*Function point* digunakan untuk mengukur proyek perangkat lunak dengan mengkuantifikasi kegunaan pemrosesan informasi yang berhubungan dengan tipe berkas, keluaran, masukan data atau kontrol eksternal. Salah satu tipe fungsi pengguna dapat menggunakan pendekatan berdasar pada *data flow diagram* dan *entity relationship diagram* dimana diagram tersebut merepresentasikan penggunaan database. Data ini digunakan

untuk menentukan bobot kompleksitas berdasar aturan data *element type*, *file type references* dan *record element type*.

*Function point* terdiri dari 5 tipe fungsi pengguna[28], yaitu sebagai berikut :

1. *External Input* (EI), fungsi yang memindahkan data ke dalam aplikasi tanpa mempresentasikan manipulasi data.
2. *External Output* (EO), fungsi yang memindahkan data ke pengguna dan menyajikan beberapa manipulasi data.
3. *Internal Logical File* (ILF), fungsi yang memindahkan data ke pengguna tanpa mempresentasikan manipulasi data.
4. *External Interface Files* (EIF), logika berupa data tetap yang dikelola oleh aplikasi melalui penggunaan *External Input* (EI).
5. *External Inquiry* (EI), logika berupa data tetap yang digunakan oleh aplikasi namun tidak berjalan didalamnya.

Kelima parameter tersebut digunakan untuk menentukan bobot yang nantinya akan digunakan untuk menghitung nilai *crude function point* yang merupakan nilai *function point* sebelum disesuaikan dengan nilai *relative complexity adjustment factor* yang didapat dari 14 karakteristik perangkat lunak[29].

Tahapan dalam melakukan perhitungan *function point* adalah sebagai berikut :

1. Menghitung *crude function point*  
*Crude function point* digunakan untuk menghitung bobot dari nilai komponen-komponen *function point* yang dikaitkan dengan perangkat lunak yang akan dikembangkan[14].  
 Kemudian menghitung nilai *crude function point* yang mungkin pada setiap kategori dan setiap nilai perhitungan dikalikan dengan faktor kompleksitas yang ditunjukkan oleh tabel 2.6 berdasarkan kriteria dari setiap kategori[30].

**Tabel 2. 6 Bobot kompleksitas crude function point**

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

Penentuan bobot kompleksitas didasarkan pada jumlah tipe elemen data tiap fungsi dan jumlah referensi tipe *file*. Definisi dari masing-masing penentu bobot kompleksitas *function point* adalah sebagai berikut :

- a. *Data element type* merupakan *field* yang tak berulang dan dapat diidentifikasi oleh pengguna sebagai *field* yang unik.
- b. *Record element type* merupakan subgrup dari elemen data yang berada di dalam ILF atau EIF.
- c. *File type refrence* merupakan sebuah EIF yang dibaca oleh fungsi transaksional.

Tabel 2.7 adalah tabel perhitungan bobot kompleksitas.

**Tabel 2. 7 Perhitungan bobot kompleksitas**

<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>
2-3	<i>Low</i>	<i>Avg</i>	<i>High</i>
4+	<i>Avg</i>	<i>High</i>	<i>High</i>
<i>Internal Input</i>			
		Elemen Data	
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>

2. Menghitung *relative complexity adjustment factor*  
*Relative complexity adjustment factor* digunakan untuk menghitung bobot kompleksitas berdasarkan 14 karakteristik perangkat lunak. Penilaian kompleksitas memiliki skala antar nol sampai lima dimana nol berarti sangat sederhana dan lima yang berarti sangat kompleks atau rumit. Adapun 14 faktor tersebut adalah sebagai berikut[31] :
  1. *Data commuications* – didefinisikan sebagai tingkat kebutuhan komunikasi langsung antara aplikasi dengan *processor*.
  2. *Distributed function* – didefinisikan sebagai tingkat kebutuhan transfer data antara komponen-komponen aplikasi.
  3. *Performance objectives* – didefinisikan sebagai tingkat *response time* dan *throughput* yang perlu dipertimbangkan dalam pengembangan aplikasi.
  4. *Heavily used configuration* – didefinisikan sebagai tingkat kebutuhan dimana pengaturan konfigurasi komputer berpengaruh terhadap pengembangan aplikasi.
  5. *Transaction rate* – didefinisikan sebagai tingkat transaksi bisnis yang berpengaruh terhadap pengembangan aplikasi.

6. *Online Data Entry* – didefinisikan sebagai tingkat kebutuhan masukan data secara interaktif.
7. *End user efficiency* – didefinisikan sebagai tingkat kemudahan aplikasi.
8. *Online update* – didefinisikan sebagai tingkat kebutuhan ILF diperbaharui secara interaktif.
9. *Complex processing* – didefinisikan sebagai tingkat kesulitan logika proses yang mempengaruhi proses pengembangan.
10. *Reusability* – didefinisikan sebagai tingkat kebutuhan aplikasi dan kode program dirancang serta dikembangkan untuk bisa digunakan pada aplikasi lain.
11. *Installation Ease* – didefinisikan sebagai tingkat kemudahan konversi ke sistem baru yang berpengaruh pada proses pengembangan.
12. *Operational Ease* – didefinisikan sebagai tingkat kemudahan aplikasi dalam aspek operasional seperti *start up*, *back up* dan proses *recovery*.
13. *Multiple sites* – didefinisikan sebagai tingkat kebutuhan aplikasi dapat dioperasionalkan pada lingkungan perangkat keras dan perangkat lunak yang berbeda-beda.
14. *Facilitate change* – didefinisikan sebagai tingkat kemudahan aplikasi untuk modifikasi logika proses maupun struktur data.

Dengan penilaian sebagai berikut :

- 0 = Tidak berpengaruh
- 1 = Insidental
- 2 = Moderat
- 3 = Rata-rata
- 4 = Signifikan
- 5 = Essential

Tabel 2.8 merupakan tabel yang dapat dijadikan pedoman dalam perhitungan RCAF[10]:

**Tabel 2. 8 Karakteristik perangkat lunak**

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

Pemberian nilai bobot kepentingan dengan besar 0 sampai 5 untuk masing-masing faktor mengacu pada dokumen *Standar Documentian* untuk *General System Characteristics (GSC)* yang dipublikasikan oleh *Software Matric* yang dijelaskan oleh peneliti dalam lampiran.

### 3. Menghitung *function point*

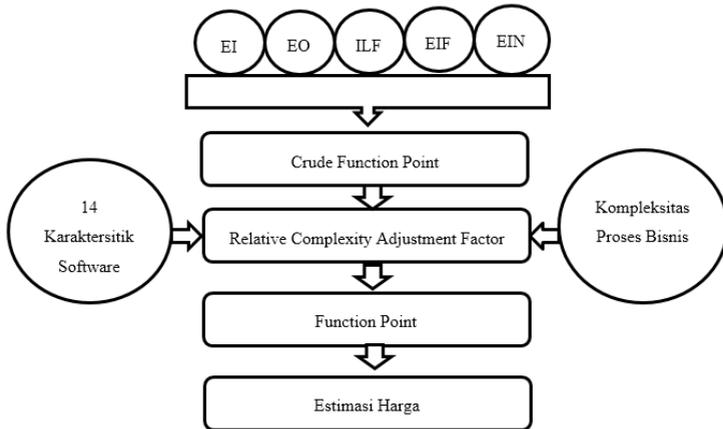
Untuk menghitung *function point* dapat digunakan persamaan satu[31]:

$$FP = \text{count total} \times [0,65 + 0,01 \times \text{Nilai kompleksitas}] \quad (1)$$

Dimana count total adalah total yang diperoleh dari tabel *function point* sementara nilai kompleksitas adalah jumlah dari 14 faktor kompleksitas yang bernilai nol sampai lima[32].

Angka 0.65 dan 0.01 adalah ketetapan atau konstanta yang dibuat oleh *Function Point International User Group* (IFPUG)[14].

#### 2.2.18 Function Point Modified



**Gambar 2. 7 Modifikasi function point**

Modifikasi pada metode *function point* yaitu dengan menambahkan kompleksitas proses bisnis pada 14 faktor *General System Characteristic*-nya[33]. Proses penambahan kompleksitas proses bisnis pada perangkat lunak didapatkan dengan cara menghitung proses bisnis menggunakan metric kompleksitas yaitu Cyclomatic.

Menurut Hannus 1994, proses bisnis adalah entitas yang terdiri dari aktivitas dan operasi terkait yang dimulai dari kebutuhan *customer* dan selesai dengan memenuhi kebutuhannya. Sebagai hasil dari kebutuhan eksternal, seperti globalisasi, kemajuan teknologi, siklus pengembangan yang lebih pendek dan persaingan yang semakin ketat, proses bisnis menjadi semakin kompleks. Untuk mengatasi kompleksitas proses bisnis intrinsic ini, pada dasarnya ada dua pendekatan yang berbeda: mengurangi kompleksitas yang tidak perlu dan mengelola kerumitan lainnya. Tujuan utamanya adalah untuk memperbaiki proses bisnis sehingga memberikan utilitas yang semakin meningkat bagi pemangku kepentingannya.

Salah satu metric kompleksitas perangkat lunak untuk mengukur kompleksitas program yaitu *cyclomatic complexity* yang dikembangkan oleh McCabe pada tahun 1976[34].

Berikut adalah langkah-langkah yang dilakukan untuk mendapatkan nilai *modified function point*[33]:

- 1) Menentukan metric kompleksitas

Langkah pertama adalah menentukan jumlah proses bisnis yang ada pada suatu perangkat lunak serta menggunakan rumus *cyclomatic* ketika sudah mendapatkan proses bisnis dari tiap fitur yang ada pada aplikasi untuk mendapatkan nilai *cyclomatic* pada proses bisnis suatu perangkat lunak. Berikut merupakan contoh perhitungan mendapatkan nilai *cyclomatic*.

Proses bisnis suatu perangkat lunak terdiri dari delapan proses bisnis, yaitu 1) *login*, 2) profil usaha, 3) mutasi keluar, 4) penutupan, 5) permohonan baru, 6) perpanjangan, 7) perubahan dan 8) status perijinan. Tabel 2.9 menjelaskan nilai kompleksitas proses bisnis pada suatu perangkat lunak.

Tabel 2. 9 Nilai kompleksitas proses bisnis

Metric	Proses Bisnis								Nilai
	1	2	3	4	5	6	7	8	
Cyclomatic	2	1	1	2	4	2	3	1	16

- 2) Mengkonversi nilai kompleksitas proses bisnis
- Setelah memperoleh nilai kompleksitas pada metric kompleksitas, tahap selanjutnya yaitu mengkonversi nilai kompleksitas tersebut dalam bentuk skala indeks. Hal tersebut agar kompleksitas proses bisnis dapat menyesuaikan dengan skala *likert* pada 14 faktor yang ada di dalam GSC (*General System Characteristic*). Dalam menentukan skalanya yaitu dengan menggunakan metode *FiveBox* yang ditemukan oleh Ferdinand[Sitasi]. Perhitungan *FiveBox Method* yaitu *range* nilai tertinggi dengan nilai terendah dari nilai kompleksitas dibagi menjadi lima dan hasilnya sebagai jarak antar indeks. Kemudian indeks awal yaitu batas nilai terendah dari sebaran data kompleksitas proses bisnis. Tabel 2.10 menunjukkan hasil konversi berdasarkan perhitungan nilai kompleksitas proses bisnis pada Tabel 2.9.

Tabel 2. 10 Konversi nilai kompleksitas kedalam indeks

Metric	Nilai Tertinggi	Nilai Terendah	Rang e	Nilai	Indeks
Cyclomatic	16	14	2	$\frac{2}{5} = 0.4$	14 – 14.4 = 1  14.5 – 14.9 = 2

Metric	Nilai Tertinggi	Nilai Terendah	Rang e	Nilai	Indeks
					15- 15.4 = 3
					15.5 – 15.9 = 4
					16 – 16.4 = 5

3) Menghitung *modified function point*

Setelah memperoleh nilai kompleksitas proses bisnis dan indeksinya selanjutnya yaitu menerapkannya pada perhitungan *function point* dengan menginputkan nilai-nilai dari hasil modifikasi pada tabel 2.11.

**Tabel 2. 11 Modifikasi karakteristik perangkat lunak**

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

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

### 2.2.19 Estimasi Effort

Salah satu aspek terpenting dalam tahapan perencanaan adalah melakukan estimasi atau perkiraan, baik dari segi biaya, waktu maupun sumber daya. Definisi dari estimasi adalah sebuah pengukuran yang didasarkan pada hasil secara kuantitatif atau dapat diukur dengan angka tingkat akurasi[35].

Sisi penting estimasi dalam perencanaan proyek adalah munculnya jadwal serta anggaran yang tepat, meski tidak sepenuhnya sebuah estimasi akan berakhir dengan tepat. Tetapi, tanpa sebuah estimasi dalam pelaksanaan proyek perangkat lunak maka dapat dikatakan bahwa proyek perangkat lunak tersebut adalah sebuah *blind project*.

Estimasi perangkat lunak yaitu suatu kegiatan melakukan prediksi atau ramalan mengenai keluaran dari sebuah proyek dengan meninjau jadwal, usaha, biaya bahkan hingga ke resiko yang akan ditanggung dalam proyek tersebut. Meski estimasi tidak mungkin dapat menghasilkan sebuah hasil yang sangat akurat, tetapi ketidakakuratan tersebut dapat diminimalkan dengan menggunakan beberapa metode yang sesuai dengan proyek yang akan dilakukan estimasi.

*Effort* adalah kerja *real* yang dilakukan dalam menyelesaikan suatu proyek dimana satuannya adalah *man-days* atau *man-*

*hour*. Misalnya suatu aplikasi diestimasi membutuhkan *effort* 10 *man-days* artinya aplikasi ini akan selesai bila dikerjakan satu orang selama sepuluh hari terus meneurus atau lima hari bila ada dua pekerja. *Effort* tidak mempertimbangkan libur ataupun cuti.

Estimasi *effort* adalah suatu kegiatan melakukan prediksi atau ramalan mengenai berapa banyak pekerja dan berapa lama waktu yang diperlukan untuk menyelesaikan proyek tersebut. Estimasi *effort* akan didapatkan setelah melakukan perhitungan menggunakan metode *function point*.

Untuk mengubah nilai FP menjadi nilai usaha maka nilai FP harus dikalikan dengan dengan nilai konstanta *Productivity Factor* (PF)[36]. Persamaan 2 mengkalikan nilai *function point* dengan nilai *productivity factor*.

$$\text{Effort} = \text{FP} \times \text{PF} \quad (2)$$

PF dikenal juga sebagai *Person-Hours*. Nilai PF ditetapkan sama dengan 20 jam kerja[37]. Beberapa penelitian lain mendasarkan nilai PF pada kompleksitas proyek dengan menetapkan nilai PF = 20 man-hours ke proyek sederhana, 24 jam kerja untuk proyek menengah, 36 jam kerja untuk proyek yang kompleks[38]. Nilai PF yang ditetapkan antara 15 hingga 30 tergantung pada kualitas personil tim pengembangan[39]. Penelitian terbaru yang dilakukan oleh Subriadi, et.al memberikan nilai PF 8,2 man-hours untuk proyek pengembangan perangkat lunak skala kecil dan menengah[40].

### **2.2.20 Estimasi Biaya**

*Effort* yang telah didapatkan pada tahap sebelumnya dibagi menjadi tiga aktivitas yang dijalankan dalam pengembangan perangkat lunak yaitu : 1) *Software development*, meliputi analisis kebutuhan pengguna seperti permintaan dan spesifikasi, desain, implementasi, pengujian integrasi yang mencapai sekitar 42% dari total *effort*, 2) *Ongoing activity*, atau disebut aktivitas yang berkesinambungan seperti manajemen proyek, manajemen konfigurasi, dokumentasi penerimaan dan

penyebaran yang mencapai sekitar 21% dari total *effort*, 3) *Quality and Testing*, aktivitas yang berhubungan dengan kualitas dan pengujian seperti pengujian integrasi penjaminan kualitas, evaluasi dan pengujian yang mencapai sekitar 37% dari total *effort*[41].

**Tabel 2. 12 Distribusi usaha dari masing-masing aktivitas**

No.	Aktivitas	Persentase (%)
1.	Software Development	
	a. Requirement	7.50%
	b. Spesification & Design	17.50%
	c. Coding	10.00%
	d. Testing	7.00%
2	Ongoing Activity	
	a. Project management	7.00%
	b. Configuration Management	3.00%
	c. Documentation	3.00%
	d. Training & Support	3.00%
	e. Acceptance & Deployment	5.00%
3	Quality and Testing	
	a. Quality Assurance & Control	12.34%
	b. Evaluation & Testing	24.66%
	Total	100%

Untuk standar gaji sendiri, mengacu pada Indonesian Sallary Guide yang dikeluarkan oleh Kelly Services. Tabel 2.13 adalah daftar rata-rata gaji yang dikeluarkan oleh Kelly 2017[42].

**Tabel 2. 13 Standar gaji tenaga IT**

Jabatan	Pendidikan/Pengalaman	Gaji/bulan (Rp)
<i>Analyst Programmer/Software Engineer</i>	Sarjana (3 – 6 tahun)	Rp.5.000.000

Jabatan	Pendidikan/Pengalaman	Gaji/bulan (Rp)
<i>Business/System Analyst</i>	Sarjana (3 – 6 tahun)	Rp.7.000.000
<i>Project Manager</i>	Sarjana (5 – 10 tahun)	Rp.20.000.000
<i>Software QA/test analyst</i>	Sarjana (3 – 5 tahun)	Rp.8.000.000
<i>Secretary</i>	Sarjana (1 – 3 tahun)	Rp.4.000.000

Untuk mengalikan dengan nilai *effort*, gaji tiap bulan pada masing-masing profesi diubah kedalam bentuk jam. Nilai gaji tersebut kemudian dikalikan dengan nilai *effort* untuk mengetahui nilai biaya. Adapun profesi diatas ditentukan berdasarkan suatu tim dalam proyek pengembangan perangkat lunak yang memiliki deskripsi pekerjaan pada tabel 2.14.

**Tabel 2. 14 Tim pengembang perangkat lunak**

No.	Jabatan Tim	Jobdesk
1.	Developer	Bertugas dalam pembuatan aplikasi dan melayani dalam memberikan solusi IT yang tepat kepada <i>customer</i> .
2.	System Analyst	Bertugas untuk membantu dalam menyiapkan data SKPL awal untuk kebutuhan dokumen penawaran proyek
3.	Project Manager	Bertugas untuk mengarahkan sekaligus mengawasi jalannya proyek serta memberikan tugas kepada tim proyek sesuai <i>jobdesk</i> -nya masing-masing
4.	Software Quality Assurance	Bertugas dalam mengkontrol mengenai kualitas dari <i>software</i> yang dibuat.
5.	Secretary	Bertugas untuk mengurus dalam pembuatan dokumen penawaran

Untuk mendapatkan biaya total keseluruhan proyek pengembangan perangkat lunak. Harga yang didapatkan dari distribusi usaha ditambahkan dengan biaya operasional dari proyek tersebut. Biaya operasional yang dimaksud adalah biaya *non-personil direct cost* berupa biaya *fix unit rate*. Tabel 2.15 menjelaskan mengenai macam dari biaya *fix unit rate* dalam pengembangan perangkat lunak[43].

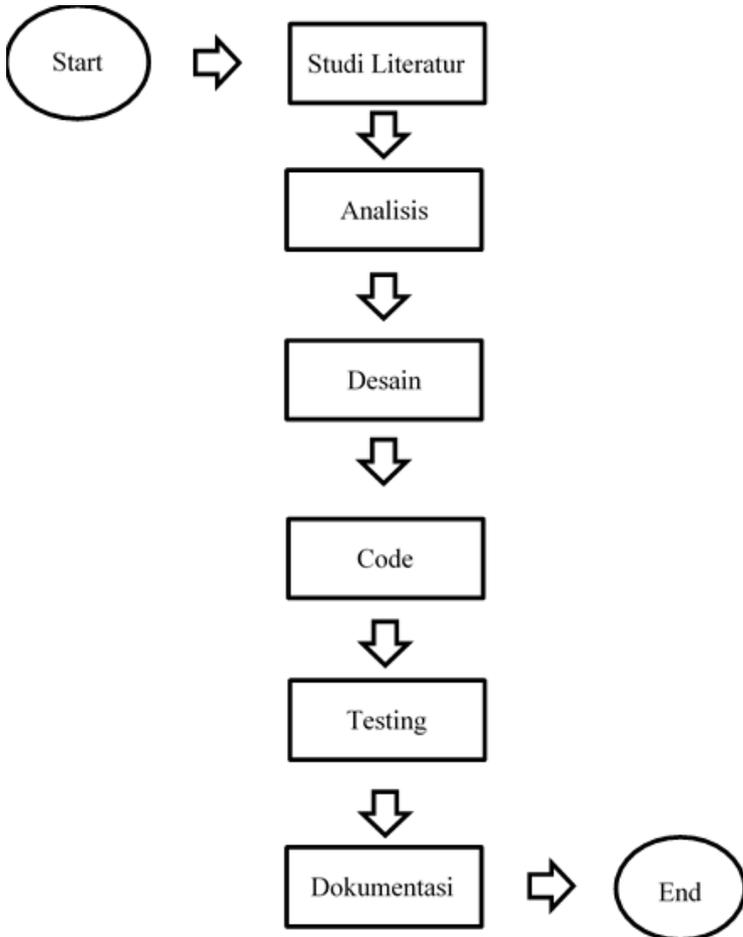
**Tabel 2. 15 Kategori biaya operasional**

Jenis Pengeluaran	Satuan	Harga (RP)
Biaya Operasional Kantor Proyek	Bulan	16.000.000
Biaya ATK	Bulan	8.000.000
Biaya Komputer dan Printer	Unit-Bulan	5.000.000
Biaya Komunikasi	Bulan	7.750.000

*Halaman ini sengaja dikosongkan*

### BAB III METODOLOGI PENELITIAN

Metodologi dalam penelitian tugas akhir diperlukan sebagai panduan dalam pengerjaan tugas akhir agar tahapan dalam pengerjaan tugas akhir berjalan secara terarah dan sistematis seperti berikut:



Gambar 3. 1 Metodologi

### 3.1 Tahap Analisis

Tahapan ini dilakukan setelah melakukan tahap perencanaan dimana pembuat aplikasi harus menganalisis kebutuhan aplikasi terlebih dahulu. Analisis kebutuhan dapat dilakukan dengan cara pengumpulan data. Adapun masukan dari tahapan ini adalah kertas kerja untuk mengetahui kebutuhan sistem dan catatan untuk pengamatan proses bisnis. Hal yang dilakukan dalam tahapan ini adalah melakukan analisis kertas kerja, melakukan analisis kebutuhan pengguna dan analisis kebutuhan aplikasi. Luaran dari tahapan ini adalah berupa dokumen SKPL.

### 3.2 Tahap Desain

Tahapan ini berfokus pada perancangan arsitektur perangkat lunak. Perancangan meliputi pembuatan *sequence diagram*, *class diagram* dan *activity diagram* dan desain antarmuka pengguna. Adapun masukan pada tahapan ini adalah model proses bisnis dari hasil analisa kebutuhan. Hal yang dilakukan pada tahapan ini adalah membuat desain proses bisnis ke dalam bentuk *sequence diagram*, *class diagram* dan *activity diagram* dan desain antarmuka pengguna. Luaran dari tahapan ini adalah berupa dokumen yang berisi *sequence diagram*, *class diagram* dan *activity diagram* dan desain antarmuka pengguna.

### 3.3 Tahap Coding

Tahapan ini, peneliti dapat mengetahui alur proses bisnis pada aplikasi berdasarkan hasil tahap desain. Tahapan ini merupakan tahapan pembuatan aplikasi dengan cara pembuatan kode atau *script*. Pembuatan kode atau *script* menggunakan bahasa PHP dan menggunakan *framework* Codeigniter sehingga nantinya akan terdapat tiga jenis *file* yang perlu dikerjakan yaitu *model*, *view* dan *controller*. Pembuatan kode atau *script* pada *view* untuk tampilan pengguna *project manager*, analis dan sekretaris. Kemudian *controller* untuk menghubungkan antara *view* dan *model*. Model digunakan untuk memanggil data dari *database* yang berisi *query*. Adapun masukan dari tahapan ini

adalah hasil dari desain. Hal yang dilakukan dalam tahapan ini adalah mengimplementasikan hasil rancangan desain ke dalam kode pemrograman. Luaran yang dihasilkan dari tahapan ini adalah aplikasi estimasi biaya perangkat lunak menggunakan metode *function point* yang belum diuji.

### **3.4 Tahap Testing**

Tahapan ini digunakan untuk menemukan *error* pada aplikasi yang telah dibuat. *Testing* dilakukan dengan cara menguji setiap fitur-fitur yang ada pada aplikasi estimasi biaya perangkat lunak ini. Pengujian fitur menggunakan *Unit Testing*. Adapun masukan dari tahapan ini adalah aplikasi estimasi biaya perangkat lunak menggunakan metode *function point* yang belum diuji. Hal yang dilakukan pada tahap ini adalah melakukan uji coba terhadap fitur yang ada pada aplikasi ini menggunakan *unit testing*. Luaran dari tahapan ini adalah menghasilkan aplikasi estimasi biaya perangkat lunak yang setiap fiturnya sudah teruji.

### **3.5 Tahap Dokumentasi**

Tahapan ini digunakan untuk menyusun atau membuat dokumentasi rancang bangun aplikasi estimasi biaya perangkat, semua kegiatan pengerjaan dari tahapan awal analisis kebutuhan, desain, code dan *testing* disusun dalam satu bentuk dokumen yaitu laporan atau buku tugas akhir. Penyusunan dokumen tugas akhir ini bertujuan untuk pendokumentasian dan bukti bahwa semua aktivitas telah dilakukan mulai dari awal hingga akhir.

*Halaman ini sengaja dikosongkan*

## **BAB IV PERANCANGAN**

Pada bab ini akan dijelaskan mengenai perancangan sistem sesuai dengan metode penelitian tugas akhir yang telah ditentukan. Setelah dilakukan studi literature maka dilanjutkan dengan tahapan yang ada pada metode *waterfall* yaitu analisis, desain, membuat kode program dan melakukan uji coba. Pada tahapan analisis diperoleh informasi mengenai gambaran aplikasi dan pengguna aplikasi. Gambaran aplikasi yang diperoleh kemudian dikonversikan kedalam bentuk *usecase diagram*. Pada tahap analisis juga menghasilkan kebutuhan fungsional. Pada tahap kedua, yaitu desain dimana hasil gambaran informasi dari kebutuhan akan mulai dilakukan desain aplikasi yang meliputi *sequence diagram*, *class diagram*, *activity diagram* dan desain antarmuka pengguna. Tahap ketiga yaitu mengimplementasikan hasil rancangan desain kedalam kode pemrograman menggunakan bahasa pemrograman PHP dan *framework* Codeigniter. Tahap terakhir adalah pengujian semua fitur yang ada pada aplikasi estimasi biaya perangkat lunak menggunakan metode *function point* ini dengan menggunakan *unit testing*.

Berdasarkan uraian tersebut, perancangan sistem secara umum meliputi gambaran aplikasi, pengguna aplikasi, kebutuhan aplikasi, desain aplikasi dan uji coba aplikasi. Perancangan aplikasi yang dijelaskan dalam bab ini terbatas pada bagian utama dari aplikasi karena untuk penjelasan secara lebih rinci tentang perancangan aplikasi secara keseluruhan dapat dilihat pada lampiran.

### **4.1 Tahapan Analisis**

Tahapan analisis merupakan tahapan yang dilalui oleh peneliti dalam membuat aplikasi estimasi biaya perangkat lunak setelah mencari dan mempelajari literatur terkait penelitian. Berikut adalah analisis aplikasi estimasi biaya perangkat lunak menggunakan metode *function point*.

### 4.1.1 Analisis Kebutuhan Pengguna

Dalam membangun aplikasi, proses ini menjadi bagian pertama yang dilakukan oleh peneliti dengan menggali kebutuhan pengguna di dalam aktivitas proyek pengembangan perangkat lunak. Proses ini bertujuan agar peneliti dapat melakukan penggalian kebutuhan mengenai aplikasi yang akan dikembangkan nantinya. Hal yang pertama kali dilakukan adalah melakukan identifikasi kebutuhan yang akan diimplementasikan ke fitur aplikasi nantinya. Hasil analisis kebutuhan pengguna akan dijadikan bahan untuk menganalisis kebutuhan fungsional dari aplikasi yang dibuat. Berikut adalah hal yang perlu dilakukan sebelum melakukan analisis aplikasi.

#### 1. Menentukan Pengguna Aplikasi

Dalam sebuah proyek dikerjakan oleh beberapa orang yang terbentuk dalam suatu tim. Suatu tim dalam proyek khususnya dalam proyek teknologi informasi terdiri dari seorang *project manager*, *programmer*, *system analyst*, *quality assurance* dan *secretary* dimana setiap profesi memiliki tugas masing-masing dalam suatu proyek pengembangan perangkat lunak.

#### 2. Melakukan Analisis Kebutuhan Pengguna

##### a. Translasi Hasil Analisis Kedalam Kebutuhan

Tabel 4. 1 Translasi kebutuhan

Kode	Point of Analysis	Requirement
AI-01-01	<p><b>Aplikasi mudah diakses dimanapun</b></p> <p>Aplikasi yang akan dibuat nanti harus mudah diakses oleh pengguna dari aplikasi ini</p>	Aplikasi berbasis <i>website</i>

Kode	Point of Analysis	Requirement
AI-01-02	<p><b>Estimasi harga lebih mudah</b></p> <p>Tim pengembang bisa melakukan estimasi harga perangkat lunak lebih akurat dari yang biasanya dilakukan</p>	<p>Estimasi biaya perangkat lunak menggunakan metode <i>function point</i> karena metode ini memiliki tingkat akurasi yang cukup baik. Jadi fitur yang ada sebagai berikut.</p> <ol style="list-style-type: none"> <li>1. Fitur Perhitungan CFP (Crude Function Point)</li> <li>2. Fitur Perhitungan RCAF (Relative Complexity Adjustment Factor)</li> <li>3. Fitur Perhitungan FP (Function Point)</li> <li>4. Fitur Perhitungan Biaya berdasarkan nilai <i>Function Point</i> menggunakan distribusi usaha</li> </ol>

Kode	Point of Analysis	Requirement
AI-01-03	<p><b>Estimasi Aplikasi nanti bisa digunakan oleh orang yang ingin mengembangkan sebuah aplikasi dan mengetahui estimasi harga aplikasi tersebut</b></p> <p>Aplikasi nanti bisa digunakan oleh semua orang yang ingin mengembangkan suatu aplikasi dan mendapatkan estimasi harga dari aplikasi yang telah dibuat</p>	Fitur <i>login</i> dan <i>logout</i>
AI-01-04	<p><b>Pengubahan data dari proses estimasi</b></p> <p>Dalam proses estimasi data hasil perangkat lunak, dimungkinkan ada kesalahan pada saat proses estimasi, oleh karena itu perlu adanya fitur untuk mengubah data estimasi.</p>	Fitur CRUD (Create, Read, Update, Delete) dari masing-masing fitur yang ada pada <b>AI-01-02</b>

Kode	Point of Analysis	Requirement
AI-01-05	<p><b>Penambahan pengguna dari proyek pembuatan aplikasi</b></p> <p>Adanya penambahan pengguna yang ditambahkan oleh seorang proyek manajer terhadap proyek estimasi aplikasi</p>	Fitur Tambah Pengguna
AI-01-06	<p><b>Penambahan Anggota Tim Pengembang</b></p> <p>Adanya penambahan tim dari anggota pengembang berdasarkan profesi masing-masing orang</p>	Fitur Tambah Anggota Tim
AI-01-07	<p><b>Melihat Daftar Aktivitas</b></p> <p>Aktivitas yang dilakukan pada saat pembuatan suatu aplikasi untuk dilakukan pemantauan aktivitas</p>	Fitur Melihat Daftar Aktivitas

Kode	Point of Analysis	Requirement
	berdasarkan proyek tersebut	
AI-01-08	<p><b>Mengubah pelaku aktivitas</b></p> <p>Pelaku yang ada pada masing-masing aktivitas dapat dilakukan perubahan sesuai dengan perkembangan yang ada</p>	Fitur perubahan pelaku aktivitas
AI-01-09	<p><b>Tambah Profesi</b></p> <p>Adanya perbedaan dari tiap proyek yang memiliki kebutuhan profesi yang berbeda-beda</p>	Fitur Tambah Profesi
AI-01-10	<p><b>Biaya estimasi berdasarkan gaji masing-masing profesi</b></p> <p>Estimasi biaya perangkat lunak didasarkan dari gaji profesi tim pengembangan perangkat lunak</p>	Fitur Entry Gaji Profesi
AI-01-11	<p><b>Gaji yang berubah-ubah</b></p> <p>Gaji ada yang ada pada setiap profesi</p>	Fitur perubahan gaji masing-masing profesi

Kode	Point of Analysis	Requirement
	bisa berubah-ubah sesuai dengan tingkat inflasi yang ada di Indonesia, perubahan dilakukan oleh project manager	
AI-01-12	<p><b>Pemberitahuan hasil estimasi</b></p> <p>Untuk memudahkan validasi, proyek manager akan mendapatkan informasi melalui notifikasi</p>	Fitur Notifikasi email yang dilakukan oleh proyek manager
AI-01-13	<p><b>Melihat rekap hasil biaya estimasi</b></p> <p>Pengguna aplikasi bisa melihat hasil rekap estimasi yang dilakukan oleh tim pengembang</p>	Fitur Rekap hasil biaya estimasi
AI-01-14	<p><b>Mengubah biaya hasil estimasi</b></p> <p>Proyek manager mempunyai kuasa dalam melakukan perubahan terhadap biaya dari hasil estimasi yang</p>	Fitur perubahan biaya dari hasil estimasi perangkat lunak

Kode	Point of Analysis	Requirement
	sudah dilakukan oleh tim pengembang	
AI-01-15	<p><b>Mencetak dokumen laporan</b></p> <p>Setelah proyek manajer melakukan validasi, sekretaris bisa melakukan pencetakan dokumen penawaran harga yang akan diserahkan ke client</p>	Fitur cetak dokumen penawaran
AI-01-16	<p><b>Validasi hasil estimasi</b></p> <p>Proyek manajer mempunyai hak dalam menolak maupun menerima hasil estimasi yang dilakukan oleh tim pengembang</p>	Fitur validasi hasil estimasi
AI-01-17	<p><b>Fitur otomatisasi nilai Effort Rate</b></p> <p>Nilai ER yang digunakan untuk mendapatkan nilai usaha bersifat dinamis</p>	Fitur nilai ER

Kode	Point of Analysis	Requirement
	berdasarkan historis estimasi	
AI-01-18	<p><b>Fitur melakukan feedback estimasi</b></p> <p>Setiap hasil estimasi akan dilakukan dengan feedback dengan cara merevisi masing-masing komponen perhitungan estimasi</p>	Refine masing-masing komponen perhitungan function point

#### b. Karakteristik dan Klasifikasi Pengguna

Setelah melakukan translasi kebutuhan pengguna, peneliti dapat menggunakan translasi untuk menentukan seperti apa sistem tersebut nantinya. Sebelum masuk ke tahap berikutnya, pada bagian ini akan dijelaskan siapa saja pengguna yang terlibat dalam aplikasi estimasi biaya perangkat lunak. Tujuan peneliti mengetahui siapa saja pengguna yang terlibat supaya lebih memahami peran apa saja yang dapat dikerjakan oleh setiap pengguna sehingga akan memudahkan dalam memetakan setiap aktivitasnya serta agar lebih mudah dalam membuat *usecase diagram*. Pengguna aplikasi dibagi menjadi tiga yaitu *project manager*, analis dan sekretaris. Tabel 4.2 menjelaskan beberapa aktivitas yang bisa dilakukan oleh pengguna dalam aplikasi.

**Tabel 4. 2 Deskripsi Pengguna**

Pengguna	Nama Aktivitas	Penjelasan
<i>Project Manager</i>	Melakukan estimasi biaya	<i>Project manager</i> dapat melakukan

	perangkat lunak	estimasi biaya perangkat lunak dengan cara mengisi form informasi client, deskripsi aplikasi, CFP dan RCAF
	Mengubah hasil estimasi perangkat lunak	<i>Project manager</i> dapat mengubah hasil estimasi biaya perangkat lunak dengan cara mengubah nilai di form CFP dan RCAF.
	Mengubah biaya estimasi perangkat lunak	<i>Project manager</i> dapat mengubah hasil biaya dengan cara menekan <i>edit</i> pada rekap biaya aktivitas serta menambahkan biaya operasional.
	Mengelola akun pengguna	<i>Project manager</i> dapat menambahkan, mengubah serta menghapus pengguna yang ada pada aplikasi beserta peran masing-masing pengguna
	Mengelola daftar profesi	<i>Project manager</i> dapat menambahkan, mengubah serta menghapus daftar profesi beserta gaji yang ada pada setiap profesi

	Mengelola daftar anggota tim	<i>Project manager</i> dapat menambahkan, mengubah serta menghapus anggota tim pada aplikasi
	Mengelola daftar aktivitas	<i>Project manager</i> dapat mengubah daftar aktivitas yang ada pada proyek tersebut beserta profesi yang berperan terhadap aktivitas tersebut
	Melakukan validasi hasil estimasi	<i>Project manager</i> dapat melakukan validasi hasil estimasi dengan menekan tombol validasi setelah dikirimkan hasil estimasi oleh analis
Analis	Melakukan estimasi biaya perangkat lunak	Analis dapat melakukan estimasi biaya perangkat lunak dengan mengisi form informasi client, deskripsi aplikasi, CFP dan RCAF
	Mengubah hasil estimasi perangkat lunak	Analis dapat melakukan perubahan hasil estimasi dengan merubah nilai CFP dan RCAF
	Mengubah hasil biaya estimasi	Analis dapat melakukan perubahan hasil

	perangkat lunak	biaya dengan menekan edit pada tabel rekap biaya serta menambahkan biaya operasional.
	Mengirim hasil estimasi	Analisis dapat mengirim hasil estimasi kepada <i>project manager</i> untuk divalidasi dengan menekan tombol kirim validasi setelah melakukan estimasi
Sekretaris	Mencetak dokumen penawaran	Sekretaris dapat mencetak dokumen penawaran setelah estimasi di validasi oleh <i>project manager</i> dengan menekan tombol <i>print</i> pada aplikasi yang telah di estimasi.

**c. Hak Akses Pengguna**

Pada tahap ini terdapat fitur untuk melakukan pembedaan hak akses pengguna aplikasi. Berdasarkan deskripsi pengguna menunjukkan bahwa dalam aplikasi nanti terdapat tiga aktor yang menggunakan aplikasi dimana masing-masing aktor memiliki hak akses (*previlage*) yang berbeda. Tabel 4.3 menjelaskan detail dari hak akses pengguna aplikasi estimasi biaya perangkat lunak menggunakan metode *function point* ini.

Tabel 4. 3 Hak akses pengguna

Nama Peran	Peran
Project Manager	<ul style="list-style-type: none"> <li>• Melakukan estimasi biaya perangkat lunak</li> <li>• Melakukan perubahan hasil estimasi</li> <li>• Melakukan perubahan estimasi biaya perangkat lunak</li> <li>• Mengelola akun pengguna</li> <li>• Mengelola data anggota tim</li> <li>• Mengelola data aktivitas</li> <li>• Mengelola data profesi</li> <li>• Melakukan validasi hasil estimasi</li> </ul>
Analisis	<ul style="list-style-type: none"> <li>• Melakukan estimasi biaya perangkat lunak</li> <li>• Melakukan perubahan hasil estimasi</li> <li>• Melakukan perubahan estimasi biaya perangkat lunak</li> <li>• Mengirim hasil estimasi ke project manager</li> </ul>
Sekretaris	Melakukan pencetakan dokumen

## 4.1.2 Analisis Kebutuhan Aplikasi

### a. Deskripsi Umum Perangkat Lunak

Perangkat lunak yang dikembangkan adalah perangkat lunak untuk melakukan estimasi biaya perangkat lunak dengan menggunakan metode *function point* dengan adanya penambahan kompleksitas proses bisnis dan menggunakan estimasi usaha serta distribusi usaha untuk mendapatkan harga perangkat lunak. Tujuan pembuatan perangkat lunak yaitu sebagai solusi dalam mengatasi permasalahan suatu tim proyek dalam hal melakukan estimasi biaya pembuatan perangkat lunak.

### b. Membuat Daftar Kebutuhan Sistem

Daftar kebutuhan sistem dibuat berdasarkan hasil analisis kebutuhan pengguna. Daftar kebutuhan aplikasi ini nantinya akan dijadikan acuan dalam pembuatan dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Tabel 4.4 merupakan daftar kebutuhan yang ada pada aplikasi estimasi biaya perangkat lunak menggunakan metode *function point*.

**Tabel 4. 4** Daftar kebutuhan sistem aplikasi estimasi biaya perangkat lunak

Kode	Nama Fitur	Penjelasan
KFA01	Fitur Autentifikasi	Pada fitur autentifikasi dibagi menjadi dua aktivitas yang dapat dilakukan yaitu <i>login</i> dan <i>logout</i> .
KFA02	Fitur Pembedaan Hak Akses Pada Aplikasi	Pada fitur pembedaan hak akses membuat pengguna dengan masing-masing <i>username</i> masuk ke halaman yang sama tetapi memiliki menu yang berbeda.
KFA03	Fitur Homepage	Pada fitur homepage terdapat tulisan selamat datang terhadap pengguna serta menampilkan menu-menu yang ada pada aplikasi.
KFA04	Fitur Create Estimation	Pada fitur create estimation terdapat lima tab yang berisi informasi client, deskripsi aplikasi, CFP, RCAF dan result dimana pengguna perlu

Kode	Nama Fitur	Penjelasan
		melakukan pengisian di setiap form pada tab yang ada.
KFA05	Fitur Log Estimasi	Pada fitur log estimasi terdapat riwayat aplikasi yang sudah pernah di estimasi dimana tampilan berbentuk tabel sehingga pengguna dapat melihat secara lebih detail.
KFA06	Fitur Tambah Pengguna	Fitur ini digunakan oleh <i>project manager</i> untuk menambah pengguna pada aplikasi dengan <i>role</i> yang sudah ada.
KFA07	Fitur Daftar Pengguna	Fitur ini digunakan oleh <i>project manager</i> untuk melihat daftar pengguna pada aplikasi dimana <i>project manager</i> dapat menghapus serta mengubah daftar pengguna dimana tampilan halaman ini berbentuk tabel sehingga lebih detail.
KFA08	Fitur Tambah Profesi	Fitur ini digunakan oleh <i>project manager</i> untuk bisa menambahkan profesi yang ada dalam proyek tersebut.
KFA09	Fitur Daftar Profesi	Fitur ini digunakan oleh <i>project manager</i> untuk melihat daftar profesi yang ada pada proyek

Kode	Nama Fitur	Penjelasan
		dimana dapat diubah serta dihapus mengenai nama profesi dan daftar gaji.
KFA10	Fitur Tambah Anggota Tim	Fitur ini digunakan oleh <i>project manager</i> untuk menambah anggota tim sesuai dengan profesi yang sudah ada dalam aplikasi.
KFA11	Fitur Daftar Anggota Tim	Fitur ini digunakan oleh <i>project manager</i> untuk melihat daftar anggota tim dimana dapat diubah serta dihapus dan halaman ini berbentuk tabel sehingga dapat terlihat secara lebih detail.
KFA12	Fitur Daftar Aktivitas	Fitur ini digunakan oleh <i>project manager</i> untuk dapat melihat daftar aktivitas dari proyek dan bisa diganti profesi yang menjalankan aktivitas tersebut.
KFA13	Fitur Edit Profile	Fitur ini digunakan untuk mengganti nama, <i>username</i> , <i>password</i> dan <i>email</i> dari pengguna
KFA14	Fitur Mengirim Hasil Estimasi	Fitur ini digunakan oleh analis untuk mengirim hasil estimasi kepada <i>project manager</i> dengan menekan tombol kirim estimasi pada bagian tab

Kode	Nama Fitur	Penjelasan
		<i>result</i> di menu <i>create estimaton</i> .
KFA15	Fitur Notifikasi Email	Fitur ini digunakan untuk memberikan notifikasi <i>email project manager</i> dari hasil kirim estimasi yang dilakukan oleh analis sehingga <i>project manager</i> dapat mengetahui tim proyek telah melakukan estimasi.
KFA16	Fitur Validasi Hasil Estimasi	Fitur ini digunakan oleh <i>project manager</i> untuk memvalidasi hasil estimasi yang telah dilakukan oleh analis apabila sudah sesuai dengan yang diharapkan.
KFA17	Fitur Cetak Dokumen Penawaran	Fitur ini digunakan oleh sekretaris untuk mencetak dokumen penawaran dari hasil estimasi yang sudah dilakukan oleh analis dan sudah di validasi oleh <i>project manager</i> .
KFA18	Fitur Memilih Effort Rate	Fitur ini ada pada tab deskripsi aplikasi pada menu <i>create estimation</i> dimana fitur ini digunakan untuk menentukan rata-rata usaha yang digunakan untuk mendapatkan nilai total usaha proyek.

<b>Kode</b>	<b>Nama Fitur</b>	<b>Penjelasan</b>
KFA19	Fitur Memasukkan Daftar Fitur Aplikasi	Fitur ini muncul ketika sedang berada di menu <i>create estimation</i> dan sudah pada tahap deskripsi aplikasi dan pengguna telah menyimpan deskripsi aplikasi dimana fitur ini digunakan untuk memberikan informasi dari fitur apa saja yang ada pada aplikasi tersebut.
KFA20	Fitur Memasukkan Biaya Operasional	Fitur ini digunakan ketika sudah pada tahap <i>result</i> melakukan pembuatan estimasi dimana fitur ini digunakan untuk memasukkan biaya lain diluar aktivitas proyek.
KFA21	Fitur Refine Nilai Function Point	Fitur ini digunakan ketika hasil function point sudah didapatkan tetapi terjadi kesalahan dalam memasukkan data sehingga dapat diubah dengan masuk kembali ke tab CFP dan RCAF
KFA22	Fitur Memasukkan Actual Effort	Fitur ini digunakan oleh analis ketika proyek pengembangan perangkat lunak sudah selesai untuk mengetahui perbandingan estimasi di

Kode	Nama Fitur	Penjelasan
		awal dengan usaha yang dilakukan.

### c. Kebutuhan Fungsional Sistem

Kebutuhan fungsional dari aplikasi estimasi biaya perangkat lunak dengan adanya penambahan kompleksitas proses bisnis sudah dijelaskan sebelumnya pada daftar kebutuhan, tabel 4.5 menjelaskan kebutuhan fungsional dari aplikasi estimasi biaya perangkat lunak dengan adanya penambahan kompleksitas proses bisnis.

**Tabel 4. 5 Daftar kebutuhan fungsional aplikasi estimasi biaya perangkat lunak**

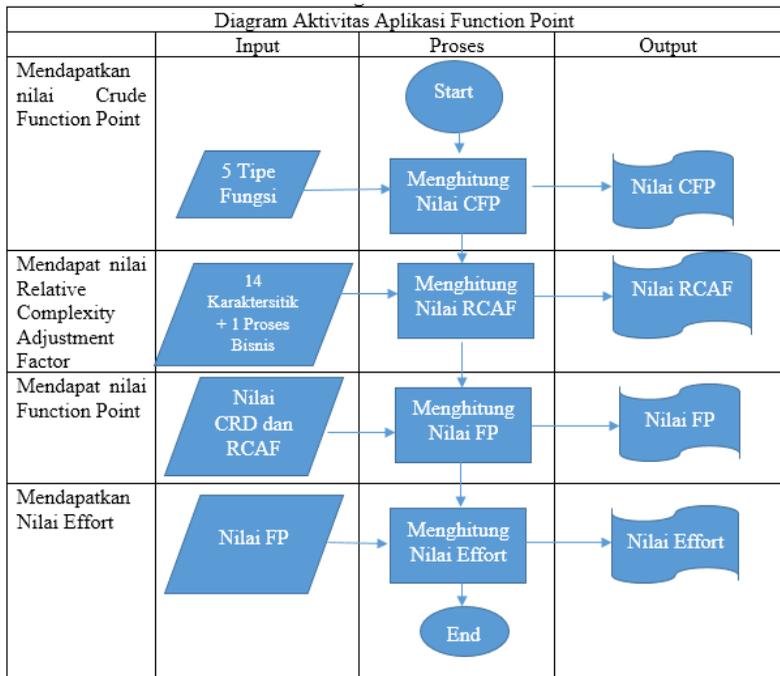
Kode	Daftar Kebutuhan Fungsional
<i>Fitur Project Manager, Analis dan Sekretaris</i>	
KFA01	Fitur Autentifikasi (Login & Logout)
KFA02	Fitur Pembedaan Hak Akses
KFA03	Fitur Homepage
KFA13	Fitur Edit Profile
<i>Fitur Project Manager dan Analis</i>	
KFA01	Fitur Autentifikasi (Login & Logout)
KFA02	Fitur Pembedaan Hak Akses
KFA03	Fitur Homepage
KFA04	Fitur Create Estimation
KFA05	Fitur Log Estimation
KFA13	Fitur Edit Profile
KFA18	Fitur Memilih Effort Rate
KFA19	Fitur Memasukkan Daftar Fitur Aplikasi
KFA20	Fitur Memasukkan Biaya Operasional
KFA21	Fitur Refine Nilai Function Point
<i>Fitur Project Manager</i>	
KFA01	Fitur Autentifikasi (Login & Logout)
KFA02	Fitur Pembedaan Hak Akses
KFA03	Fitur Homepage
KFA04	Fitur Create Estimation

KFA05	Fitur Log Estimation
KFA06	Fitur Tambah Pengguna
KFA07	Fitur Daftar Pengguna
KFA08	Fitur Tambah Profesi
KFA09	Fitur Daftar Profesi
KFA10	Fitur Tambah Anggota Tim
KFA11	Fitur Daftar Anggota Tim
KFA12	Fitur Daftar Aktivitas
KFA13	Fitur Edit Profile
KFA15	Fitur Notifikasi Email
KFA16	Fitur Validasi Hasil Estimasi
KFA18	Fitur Memilih Effort Rate
KFA19	Fitur Memasukkan Daftar Fitur Aplikasi
KFA20	Fitur Memasukkan Biaya Operasional
KFA21	Fitur Refine Nilai Function Point
<i>Fitur Analis</i>	
KFA01	Fitur Autentifikasi (Login & Logout)
KFA02	Fitur Pembendaan Hak Akses
KFA03	Fitur Homepage
KFA04	Fitur Create Estimation
KFA05	Fitur Log Estimation
KFA13	Fitur Edit Profile
KFA14	Fitur Kirim Hasil Estimasi
KFA18	Fitur Memilih Effort Rate
KFA19	Fitur Memasukkan Daftar Fitur Aplikasi
KFA20	Fitur Memasukkan Biaya Operasional
KFA21	Fitur Refine Nilai Function Point
KFA22	Fitur Memasukkan Actual Effort
<i>Fitur Sekretaris</i>	
KFA01	Fitur Autentifikasi (Login & Logout)
KFA02	Fitur Pembendaan Hak Akses
KFA03	Fitur Homepage
KFA13	Fitur Edit Profile
KFA18	Fitur Cetak Dokumen Penawaran

#### d. Alur Proses Bisnis Aplikasi

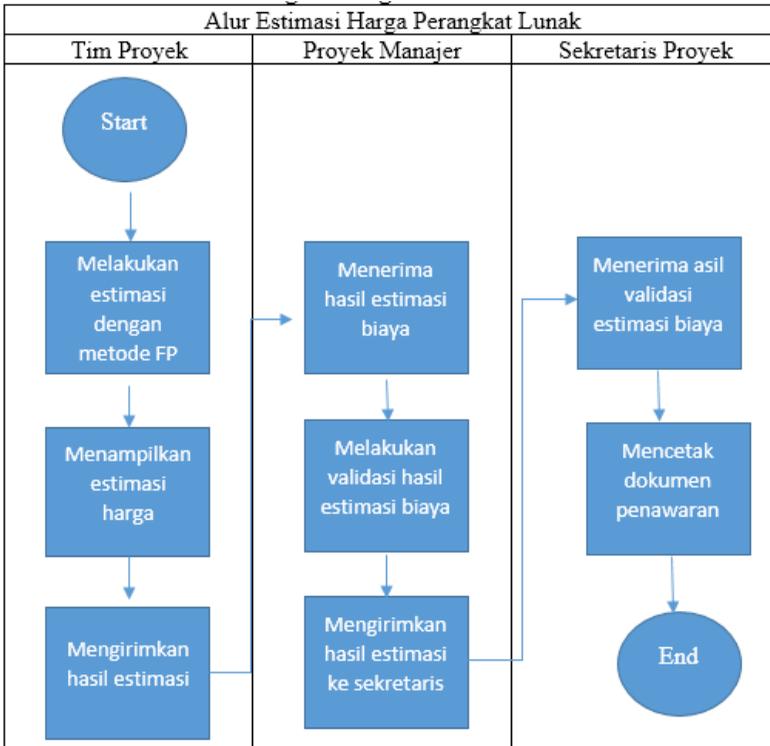
##### 1. Alur Proses Estimasi dengan Metode Function Point

Dalam melakukan pembuatan sistem akan lebih mudah jika kita sudah mengetahui bagaimana alur proses bisnis dari metode *function point* dan mengestimasi harga perangkat lunak. Gambar 4.1 dan Gambar 4.2 menjelaskan mengenai alur proses bisnis dari metode *function point* dan proses dalam melakukan estimasi harga perangkat lunak.



Gambar 4. 1 Alur estimasi perhitungan usaha metode function point

## 2. Alur Proses Estimasi Harga Perangkat Lunak



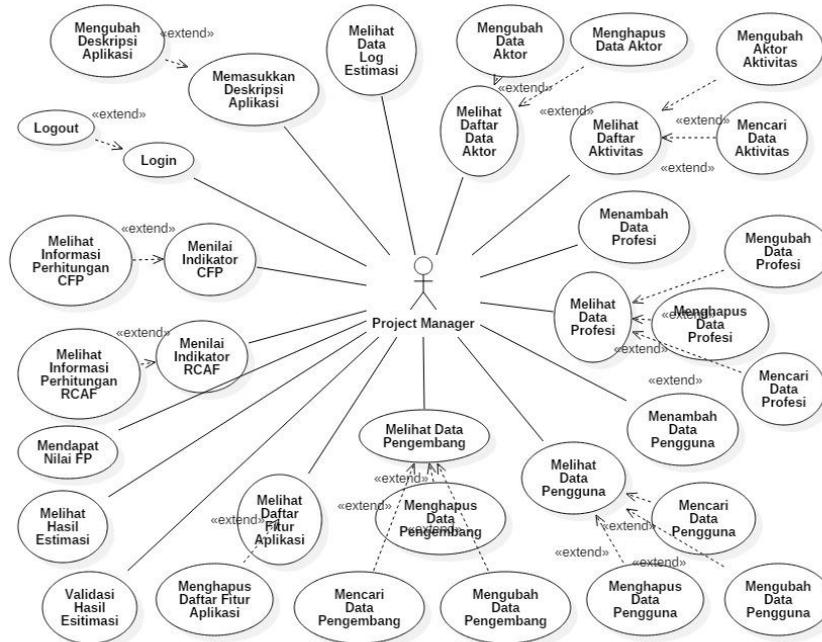
Gambar 4. 2 Alur estimasi harga perangkat lunak

### 4.1.3 Use Case Diagram

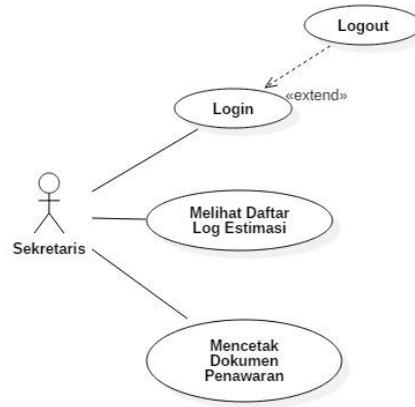
Use case diagram atau diagram use case merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Dalam use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut uraian singkat dan use case diagram berdasarkan aktor dan fitur yang ada di dalam aplikasi estimasi biaya perangkat lunak dengan adanya penambahan kompleksitas proses bisnis.

**a. Use case Diagram Aktor**

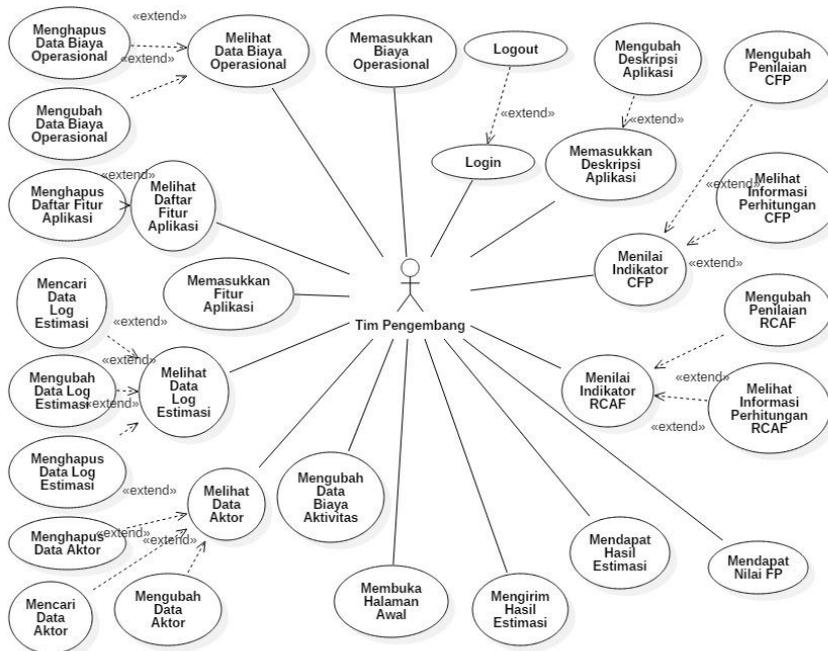
Use case diagram berdasarkan aktor merupakan hal-hal yang bisa dilakukan oleh masing-masing aktor dalam aplikasi estimasi biaya perangkat lunak dengan adanya penambahan kompleksitas proses bisnis pada faktor kompleksitas. Dalam aplikasi terdapat tiga aktor yaitu *project manager*, analis dan sekretaris dimana masing-masing aktor memiliki use case masing-masing. Berikut adalah use case diagram berdasarkan aktor.



Gambar 4. 3 Desain use case pada aktor project manager



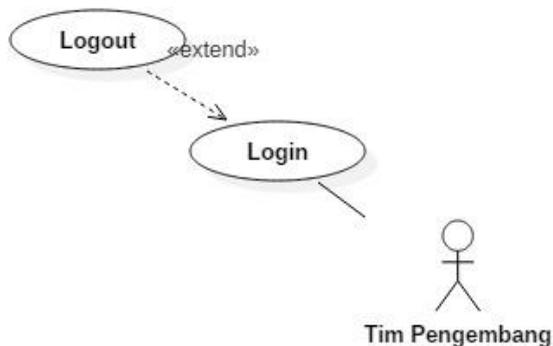
**Gambar 4. 4 Desain use case pada aktor sekretaris**



Gambar 4. 5 Desain use case pada aktor analis

### b. Use case Diagram Fitur

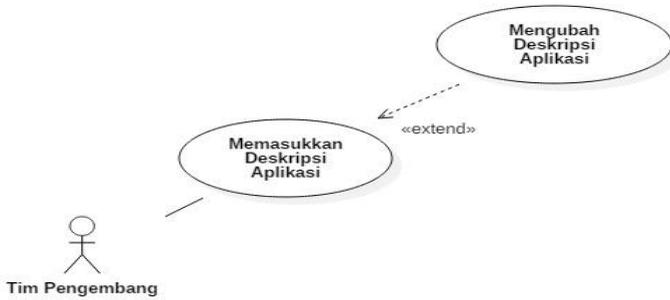
Use case diagram fitur merupakan hal-hal yang bisa dilakukan terutama melalui fitur yang ada di dalam aplikasi estimasi biaya perangkat lunak. Berikut adalah use case berdasarkan fitur yang ada di dalam aplikasi estimasi biaya perangkat lunak.



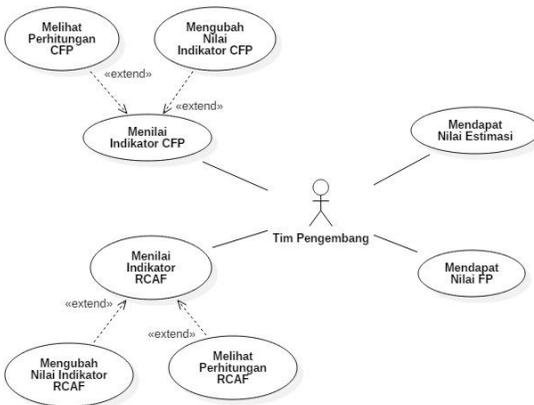
Gambar 4. 6 Desain use case fitur autentifikasi



Gambar 4. 7 Desain use case fitur informasi client



**Gambar 4. 8** Desain use case fitur memasukkan deskripsi aplikasi



**Gambar 4. 9** Desain use case fitur perhitungan function point

#### 4.1.4 Use Case Description

Deskripsi sebuah use case (Use Case Description) berisikan tentang perilaku yang ada pada use case. Use case description akan menjelaskan bagaimana skenario sistem berjalan dari masing-masing *usecase* yang telah dirancang. Di bawah ini merupakan tabel yang berisi deskripsi dari setiap *usecase*.

Tabel 4. 6 Use case description memasukkan informasi client

Nama Usecase	UCD.01.04 Memasukkan Informasi Client		
Aktor	Analisis dan Project Manager		
Deskripsi	Use case ini bermaksud agar aktor dapat memasukkan informasi dari client		
Precondition	Aktor telah melakukan <i>login</i> dan memilih menu <i>create estimation</i>		
Pemicu	Aktor ingin melakukan estimasi harga		
Urutan Kejadian	No	Aktor	Sistem
	1.	Pengguna mengisikan kolom yang ada pada form informasi client	Sistem menampilkan data yang diisikan oleh pengguna
	2.	Pengguna menekan tombol simpan	Sistem menyimpan data informasi client dan menampilkan pesan data

			berhasil disimpan
Langkah Alternatif	-		
Kesimpulan	Use case ini berhasil jika aktor telah mengisi semua kolom pada form pengisian informasi <i>client</i>		
Kondisi Setelah	Aktor dapat memasukkan deskripsi aplikasi		
Aturan Proses	Aktor berada pada halaman <i>create estimation</i> pada tab informasi <i>client</i>		
Asumsi	Aktor ingin melakukan estimasi harga perangkat lunak		

**Tabel 4. 7 Use case description memasukkan deskripsi aplikasi**

Nama Usecase	UCD.01.05 Memasukkan Deskripsi Aplikasi		
Aktor	Analisis dan Project Manager		
Deskripsi	Use case ini dimaksudkan agar aktor bisa memasukkan deskripsi aplikasi		
Precondition	Aktor sudah mengisi informasi <i>client</i>		
Pemicu	Aktor ingin melakukan estimasi harga perangkat lunak		
Urutan Kejadian	No	Aktor	Sistem
	1.	Pengguna mengisikan kolom yang ada pada form deskripsi aplikasi	Sistem menampilkan data yang diisikan oleh pengguna
	2.	Pengguna menekan tombol simpan	Sistem menyimpan data deskripsi

			aplikasi dan menampilkan pesan data berhasil disimpan
Langkah Alternatif	-		
Kesimpulan	Use case berhasil jika aktor telah mengisi semua kolom pada form memasukkan deskripsi aplikasi		
Kondisi Setelah	Aktor dapat memasukkan fitur dari aplikasi		
Aturan Proses	Aktor berada pada halaman <i>create estimation</i> pada form deskripsi aplikasi		
Asumsi	<i>Aktor</i> ingin melakukan estimasi harga perangkat lunak.		

**Tabel 4. 8 Use case description menilai indikator CFP**

Nama Usecase	UCD.01.08 Menilai Indikator CFP		
Aktor	Analis dan Project Manager		
Deskripsi	Use case ini dimaksudkan agar aktor bisa menilai indikator CFP		
Precondition	Aktor sudah mengisi deskripsi aplikasi		
Pemicu	Aktor ingin melakukan estimasi harga perangkat lunak		
Urutan Kejadian	No	Aktor	Sistem
	1.	Pengguna mengisi setiap kolom yang ada pada form CFP	Sistem menampilkan data yang

			diisikan oleh pengguna
	2.	Pengguna menekan tombol simpan	Sistem menyimpan hasil lalu menampilkan hasil perhitungan CFP dan menampilkan pesan data berhasil disimpan
Langkah Alternatif	-		
Kesimpulan	Use case berhasil jika aktor telah mengisi semua kolom pada form memasukkan deskripsi aplikasi		
Kondisi Setelah	Aktor dapat menilai indikator RCAF		
Aturan Proses	Aktor berada pada halaman <i>create estimation</i> pada form CFP		
Asumsi	<i>Aktor</i> ingin melakukan estimasi harga perangkat lunak.		

**Tabel 4. 9 Use case description menilai indikator RCAF**

Nama Usecase	UCD.01.09 Menilai Indikator RCAF
Aktor	Analisis dan Project Manager
Deskripsi	Use case ini dimaksudkan agar aktor bisa menilai indikator RCAF
Precondition	Aktor sudah menilai indikator CFP

Pemicu	Aktor ingin melakukan estimasi harga perangkat lunak		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Pengguna mengisikan setiap kolom yang ada pada form RCAF	Sistem menampilkan data yang diisikan oleh pengguna
	2.	Pengguna menekan tombol simpan	Sistem menyimpan data lalu menampilkan hasil perhitungan RCAF dan menampilkan pesan data berhasil disimpan
Langkah Alternatif	-		
Kesimpulan	Use case berhasil jika aktor telah mengisi semua indikator pada form CFP		
Kondisi Setelah	Aktor mendapat nilai FP		
Aturan Proses	Aktor berada pada halaman <i>create estimation</i> pada form RCAF		
Asumsi	Aktor ingin melakukan estimasi harga perangkat lunak.		

Tabel 4. 10 Use case description mendapat nilai FP

Nama Usecase	UCD.01.10 Mendapat Nilai FP		
Aktor	Analisis dan Project Manager		
Deskripsi	Use case ini dimaksudkan agar aktor bisa mendapat nilai FP		
Precondition	Aktor sudah menilai indikator RCAF		
Pemicu	Aktor ingin melakukan estimasi harga perangkat lunak		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Aktor menekan tombol lihat hasil pada form RCAF	Sistem menampilkan halaman hasil dengan menampilkan tulisan nilai FP
Langkah Alternatif	-		
Kesimpulan	Use case berhasil jika aktor telah mengisi semua kolom pada form CFP dan RCAF		
Kondisi Setelah	Aktor dapat memasukkan biaya operasional		
Aturan Proses	Aktor berada pada halaman <i>create estimation</i> pada form result		
Asumsi	<i>Aktor</i> ingin melakukan estimasi harga perangkat lunak.		

Tabel 4. 11 Use case description mendapat nilai estimasi

Nama Usecase	UCD.01.11 Mendapat Nilai Estimasi
Aktor	Analisis dan Project Manager

Deskripsi	Use case ini dimaksudkan agar aktor bisa mendapat nilai estimasi		
Precondition	Aktor sudah mendapat nilai FP		
Pemicu	Aktor ingin melakukan estimasi harga perangkat lunak		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Aktor menekan tombol lihat hasil pada form RCAF	Sistem menampilkan halaman hasil dengan menampilkan tulisan nilai estimasi
Langkah Alternatif	-		
Kesimpulan	Use case berhasil jika aktor telah mendapat nilai FP		
Kondisi Setelah	Aktor dapat memasukkan <i>effort real</i>		
Aturan Proses	Aktor berada pada halaman <i>create estimation</i> pada form <i>result</i>		
Asumsi	Aktor ingin melakukan estimasi harga perangkat lunak.		

## 4.2 Tahap Desain

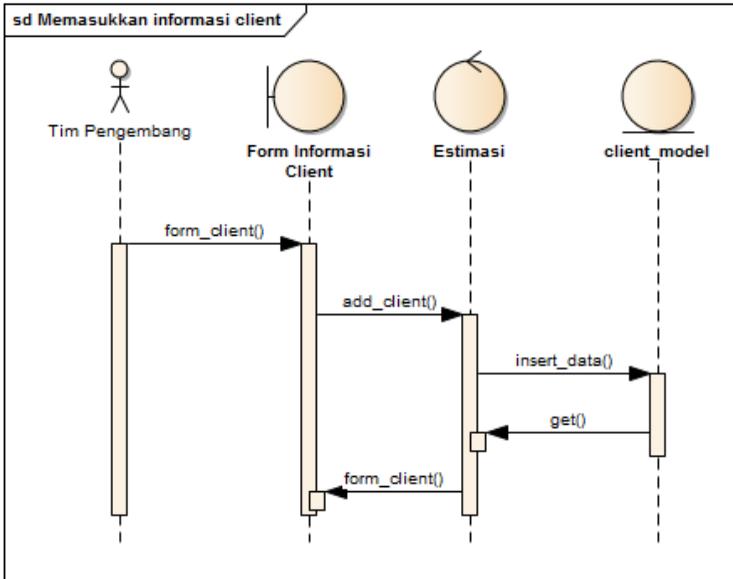
Tahap desain merupakan tahapan yang dilakukan setelah tahap analisis. Tahap desain digunakan untuk mengubah kebutuhan yang sudah dijabarkan pada tahap analisis menjadi representasi ke dalam bentuk “*blueprint*” sistem yakni *sequence diagram*, *class diagram*, *activity diagram* dan desain antarmuka pengguna sebelum tahap pembuatan kode program dimulai.

Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya.

#### 4.2.1 Desain Sequence Diagram

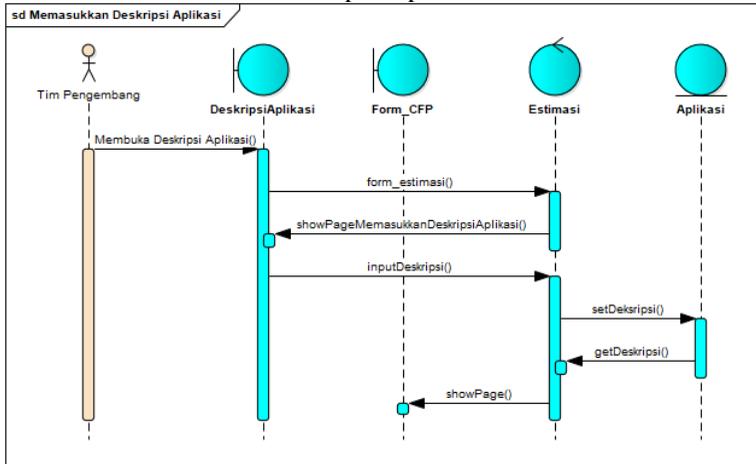
Desain sequence diagram dibuat berdasarkan dari desain *use case* dan deskripsi dari masing-masing *use case* yang ada pada aplikasi. Gambar 4.10 sampai 4.15 ini adalah beberapa contoh *sequence diagram* yang ada pada aplikasi estimasi biaya perangkat lunak.

##### 1. Memasukkan Informasi Client



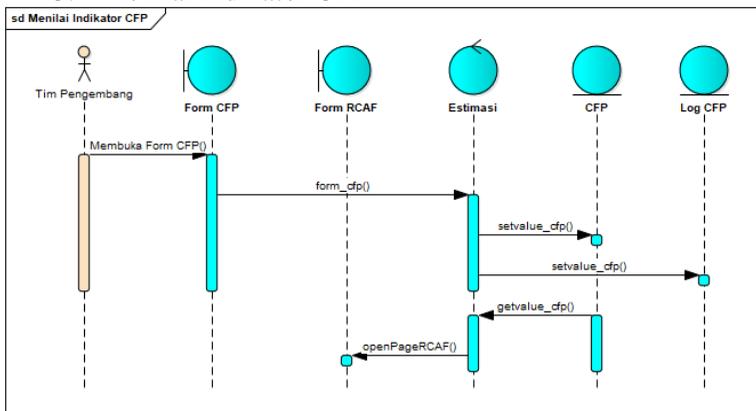
Gambar 4. 10 Sequence diagram fitur memasukkan informasi client

## 2. Memasukkan Deskripsi Aplikasi



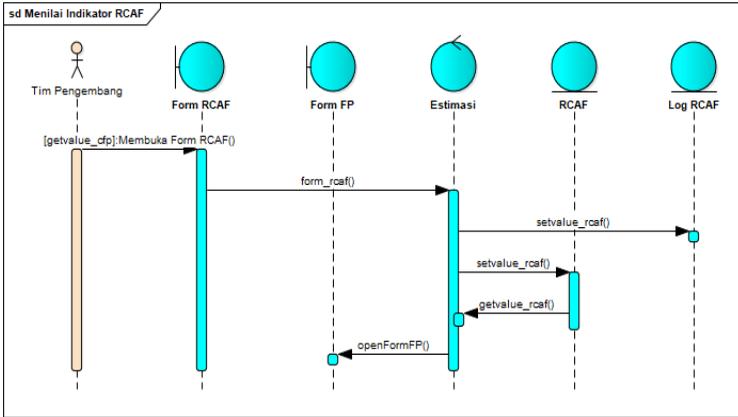
Gambar 4. 11 Sequence diagram fitur memasukkan nama fitur

## 3. Menilai Indikator CFP



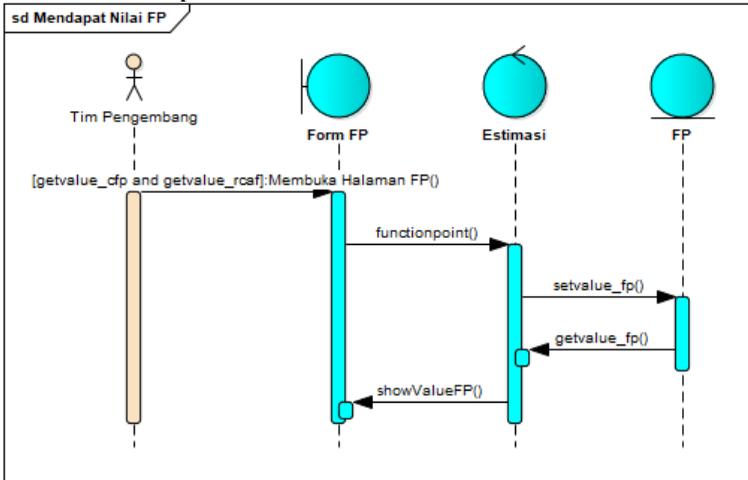
Gambar 4. 12 Sequence diagram fitur menilai indikator CFP

#### 4. Menilai Indikator RCAF



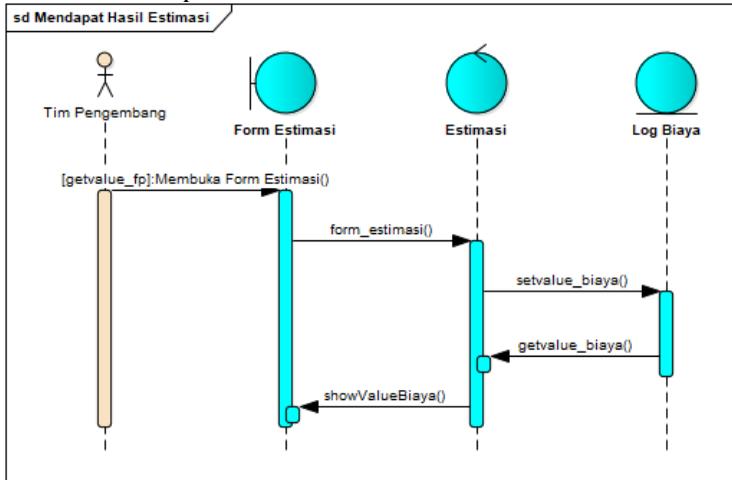
**Gambar 4. 13** Sequence diagram fitur menilai indikator RCAF

#### 5. Mendapat Nilai FP



**Gambar 4. 14** Sequence diagram fitur mendapat nilai FP

## 6. Mendapat Nilai Estimasi

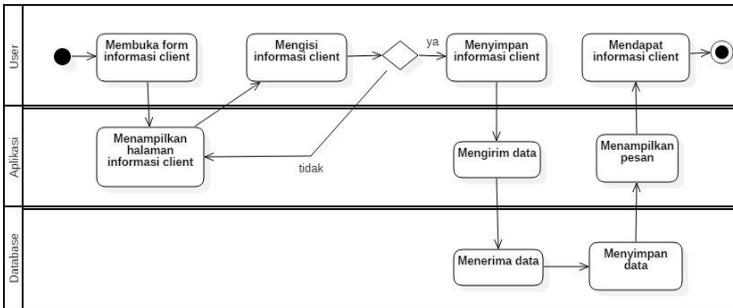


Gambar 4. 15 Sequence diagram fitur mendapat nilai estimasi

### 4.2.2 Desain Activity Diagram

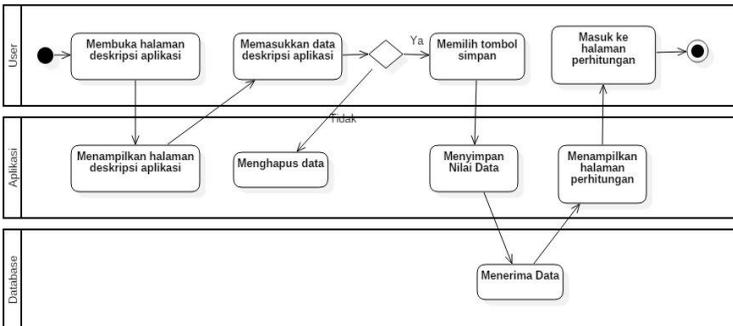
Desain *activity diagram* dibuat berdasarkan dari desain *use case* dan deskripsi masing-masing *use case* yang ada pada aplikasi estimasi biaya perangkat lunak. Gambar 4.16 sampai 4.21 adalah beberapa contoh *activity diagram* yang ada pada aplikasi estimasi biaya perangkat lunak.

### 1. Memasukkan Informasi Client



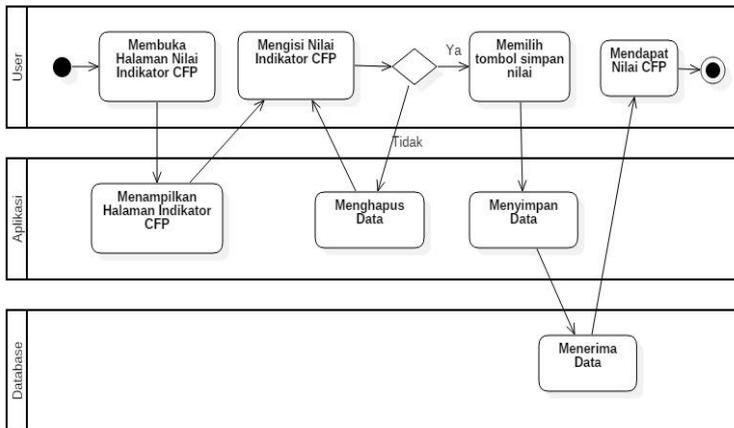
Gambar 4. 16 Activity diagram memasukkan informasi client

### 2. Memasukkan Deskripsi Aplikasi



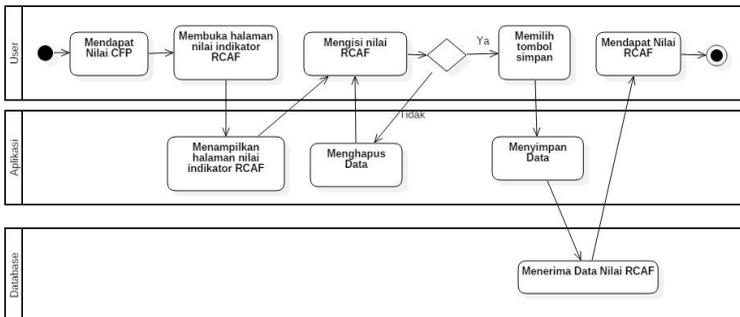
Gambar 4. 17 Activity diagram memasukkan deskripsi aplikasi

### 3. Menilai Indikator CFP



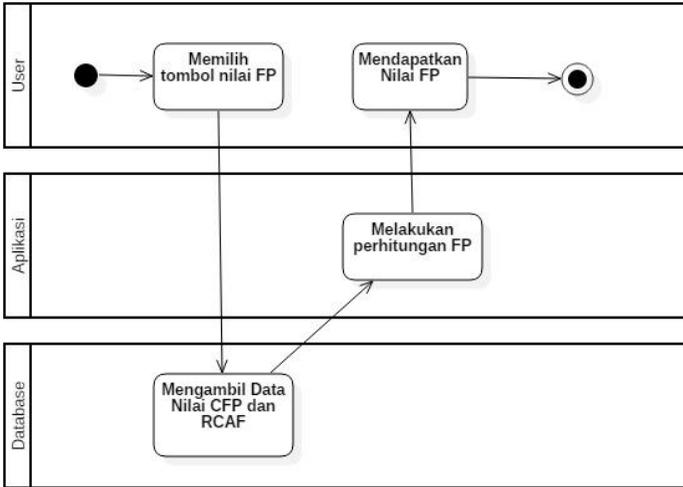
Gambar 4. 18 Activity diagram menilai indikator CFP

### 4. Menilai Indikator RCAF



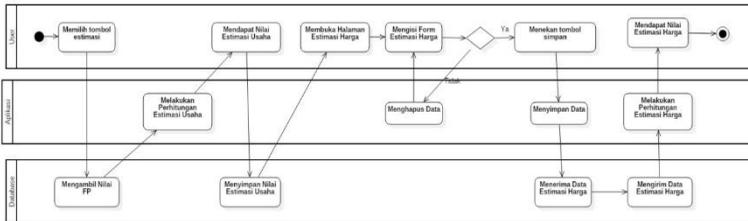
Gambar 4. 19 Activity diagram menilai indikator RCAF

### 5. Mendapat Nilai FP



Gambar 4. 20 Activity diagram mendapat nilai FP

### 6. Mendapat Nilai Estimasi



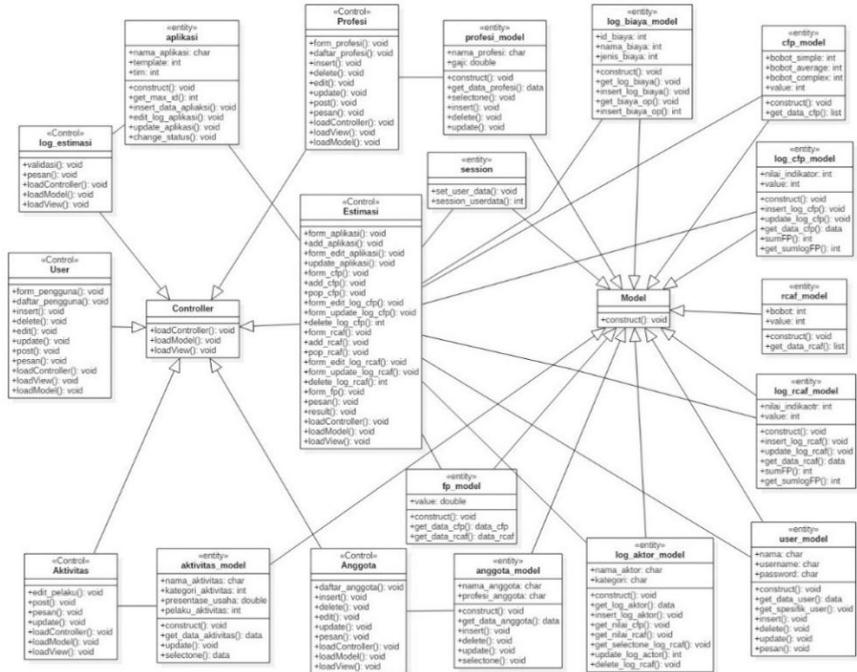
Gambar 4. 21 Activity diagram mendapat nilai estimasi

### 4.2.3 Desain Class Diagram

Class diagram digunakan untuk menampilkan kelas-kelas di dalam sistem dan hubungan relasi yang memberikan gambaran sistem secara statis. Tabel 4.12 merupakan tabel yang akan dibuat untuk class diagram.

**Tabel 4. 12 Daftar tabel data aplikasi function point**

No	Nama tabel
1.	Aktivitas
2.	Anggota
3.	Anggota_tim
4.	Aplikasi
5.	Biaya_op
6.	Client
7.	Fitur
8.	Log_biaya
9.	Log_indikator_cfp
10.	Log_indikator_rcaf
11.	Log_konstanta_effort
12.	Pembobotan_cfp
13.	Pembobotan_rcaf
14.	Penjelasan_cfp
15.	Penjelasan_rcaf
16.	Profesi
17.	Standar_gaji
18.	Tim
19.	User



Gambar 4. 22 Class Diagram aplikasi estimasi biaya perangkat lunak

#### 4.2.4 Desain Antarmuka Pengguna

Desain antarmuka atau yang lebih sering dikenal dengan istilah desain *interface*. Desain antarmuka merupakan desain untuk komputer, perangkat komunikasi *mobile*, sistem perangkat lunak dan situs web yang berfokus pada pengalaman dan interaksi penggunanya. Tujuan dari penggunaan desain antarmuka ini adalah untuk membuat interaksi pengguna sesederhana dan seefisien mungkin. Bagaimana pengguna berinteraksi dengan komputer menggunakan tampilan antarmuka (*interface*) yang ada pada layer komputer atau *mobile*. Desain antarmuka ini sangat penting karena akan sangat berpengaruh pada pengguna dalam menggunakan atau berkomunikasi dengan komputer. Apabila suatu sistem sulit untuk digunakan maka hal ini akan memaksa pengguna untuk melakukan suatu kesalahan saat menggunakan sistem tersebut.

Berikut adalah penjelasan dari desain antarmuka dari aplikasi yang dibangun :

1. Fitur Login

Fitur *login* dibuat sebelum pengguna masuk ke dalam aplikasi. Gambar 4.23 menjelaskan pengguna yang ingin melakukan *login* harus mengisikan *username* dan *password* yang telah dibuat.

## Aplikasi Function Point

Please login with your Username and Password.

analis

.....

Login

**Gambar 4. 23 GUI Login**

2. Halaman Pengguna  
Bagian depan merupakan halaman khusus untuk pengguna. Tabel 4.13 menjelaskan daftar halaman yang ada pada pengguna.

**Tabel 4. 13 Daftar halaman pengguna**

Halaman Pengguna	
No	Nama Halaman
1.	Halaman beranda
2.	Halaman estimasi
3.	Halaman pengguna
4.	Halaman profesi
5.	Halaman anggota tim
6.	Halaman aktivitas

- **Halaman beranda**  
Halaman *home* atau yang biasa dikenal dengan istilah halaman utama atau beranda merupakan halaman awal yang akan muncul ketika kita membuka sistem. Pada halaman utama ini terdapat tampilan menu yang dapat dipilih oleh pengguna serta penjelasan mengenai aplikasi serta ucapan selamat datang kepada pengguna yang telah *login*.



**Gambar 4. 24 GUI Halaman Beranda**

- **Halaman estimasi**  
Halaman estimasi adalah halaman untuk menampilkan proses dari estimasi harga perangkat lunak yang akan dilakukan. Halaman ini berbentuk tab yang berisi dari informasi *client*, deskripsi aplikasi, CFP, RCAF dan *result*. Pengguna harus mengisi setiap kolom yang ada pada tab tersebut sehingga tab tersebut merupakan langkah-langkah yang harus dilakukan pengguna dalam melakukan estimasi. Langkah awal dalam melakukan estimasi harga pemilih harus memasukkan informasi dari *client* yang akan dibuatkan aplikasi.

**Form Estimasi Harga Perangkat Lunak**

Informasi Client | Deskripsi Aplikasi | CFP | RCAF | Result

### Informasi Client

**Nama Client\***

Masukan nama client

Contoh: Bapak Fandhi

**Alamat Client\***

Masukan alamat client

Contoh: Mulyosari Utara 7 No 29

**Tanggal Pengajuan Aplikasi\***

Masukan tanggal pengajuan

Contoh: 01-07-2018

NB: \* Wajib Diisi

**Simpan**

**Gambar 4. 25 GUI Halaman informasi client**

Setelah pengguna mengisikan informasi *client* yang terdiri dari nama *client*, alamat *client* dan tanggal pengajuan aplikasi. Pengguna menekan tombol simpan sehingga akan masuk ke tab deskripsi aplikasi seperti gambar gambar 4.26.

**Form Estimasi Harga Perangkat Lunak**

Informasi Client | Deskripsi Aplikasi **edit** | CFP | RCAF | Result

### Deskripsi Aplikasi

**Nama Aplikasi\***

Aplikasi FP

Contoh: Aplikasi FP

**Penggunaan Metode\***

Function Point  Function Point Modified

Adanya penambahan karakteristik software berupa kompleksitas proses bisnis

**Effort Rate\***

6.2  20  24  36

1) Nilai Effort 6.2 digunakan oleh Subriadi,et al pada penelitiannya untuk proyek kecil-menengah

2) Nilai Effort 20 digunakan oleh Roy K.Clemons,2006 pada penelitiannya untuk proyek kecil

3) Nilai Effort 24 digunakan oleh G.Schneider dan J.Winters pada penelitiannya untuk proyek sedang

4) Nilai Effort 36 digunakan oleh G.Schneider dan J.Winters pada penelitiannya untuk proyek besar

### Fitur Aplikasi

**Nama Fitur**

Masukan nama fitur **+ Tambah**

Contoh: Fitur Pengelola pengguna.

**Daftar Fitur**

Show **5** entries Search:

No	Nama Fitur	Actions
Tidak ada Data		

Showing 0 to 0 of 0 entries PreviousNext

**Berkutnya**

**Gambar 4. 26 GUI Halaman deskripsi aplikasi**

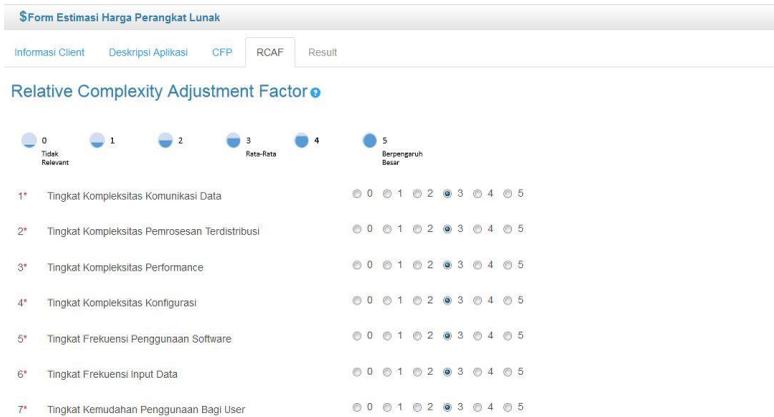
Pengguna harus mengisikan semua kolom yang ada pada deskripsi aplikasi seperti nama aplikasi, metode *function*

*point* yang digunakan, *effort rate* yang digunakan serta anggota tim dalam proyek tersebut. Pengguna menekan tombol simpan sehingga memunculkan tampilan dimana pengguna dapat mengisikan fitur yang ada pada aplikasi yang akan diestimasi tetapi bersifat opsional sehingga pengguna harus menekan tombol berikutnya untuk masuk pada halaman CFP seperti pada gambar 4.27.

Form Estimasi Harga Perangkat Lunak						
Informasi Client		Deskripsi Aplikasi		CFP	RCAF	Result
<b>Crude Function Point</b>						
1*	Number Of User Inputs	Simple	<input type="text"/>	<input type="button" value="OK"/>		
2*	Number Of User Inputs	Average	<input type="text"/>	<input type="button" value="OK"/>		
3*	Number Of User Inputs	Complex	<input type="text"/>	<input type="button" value="OK"/>		
4*	Number Of User Outputs	Simple	<input type="text"/>	<input type="button" value="OK"/>		
5*	Number Of User Outputs	Average	<input type="text"/>	<input type="button" value="OK"/>		
6*	Number Of User Outputs	Complex	<input type="text"/>	<input type="button" value="OK"/>		

**Gambar 4. 27 GUI Halaman CFP**

Pengguna harus mengisikan semua kolom yang ada pada halaman CFP dimana kolom terdiri dari 15 kolom yang bertipe *number field* sehingga hanya bisa diisi angka oleh pengguna untuk melakukan perhitungan *crude function point*. Apabila semua kolom sudah terisi semua pengguna menekan tombol simpan untuk mendapatkan nilai CFP. Lalu akan muncul tombol lihat hasil pada bagian pojok kanan bawah dan pengguna harus menekan tombol lihat hasil untuk masuk ke dalam halaman RCAF seperti pada gambar 4.28.



**Gambar 4. 28 GUI Halaman RCAF**

Setelah menekan tombol lihat hasil pengguna akan berada pada halaman hasil untuk mengetahui hasil nilai *function point* yang didapatkan serta harga perangkat lunak yang didapatkan berdasarkan distribusi usaha seperti pada gambar 4.29 dan gambar 4.30.



**Gambar 4. 29 GUI Halaman hasil function point**

## Rekap Biaya Tiap Aktivitas

[Edit](#)

Aktivitas	Effort		Biaya		
	%	Hour of Effort	Standard gaji/bulan(Rp)	Gaji per Jam (Rp)	Biaya (Rp)
<b>SOFTWARE DEVELOPMENT</b>					
Requirements	7.5 %	63.17	7.000.000,00	43.750,00	2.763.687,50
Specifications & Design	17.5 %	147,40	7.000.000,00	43.750,00	6.448.750,00
Coding	10 %	84,23	5.000.000,00	31.250,00	2.632.187,50
Testing	7 %	58,96	8.000.000,00	50.000,00	2.948.000,00
Sub Total	42,00%	353,76			14.792.625,00
<b>ON GOING ACTIVITY</b>					
Project management	7 %	58,96	20.000.000,00	125.000,00	7.370.000,00

## Biaya Operasional

No	Deskripsi	Biaya (Rp)	Actions
Tidak ada data			
<b>TOTAL</b>		0,00	
<a href="#">+ Tambah</a>			

## Biaya Keseluruhan

No	Deskripsi Biaya	Jumlah (Rp)
1	Biaya Pokok Pembuatan Perangkat Lunak	53.648.500,00
2	Total Biaya Operasional	0,00
<b>TOTAL KESELURUHAN</b>		<b>53.648.500,00</b>
<a href="#">Kirim estimasi</a>		

Gambar 4. 30 GUI Halaman hasil harga aplikasi

Setelah melakukan estimasi pengguna dapat melihat daftar estimasi yang telah dilakukan pada halaman *log* estimasi. Halaman ini dibuat berupa tabel sehingga pengguna dapat melihat secara detail mengenai aplikasi yang diestimasi seperti pada gambar 4.31.

**Daftar Log Estimasi Aplikasi**

10 records per page Search:

No	Tanggal estimasi	Nama Client	Nama Aplikasi	Biaya Estimasi (Total)	Tim Pengembang	Status	Actions
1	31-05-2018	Fandhi Akhmad	Aplikasi FP	Rp. 34.863.500,00	Tim Aplikasi FP	Belum lengkap	<a href="#">Edit</a>
2	28-05-2018	awdwa	Event Extract	Rp. 30.109.750,00	Tim Event Extract	Belum lengkap	<a href="#">Edit</a>
3	28-05-2018	adwa	Event Extract	Rp. 156.846.437,50	Tim Event Extract	Belum lengkap	<a href="#">Edit</a>
4	28-05-2018	Umar Abdul Aziz	Aplikasi FP	Rp. 21.741.500,00	Tim Aplikasi FP	Belum lengkap	<a href="#">Edit</a>
5	28-05-2018	Oman	Event Extract	Rp. 21.741.500,00	Tim Event Extract	Belum lengkap	<a href="#">Edit</a>

Gambar 4. 31 GUI Halaman *log* estimasi

- Halaman pengguna  
Halaman pengguna dibuat khusus untuk pengguna *project manager* dimana terdiri dari dua tampilan yaitu tambah pengguna dan daftar pengguna. Tambah pengguna digunakan untuk menambahkan pengguna pada aplikasi dengan mengisi semua kolom yang ada dan menekan

tombol simpan untuk menambah pengguna seperti pada gambar 4.32.

**Form Pengguna** **Tambah Data**

**Nama\***  
Masukkan Nama

**Username\***  
Masukkan Username

**Email\***  
analis

**New Password\***  
\*\*\*\*

**Peran\***  
 Project Manager  Sistem Analis  Sekretaris

NB: \* Wajib Diisi

[← Kembali](#) [Simpan](#)

**Gambar 4. 32 GUI Halaman tambah pengguna**

Daftar pengguna digunakan ketika pengguna ingin melihat siapa saja pengguna aplikasi tersebut. Tampilan sengaja dibentuk dalam bentuk tabel sehingga pengguna dapat melihat secara detail seperti pada gambar 4.33.

**Daftar Pengguna**

Well done! ×

10 records per page Search:

No	Nama	Username	Email	Role	Actions
1	Administrator	admin		Admin	No Action
2	Project Manager	projectmanager	fandiakhmad94@yahoo.co.id	Project Manager	<a href="#">Edit</a> <a href="#">Delete</a>
3	System Analyst	analis		Sistem Analis	<a href="#">Edit</a> <a href="#">Delete</a>
4	Secretary	sekretaris	demo	Sekretaris	<a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 to 4 of 4 entries

[← Previous](#) [1](#) [Next →](#)

**Gambar 4. 33 GUI Halaman daftar pengguna**

- **Halaman profesi**  
Halaman profesi dibuat khusus pengguna *project manager* dimana terdiri dari dua tampilan yaitu tambah profesi dan daftar profesi. Tambah profesi digunakan untuk menambahkan profesi yang ada pada proyek pengembangan perangkat lunak tersebut dimana pengguna harus mengisi semua kolom dan menekan tombol simpan untuk menambah profesi seperti pada gambar 4.34.



**Form Profesi** [Tambah Data](#)

**Nama profesi\***  
Masukkan nama profesi

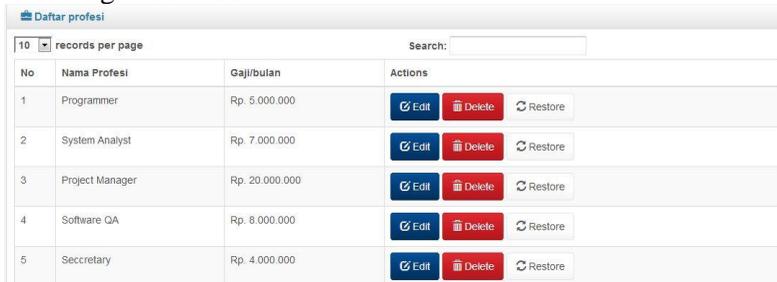
**Gaji /bulan\***  
Masukkan gaji

NB: \* Wajib Diisi

[← Kembali](#) [Simpan](#)

**Gambar 4. 34** GUI Halaman tambah profesi

Daftar profesi digunakan ketika pengguna ingin mengetahui profesi apa saja yang ada dalam proyek tersebut. Tampilan sengaja dibentuk dalam bentuk tabel sehingga pengguna dapat melihat secara detail pada gambar 4.35.



**Daftar profesi**

10 records per page Search:

No	Nama Profesi	Gaji/bulan	Actions
1	Programmer	Rp. 5.000.000	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Restore</a>
2	System Analyst	Rp. 7.000.000	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Restore</a>
3	Project Manager	Rp. 20.000.000	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Restore</a>
4	Software QA	Rp. 8.000.000	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Restore</a>
5	Secretary	Rp. 4.000.000	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Restore</a>

**Gambar 4. 35** GUI Halaman daftar profesi

- Halaman anggota tim  
Halaman anggota tim dibuat khusus pengguna *project manager* dimana terdiri dari dua tampilan yaitu tambah anggota tim dan daftar anggota tim. Tambah anggota tim digunakan untuk menambahkan anggota tim dalam proyek dimana pengguna harus mengisi semua kolom yang ada dan menekan tombol simpan untuk menambah anggota tim seperti pada gambar 4.36.

**Form Anggota** **Tambah Data**

**Nama\***  
Masukkan nama anggota

**Profesi\***  
Pilih Profesi Anggota

**Pengalaman\***  
Masukkan pengalaman anggota

NB: \* Wajib Diisi

[← Kembali](#) [Simpan](#)

**Gambar 4. 36 GUI Halaman tambah anggota tim**

Daftar anggota tim digunakan ketika pengguna ingin mengetahui daftar anggota tim yang ada dalam proyek tersebut. Tampilan sengaja dibentuk dalam bentuk tabel sehingga pengguna dapat melihat secara detail pada gambar 4.37.

**Daftar anggota tim**

10 records per page Search:

No	Nama Anggota	Profesi	Pengalaman	Actions
1	Dewangga Prasetya Praja	Programmer	2 tahun	<a href="#">Edit</a>
2	Naufai Raihan Noly	System Analyst	2 tahun	<a href="#">Edit</a>
3	Fandhi Akhmad	Project Manager	2 tahun	<a href="#">Edit</a>
4	Brilianto WS	Software QA	2 tahun	<a href="#">Edit</a>
5	M Iqbal Imaduddin	Secretary	2 Tahun	<a href="#">Edit</a>

Showing 1 to 5 of 5 entries

[← Previous](#) 1 [Next →](#)

**Gambar 4. 37 GUI Halaman daftar anggota tim**

- Halaman aktivitas  
Halaman aktivitas dibuat khusus pengguna *project manager*. Halaman aktivitas digunakan untuk melihat daftar aktivitas dari proyek pengembangan aplikasi dimana tampilan dibuat dalam bentuk tabel sehingga pengguna dapat melihat secara detail seperti pada gambar 4.38.

No	Nama Aktivitas	Pelaku Aktivitas	Presentase Usaha	Actions
1	Acceptance & Deployment	Project Manager	5%	<a href="#">Ubah Pelaku Aktivitas</a>
2	Acceptance & Deployment	Project Manager	5%	<a href="#">Ubah Pelaku Aktivitas</a>
3	Coding	Programmer	10%	<a href="#">Ubah Pelaku Aktivitas</a>
4	Coding	Programmer	10%	<a href="#">Ubah Pelaku Aktivitas</a>
5	Configuration Management	System Analyst	3%	<a href="#">Ubah Pelaku Aktivitas</a>
6	Configuration Management	System Analyst	3%	<a href="#">Ubah Pelaku Aktivitas</a>
7	Documentation	Secretary	3%	<a href="#">Ubah Pelaku Aktivitas</a>
8	Documentation	Secretary	3%	<a href="#">Ubah Pelaku Aktivitas</a>

**Gambar 4. 38 GUI Halaman daftar aktivitas**

### 4.3 Tahapan Testing

Pada proses rancang bangun suatu aplikasi khususnya menggunakan metode *waterfall* perlu dilakukan tahapan *testing* untuk memastikan aplikasi yang dibangun sudah sesuai dengan ekspetasi yang diharapkan. Tahapan *testing* ini juga merupakan sebagai bagian dari penjamin kualitas aplikasi tersebut. Oleh sebab itu perlu dilakukan perancangan pengujian agar pengujian dapat terlaksana dengan baik dan sistematis. Berikut ini adalah beberapa contoh bentuk rencana pengujian untuk kebutuhan fungsional aplikasi estimasi biaya perangkat lunak.

### **4.3.1 Rencana Pengujian Aplikasi**

Adapun rencana pengujian untuk kebutuhan fungsional aplikasi estimasi biaya perangkat lunak menggunakan metode *function point* seperti pada tabel 4.14 menjelaskan rencana pengujian aplikasi.

**Tabel 4. 14 Rencana pengujian aplikasi estimasi biaya perangkat lunak**

No.	Kode KF	Nama Use Case	Bentuk Pengujian	Identifikasi Pengujian
1.	KFA01	Login	Unit Testing	UT01
		Logout	Unit Testing	UT01
2.	KFA03	Membuka halaman awal	Unit Testing	UT02
3.	KFA05	Memasukkan deskripsi aplikasi	Unit Testing	UT03
4.	KFA08	Menilai indikator CFP	Unit Testing	UT04
5.	KFA09	Menilai indikator RCAF	Unit Testing	UT05
6.	KFA11	Melihat Hasil Estimasi	Unit Testing	UT06
	KFA12			

### 4.3.2 Prosedur Pengujian

Adapun prosedur pengujian aplikasi estimasi biaya perangkat lunak menggunakan metode *function point* adalah sebagai berikut :

Tabel 4. 15 Prosedur pengujian aplikasi estimasi biaya perangkat lunak

Identifikasi Pengujian	Skenario	No	Test Case	Input	Output Ekspetasi
UT01	Melakukan Login	001	Membuka halaman login aplikasi /FunctionPoint/login	Tidak ada	Berhasil ke halaman login
		002	Login dengan user project manager dengan password yang sesuai	Username : projectmanager Password : 12345	Masuk ke halaman homepage
		003	Login dengan user project manager menggunakan password yang tidak sesuai	Username : projectmanager password : 123	Kembali ke halaman login
		004	Login dengan user projectmanager dengan password yang dikosongkan	Username : projectmanager	Kembali ke halaman login
	Melakukan Logout	001	Melakukan klik tombol logout	Tidak ada	Berhasil ke halaman login

Identifikasi Pengujian	Skenario	No	Test Case	Input	Output Ekspetasi
		002	Melakukan logout tanpa login	Tidak ada	Kembali ke halaman login
UT02	Membuka halaman beranda aplikasi	001	Membuka halaman awal aplikasi dengan melakukan login	Tidak ada	Berhasil masuk ke halaman homepage
		002	Membuka halaman awal aplikasi tanpa melakukan login	Tidak ada	Kembali ke halaman login aplikasi
UT03	Masuk ke halaman deskripsi aplikasi	001	Masuk ke halaman deskripsi aplikasi tanpa melakukan login	Tidak ada	Kembali ke halaman login aplikasi
		002	Masuk ke halaman deskripsi aplikasi dengan melakukan login	Username = analis password = 1234	Berhasil masuk ke halaman deskripsi aplikasi
	Memasukkan data deskripsi aplikasi ke	001	Memasukkan data deskripsi aplikasi	nama_aplikasi = Aplikasi Tes Data templete = 0 effort_rate = 8.2	Menampilkan pesan “Data berhasil disimpan” dan

Identifikasi Pengujian	Skenario	No	Test Case	Input	Output Ekspetasi
	dalam database			tim pengembang = 2 id_aplikasi = 2	masuk ke halaman edit aplikasi
	Memeriksa data deskripsi aplikasi di dalam database	001	Memeriksa data deskripsi aplikasi	\$id_aplikasi = 2	Menampilkan data deskripsi aplikasi dari database
	Masuk ke halaman edit deskripsi aplikasi	001	Masuk ke halaman edit deskripsi aplikasi tanpa melakukan login	\$id_aplikasi = 2	Kembali ke halaman login aplikasi
		002	Masuk ke halaman edit deskripsi aplikasi dengan melakukan login	Username = analis Password = 1234	Berhasil masuk ke halaman edit deskripsi aplikasi
	Melakukan perubahan terhadap data	001	Mengubah data deskripsi aplikasi	nama_aplikasi = Aplikasi Tes Data templete = 1 effort_rate = 20	Menampilkan pesan “Data berhasil disimpan” dan masuk ke halaman

Identifikasi Pengujian	Skenario	No	Test Case	Input	Output Ekspetasi
	deskripsi aplikasi			tim pengembang = 2 id_aplikasi = 2	edit deskripsi aplikasi
	Menghapus data deskripsi aplikasi	001	Menghapus data deskripsi aplikasi di dalam database	id_aplikasi = 2	Menampilkan pesan “Data berhasil dihapus” dan kembali ke halaman deskripsi aplikasi
UT04	Masuk ke halaman CFP	001	Masuk ke halaman CFP tanpa melakukan login	Tidak ada	Kembali ke halaman login aplikasi
		002	Masuk ke halaman CFP dengan melakukan login	Username = analis Password = 1234	Berhasil masuk ke dalam aplikasi dan menuju halaman CFP
	Melakukan pengisian	001	Memasukkan nilai CFP	surveycfp1 = 1 surveycfp2 = 1 surveycfp3 = 1	Menampilkan pesan “data berhasil disimpan”

Identifikasi Pengujian	Skenario	No	Test Case	Input	Output Ekspetasi
	indikator CFP			surveycfp4 = 1 surveycfp5 = 1 surveycfp6 = 1 surveycfp7 = 1 surveycfp8 = 1 surveycfp9 = 1 surveycfp10 = 1 surveycfp11 = 1 surveycfp12 = 1 surveycfp13 = 1 surveycfp14 = 1 surveycfp 15 = 1	kemudian masuk ke halaman edit CFP
	Memeriksa data CFP yang sudah masuk di dalam database	001	Memeriksa data CFP di database	\$id_aplikasi = 2	Menampilkan data penilaian CFP

Identifikasi Pengujian	Skenario	No	Test Case	Input	Output Ekspetasi
	Masuk ke halaman edit CFP	001	Masuk ke halaman edit CFP tanpa melakukan login	\$id_aplikasi = 2	Kembali ke halaman login aplikasi
		002	Masuk ke halaman edit CFP dengan melakukan login	Username = analis Password = 1234	Berhasil login dan menuju ke halaman edit CFP
	Melakukan perubahan terhadap data CFP	001	Mengubah data penilaian CFP yang sudah ada	surveycfp1 = 2 surveycfp2 = 2 surveycfp3 = 2 surveycfp4 = 2 surveycfp5 = 2 surveycfp6 = 2 surveycfp7 = 2 surveycfp8 = 2 surveycfp9 = 2 surveycfp10 = 2 surveycfp11 = 2 surveycfp12 = 2	Menampilkan pesan “Data berhasil disimpan” dan kembali ke halaman edit CFP

Identifikasi Pengujian	Skenario	No	Test Case	Input	Output Ekspetasi
				surveycfp13 = 2 surveycfp14 = 2 surveycfp 15 = 2	
	Menghapus data CFP	001	Menghapus data CFP yang ada di database	\$id_aplikasi = 2	Menampilkan pesan “Data berhasil dihapus” dan kembali ke halaman CFP
UT05	Masuk ke halaman RCAF	001	Masuk ke halaman RCAF tanpa melakukan login	Tidak ada	Kembali ke halaman login aplikasi
		002	Masuk ke halaman RCAF dengan melakukan login	Username = analis Password = 1234	Berhasil masuk ke dalam aplikasi dan menuju halaman RCAF
	Melakukan pengisian nilai RCAF	001	Memasukkan data RCAF	surveyrcaf1 = 3 surveyrcaf2 = 3 surveyrcaf3 = 3	Menampilkan pesan “Data berhasil disimpan” dan

Identifikasi Pengujian	Skenario	No	Test Case	Input	Output Ekspetasi
				surveyrcaf4 = 3 surveyrcaf5 = 3 surveyrcaf6 = 3 surveyrcaf7 = 3 surveyrcaf8 = 3 surveyrcaf9 = 3 surveyrcaf10 = 3 surveyrcaf11 = 3 surveyrcaf12 = 3 surveyrcaf13 = 3 surveyrcaf14 = 3 surveyrcaf15 = 3	kembali ke halaman edit RCAF
	Memeriksa nilai RCAF	001	Memeriksa nilai RCAF yang sudah masuk di database	\$id_aplikasi = 2	Menampilkan data RCAF yang ada di database
	Masuk ke halaman edit RCAF	001	Masuk ke halaman edit RCAF tanpa melakukan login	Tidak ada	Kembali ke halaman login aplikasi

Identifikasi Pengujian	Skenario	No	Test Case	Input	Output Ekspetasi
		002	Masuk ke halaman edit RCAF dengan melakukan login	Username = analis Password = 1234	Berhasil login ke aplikasi dan menuju halaman edit RCAF
	Melakukan perubahan nilai RCAF	001	Mengubah data RCAF di database	surveyrcaf1 = 2 surveyrcaf2 = 2 surveyrcaf3 = 2 surveyrcaf4 = 2 surveyrcaf5 = 2 surveyrcaf6 = 2 surveyrcaf7 = 2 surveyrcaf8 = 2 surveyrcaf9 = 2 surveyrcaf10 = 2 surveyrcaf11 = 2 surveyrcaf12 = 2 surveyrcaf13 = 2 surveyrcaf14 = 2 surveyrcaf15 = 2	Menampilkan pesan “Data berhasil disimpan” dan kembali ke halaman edit RCAF

Identifikasi Pengujian	Skenario	No	Test Case	Input	Output Ekspetasi
	Menghapus nilai RCAF	001	Menghapus nilai RCAF yang ada di database	Id_aplikasi = 2	Menampilkan pesan “Data berhasil di hapus”
UT06	Masuk ke halaman result	001	Masuk ke halaman result tanpa melakukan login	Tidak ada	Kembali kehalaman login
		002	Masuk ke halaman result dengan melakukan login	Username = analis Password = 1234	Berhasil masuk ke dalam aplikasi dan menuju ke halaman result

*Halaman ini sengaja dikosongkan*

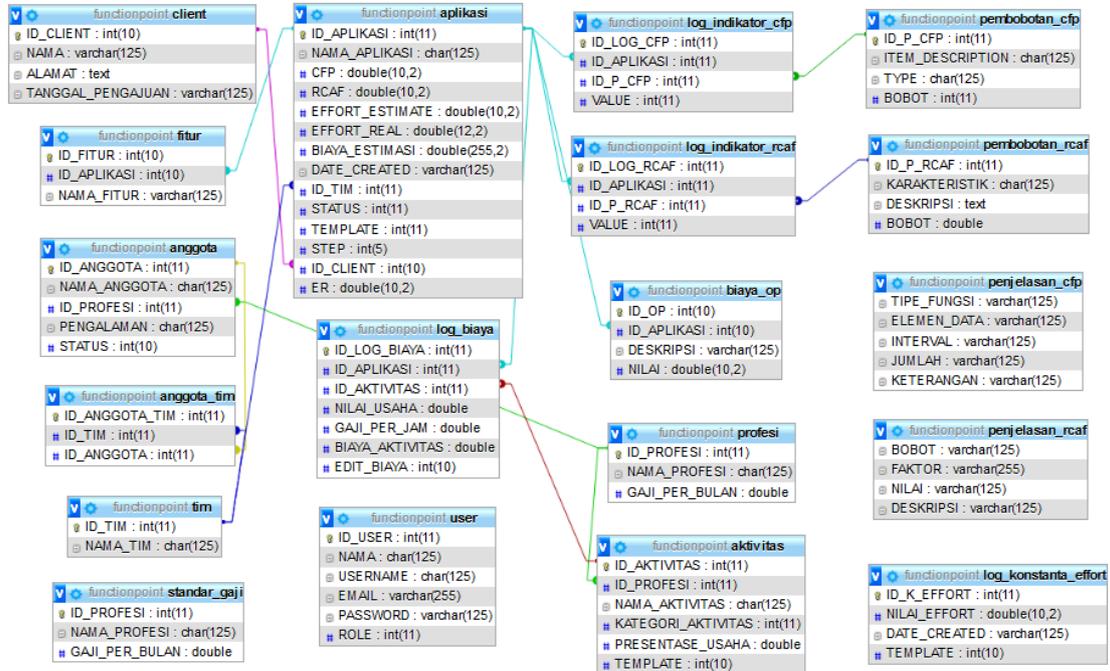
## **BAB V IMPLEMENTASI**

Pada bab ini akan dijelaskan mengenai implementasi pengembangan aplikasi estimasi biaya perangkat lunak berdasarkan hasil dari proses perancangan perangkat penggalan data yang didapatkan melalui observasi dan analisis dokumen.

Pada bab ini akan menjelaskan mengenai uraian proses pembuatan kode program.

### **5.1 Pembuatan Database**

Database yang dibuat berdasarkan desain model data fisik. Desain data model fisik yang sudah dibuat akan langsung di *generate* kedalam bentuk spesifik. Pada pembuatan aplikasi estimasi biaya perangkat lunak ini, peneliti menggunakan database yang dibuat dalam bentuk database MySQL.



Gambar 5. 1 Database aplikasi estimasi biaya perangkat lunak

## 5.2 Pembuatan Kode Program

Proses pembuatan kode program dilakukan setelah database dan *user interface* selesai dibuat. Bahasa pemrograman yang digunakan adalah PHP (*Hypertext Preprocessor*). Pembuatan kode program dilakukan pada tiga komponen utama yakni model, view dan controller ketiga komponen tersebut dikenal dengan sebutan MVC. Berikut adalah contoh kode program dari model, view dan controller halaman untuk menampilkan produk.

### a. Source code controller

Source code controller estimasi berfungsi untuk menjalankan proses estimasi dengan metode *function point* dengan menghubungkan view dan model dari proses estimasi.

```

571 // TODO: CFP RCAF
572 public function form_cfp()
573 {
574     $id_aplikasi = $this->session->userdata('id_aplikasi');
575     $berisi = $this->log_cfp_model->get_data_log($id_aplikasi)->num_rows();
576     if ($berisi !=0) {
577         redirect('estimasi_fp/form_edit_cfp/'. $id_aplikasi);
578     }
579
580     $this->load->model('cfp_model');
581     $isi['isi'] = $this->cfp_model->get_data();
582     $isi['edit'] = false;
583     $isi['step'] = $this->aplikasi_model->edit_log_aplikasi($id_aplikasi)->row()->STEP;
584     $user['nama'] = $this->session->userdata('nama');
585     $user['id_user'] = $this->session->userdata('id_user');
586     $data['header'] = $this->load->view('backend_fp/header', $user, true);
587     $role['role'] = $this->session->userdata('role');
588     $data['menu_kiri'] = $this->load->view('backend_fp/menu_kiri',$role, true);
589     $data['content'] = $this->load->view('frontend_fp/form_cfp', $isi, true);
590     $this->load->view('/backend_fp/main', $data);
591 }
592
593
594 public function add_cfp()
595 {
596     //get session id aplikasi
597     $id_aplikasi = $this->session->userdata('id_aplikasi');
598
599     //Melakukan pembaharuan log jika sudah ada sebelumnya
600     $this->log_cfp_model->delete($id_aplikasi);
601
602     $jumlah = $this->log_cfp_model->count();
603     $index = 1;
604     $status["variable"] = "";
605     $status["errpesan"] = "";
606     $status["size"] = $jumlah;
607     $data[] = "";
608     $pesan[] = "";
609     $valid = true;
610     while ($index <= $jumlah) {

```

Gambar 5. 2 Source code controller estimasi

## b. Source code view

Source code view digunakan untuk menampilkan form *create estimation* dimana informasi yang ada yaitu berupa tab yang berisi informasi *client*, deskripsi aplikasi, CFP, RCAF dan *result*.

```

232 <div class="col-md-6">
233 <label class="control-label" for="selectError">
234 <input type="text" value="Hasil Akhir Perhitungan Nilai FP"/>
235 </label>
236 <table width="400" height="287" border="0">
237 <tr>
238 <th width="227" scope="col">
239 <div align="left">Nilai CFP <br> (<i>Crude Function Point</i>)</div>
240 </th>
241 <th width="18" scope="col">
242 <div align="left"></div>
243 </th>
244 <td width="68" scope="col">
245 <div align="left">?</div>
246 </td>
247 </tr>
248 <tr>
249 <th scope="row">
250 <div align="left">Nilai RCAF <br> (<i>Relative Complexity Adjustment Factor</i>)</div>
251 </th>
252 <td><strong>?</strong></td>
253 <td>?</td>
254 </tr>
255 <tr>
256 <th scope="row">
257 <div align="left">Nilai FP <br> (<i>Crude Function Point</i> * (0.65 + 0.01 * RCAF))</div>
258 </th>
259 <td><strong>?</strong></td>
260 <td>?</td>
261 </tr>
262 </table>
263 </div>
264 </div>
265 </div>
266 </div>

```

**Gambar 5.3 Source code view halaman untuk menampilkan create estimation**

## c. Source code model

Pengambilan data berdasarkan id dari aplikasi yang telah dimasukkan supaya estimasi yang dilakukan dapat ditampilkan berdasarkan aplikasinya. Gambar 5.4 adalah kode program pengambilan data estimasi berdasarkan aplikasi. Pengambilan data diambil berdasarkan id aplikasi pada tabel aplikasi.

```

37
38 //insert log estimasi
39 function insert_data_aplikasi($data)
40 {
41     $this->db->insert('aplikasi', $data);
42 }
43
44 //update value CFP/RCAF transaction log
45 function update_aplikasi($data,$id)
46 {
47     $this->db->where('ID_APLIKASI',$id);
48     $this->db->update('aplikasi', $data);
49 }
50
51 function get_anggota_tim($id)
52 {
53     $query = $this->db->query("SELECT * FROM anggota_tim as at
54 INNER JOIN tim as t ON at.ID_TIM =t.ID_TIM
55 INNER JOIN aplikasi as a ON a.ID_TIM=t.ID_TIM
56 INNER JOIN anggota as ag ON ag.ID_ANGGOTA=at.ID_ANGGOTA
57 INNER JOIN profesi as p ON p.ID_PROFESI=ag.ID_PROFESI
58 WHERE ID_APLIKASI=".$id."");
59     return $query;
60 }
61
62 function get_anggota_tim_by_profesi($id)
63 {
64     $query = $this->db->query("SELECT * FROM anggota_tim as at
65 INNER JOIN tim as t ON at.ID_TIM =t.ID_TIM
66 INNER JOIN aplikasi as a ON a.ID_TIM=t.ID_TIM
67 INNER JOIN anggota as ag ON ag.ID_ANGGOTA=at.ID_ANGGOTA
68 INNER JOIN profesi as p ON p.ID_PROFESI=ag.ID_PROFESI
69 WHERE ID_APLIKASI=".$id."");
70     return $query;
71 }

```

**Gambar 5. 4** Source code model untuk mengambil data estimasi berdasarkan id aplikasi

### 5.3 Pembuatan Kode Pengujian

Proses pembuatan kode pengujian dilakukan setelah melakukan pembuatan kode program aplikasi. Pengujian yang digunakan menggunakan *unit testing*. Berikut adalah contoh kode program dari pengujian aplikasi.

```
// UT03-1
public function test_add_aplikasi_notlogin()
{
    $out = $this->request('GET', '/estimasi_fp/add_aplikasi');
    $this->assertRedirect('login');
}

// UT03-5
public function test_form_edit_aplikasi_notlogin()
{
    $out = $this->request('GET', '/estimasi_fp/form_edit_aplikasi/2');
    $this->assertRedirect('login');
}

// UT03-6
public function test_form_edit_aplikasi()
{
    $username = 'projectmanager';
    $password = '12345';
    $name = 'Project Manager';
    $id = 2;

    $this->request('POST', 'login/auth', [
        'username' => $username,
        'password' => $password,
    ]);
    $_SESSION['id_aplikasi'] = '1';
    $expect = '<h3>Deskripsi Aplikasi</h3>';
    $out = $this->request('GET', '/estimasi_fp/form_edit_aplikasi/2');
    $this->assertContains($expect, $out);
}
```

Gambar 5. 5 Source code pengujian aplikasi

## 5.4 Sampel Implementasi Sistem

Setelah melakukan tahap desain dan pembuatan aplikasi dengan cara membuat kode program untuk model, view dan controller maka peneliti dapat menghasilkan aplikasi estimasi biaya perangkat lunak menggunakan metode *function point*. Berikut adalah beberapa sampel tampilan aplikasi estimasi biaya perangkat lunak.

a. *Homepage*

*Homepage* merupakan tampilan antarmuka halaman awal yang digunakan sebagai tampilan awal pengguna ketika berhasil *login* ke dalam aplikasi estimasi biaya perangkat lunak.



**Gambar 5. 6 Tampilan antarmuka homepage aplikasi estimasi biaya perangkat lunak**

b. *Create Estimation*

*Create Estimation* merupakan tampilan antarmuka halaman *create estimation* yang digunakan oleh pengguna untuk melakukan estimasi suatu aplikasi.

The screenshot displays a web form titled "\$Form Estimasi Harga Perangkat Lunak". At the top, there is a navigation bar with tabs: "Informasi Client" (selected), "Deskripsi Aplikasi", "CFP", "RCAF", and "Result". Below the navigation bar, the "Informasi Client" section is visible. It contains three required input fields, each with a placeholder text and an example below it:

- Nama Client\***: Input field with placeholder "Masukan nama client" and example "Contoh: Bapak Fandhi".
- Alamat Client\***: Input field with placeholder "Masukan alamat client" and example "Contoh: Mulyosari Utara 7 No 29".
- Tanggal Pengajuan Aplikasi\***: Input field with placeholder "Masukan tanggal pengajuan" and example "Contoh: 01-07-2018".

At the bottom of the form, there is a note: "NB: \* Wajib Diisi" and a green "Simpan" button.

**Gambar 5. 7 Tampilan antarmuka create estimation aplikasi estimasi biaya perangkat lunak**

c. Hasil Estimasi

Hasil Estimasi merupakan tampilan antarmuka halaman hasil yang digunakan oleh pengguna untuk melihat hasil dari estimasi biaya suatu perangkat lunak menggunakan metode *function point*.

The screenshot shows a web application interface for software cost estimation. At the top, there is a header bar with the title "\$Form Estimasi Harga Perangkat Lunak". Below the header, there are navigation tabs: "Informasi Client", "Deskripsi Aplikasi", "CFP", "RCAF", and "Result". The "Result" tab is currently selected. The main content area is divided into two columns. The left column is titled "Hasil Akhir Perhitungan Nilai FP" and contains three rows of data: "Nilai CFP (Crude Function Point) : 96.00", "Nilai RCAF (Relative Complexity Adjustment Factor) : 45.00", and "Nilai FP  $CFP * (0.65 + 0.01 * RCAF)$  : 105.6". The right column is titled "Hasil Perhitungan Usaha" and contains two rows of data: "Nilai Effort Rate (Konstanta) : 8.2" and "Nilai Total Usaha (Hour of Effort) : 865.92".

Hasil Akhir Perhitungan Nilai FP		Hasil Perhitungan Usaha	
Nilai CFP (Crude Function Point)	: 96.00	Nilai Effort Rate (Konstanta)	: 8.2
Nilai RCAF (Relative Complexity Adjustment Factor)	: 45.00	Nilai Total Usaha (Hour of Effort)	: 865.92
Nilai FP $CFP * (0.65 + 0.01 * RCAF)$	: 105.6		

Gambar 5. 8 Tampilan antarmuka hasil estimasi aplikasi estimasi biaya perangkat lunak

d. *Log Estimasi*

*Log Estimasi* merupakan tampilan antarmuka halaman *log estimasi* yang digunakan oleh pengguna untuk melihat daftar estimasi yang sudah pernah dilakukan oleh pengguna.

Daftar Log Estimasi Aplikasi							
10 records per page		Search: <input type="text"/>					
No	Tanggal estimasi	Nama Client	Nama Aplikasi	Biaya Estimasi (Total)	Tim Pengembang	Status	Actions
1	31-05-2018	Fandhi Akhmad	Aplikasi FP	Rp. 34.863.500,00	<a href="#">Tim Aplikasi FP</a>	Belum lengkap	<a href="#">Edit</a>
2	28-05-2018	awdwa	Event Extract	Rp. 30.109.750,00	<a href="#">Tim Event Extract</a>	Belum lengkap	<a href="#">Edit</a>
3	28-05-2018	adwa	Event Extract	Rp. 156.846.437,50	<a href="#">Tim Event Extract</a>	Belum lengkap	<a href="#">Edit</a>
4	28-05-2018	Umar Abdul Aziz	Aplikasi FP	Rp. 21.741.500,00	<a href="#">Tim Aplikasi FP</a>	Belum lengkap	<a href="#">Edit</a>
5	28-05-2018	Oman	Event Extract	Rp. 21.741.500,00	<a href="#">Tim Event Extract</a>	Belum lengkap	<a href="#">Edit</a>

**Gambar 5. 9 Tampilan antarmuka log estimasi aplikasi estimasi biaya perangkat luna**

## BAB VI HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan hasil dan pembahasan dari pengembangan aplikasi estimasi biaya perangkat lunak menggunakan metode *function point*.

### 6.1 Aplikasi Estimasi Biaya Perangkat Lunak

Hasil dari rancang bangun aplikasi estimasi biaya perangkat lunak ini yaitu berupa aplikasi estimasi biaya perangkat lunak yang dibuat menggunakan metode *waterfall*. Berikut ini merupakan beberapa contoh gambar bentuk tampilan antarmuka aplikasi estimasi biaya perangkat lunak.

- *Homepage*  
Halaman *homepage* ini merupakan halaman ketika pengguna sudah melakukan *login* serta merupakan halaman awal untuk pengguna. Gambar 6.1 menampilkan halaman *homepage* aplikasi.



Gambar 6. 1 Halaman homepage

- Form Informasi *Client*  
Halaman form informasi *client* adalah tahapan pertama pengguna dalam melakukan pembuatan estimasi dengan tujuan untuk memasukkan informasi *client* dari aplikasi yang akan di estimasi. Gambar 6.2 menampilkan form informasi *client*.

**\$ Form Estimasi Harga Perangkat Lunak**

Informasi Client   Deskripsi Aplikasi   CFP   RCAF   Result

## Informasi Client

**Nama Client\***

Contoh: Bapak Fandhi

**Alamat Client\***

Contoh: Mulyosari Utara 7 No 29

**Tanggal Pengajuan Aplikasi\***

Contoh: 01-07-2018

NB: \* Wajib Diisi

**Gambar 6. 2 Form informasi client**

- **Form Deskripsi Aplikasi**  
 Halaman form deskripsi aplikasi adalah tahapan kedua untuk pengguna dalam melakukan estimasi dengan tujuan untuk memasukkan nama aplikasi dan fitur dari aplikasi yang akan diestimasi. Gambar 6.3 menampilkan form deskripsi aplikasi.

**\$ Form Estimasi Harga Perangkat Lunak**

Informasi Client   Deskripsi Aplikasi **Edit**   CFP   RCAF   Result

## Deskripsi Aplikasi

**Nama Aplikasi\***

Contoh: Aplikasi FP

**Penggunaan Metode\***

Function Point \* Function Point Modified  
 Adanya penambahan karakteristik software berupa kompleksitas proses bisnis

**Anggota Pengembang\***

Anggota pengembang bisa lebih dari 1.

NB: \* Wajib Diisi

## Fitur Aplikasi

**Nama Fitur**

Contoh: Fitur Pengelolah pengguna.

**Daftar Fitur**

Show **5** entries   Search:

No	Nama Fitur	Actions
1	Fitur Pengelolah Pengguna	<input type="button" value="Delete"/>

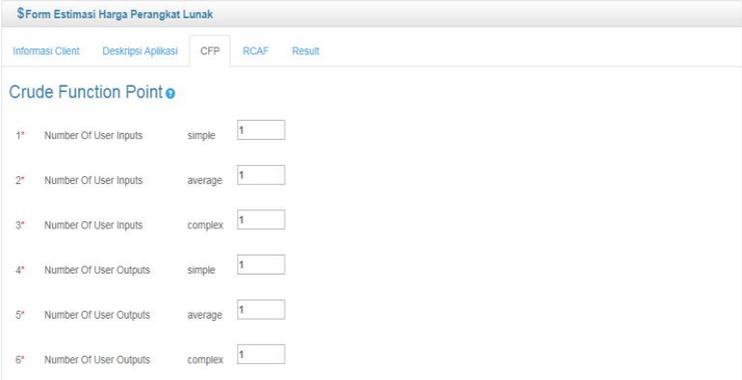
Showing 1 to 1 of 1 entries

[PreviousNext](#)

**Gambar 6. 3 Form deskripsi aplikasi**

- **Form Penilaian CFP**

Halaman form penilaian CFP adalah tahapan ketiga untuk pengguna dalam melakukan estimasi dengan tujuan untuk menghitung nilai CFP dari aplikasi yang akan di estimasi. Gambar 6.4 menampilkan form pengisian nilai CFP.



Form Estimasi Harga Perangkat Lunak						
Informasi Client		Deskripsi Aplikasi		CFP	RCAF	Result
<b>Crude Function Point</b>						
1*	Number Of User Inputs	simple		1		
2*	Number Of User Inputs	average		1		
3*	Number Of User Inputs	complex		1		
4*	Number Of User Outputs	simple		1		
5*	Number Of User Outputs	average		1		
6*	Number Of User Outputs	complex		1		

**Gambar 6. 4 Form penilaian CFP**

- **Form Penilaian RCAF**

Halaman form penilaian RCAF adalah tahapan keempat untuk pengguna dalam melakukan estimasi dengan tujuan untuk menghitung nilai RCAF dari aplikasi yang akan di estimasi. Gambar 6.5 menampilkan form pengisian nilai RCAF.

**Form Estimasi Harga Perangkat Lunak**

Informasi Client   Deskripsi Aplikasi   CFP   RCAF   **Edit**   Result

**Relative Complexity Adjustment Factor**

0 Tidak Relevan   1   2   3 Rata-Rata   4   5 Berpengaruh Besar

1\* Tingkat Kompleksitas Komunikasi Data   0 1 2 3 4 5

2\* Tingkat Kompleksitas Pemrosesan Terdistribusi   0 1 2 3 4 5

3\* Tingkat Kompleksitas Performance   0 1 2 3 4 5

4\* Tingkat Kompleksitas Konfigurasi   0 1 2 3 4 5

5\* Tingkat Frekuensi Penggunaan Software   0 1 2 3 4 5

6\* Tingkat Frekuensi Input Data   0 1 2 3 4 5

7\* Tingkat Kemudahan Penggunaan Bagi User   0 1 2 3 4 5

8\* Tingkat Frekuensi Update Data   0 1 2 3 4 5

**Gambar 6. 5 Form penilaian RCAF**

- **Halaman Hasil Estimasi**  
Halaman hasil estimasi digunakan untuk melihat hasil dari estimasi harga yang telah dilakukan oleh pengguna. Gambar 6.6 menampilkan halaman hasil estimasi.

**Form Estimasi Harga Perangkat Lunak**

Informasi Client   Deskripsi Aplikasi   CFP   RCAF   **Result**

**Hasil Akhir Perhitungan Nilai FP**

Nilai CFP  
(Crude Function Point) : 96.00

Nilai RCAF  
(Relative Complexity  
Adjustment Factor) : 42.00

Nilai FP  
 $CFP * (0.65 + 0.01 * RCAF)$  : 102.72

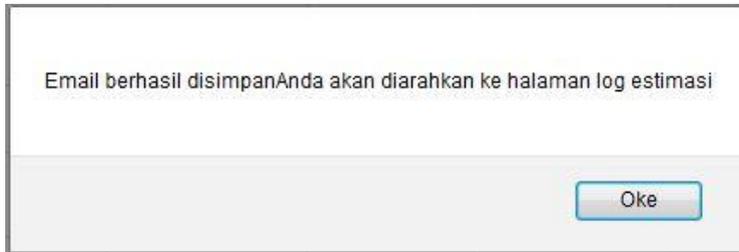
**Hasil Perhitungan Usaha**

Nilai Effort Rate  
(Konstanta) : 8.2

Nilai Total Usaha  
(Hour of Effort) : 842.304

**Gambar 6. 6 Halaman hasil estimasi**

- **Kirim Hasil Estimasi**  
Kirim hasil estimasi digunakan untuk mengirimkan hasil estimasi yang sudah dilakukan kepada *project manager*. Gambar 6.7 menampilkan pop up ketika pengiriman berhasil dilakukan.



**Gambar 6. 7 Pop up berhasil kirim estimasi**

- **Notifikasi *Email***  
Notifikasi *email* didapatkan ketika top up keberhasilan kirim estimasi telah muncul. Gambar 6.8 dan 6.9 memperlihatkan notifikasi yang masuk kepada *email* seorang *project manager*.

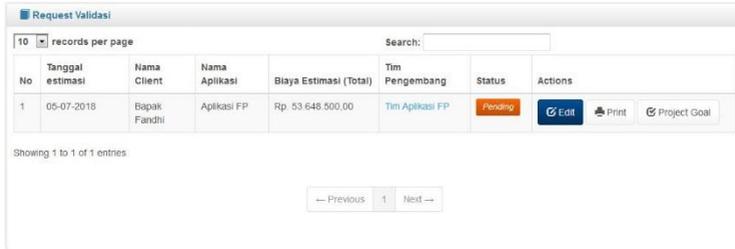


**Gambar 6. 8 Notifikasi email**



**Gambar 6. 9 Isi pesan email**

- ***Request Validasi***  
Halaman *request* validasi digunakan muncul ketika seorang *project manager* menekan *link* yang ada pada *email* yang telah masuk. Gambar 6.10 menampilkan halaman yang muncul ketika seorang *project manager* menekan *link* yang ada pada *email*.



No	Tanggal estimasi	Nama Client	Nama Aplikasi	Biaya Estimasi (Total)	Tim Pengembang	Status	Actions
1	05-07-2018	Bapak Fandhi	Aplikasi FP	Rp. 53.648.500,00	Tim Aplikasi FP	Pending	<a href="#">Edit</a> <a href="#">Print</a> <a href="#">Project Goal</a>

Showing 1 to 1 of 1 entries.

[← Previous](#)
[1](#)
[Next →](#)

**Gambar 6. 10 Halaman request validasi**

- **Validasi Hasil Estimasi**  
Validasi dilakukan ketika seorang *project manager* menekan tombol *edit* pada halaman *request validasi*. Gambar 6.11 menampilkan halaman jika *project manager* ingin melakukan validasi terhadap hasil estimasi.

## Biaya Keseluruhan

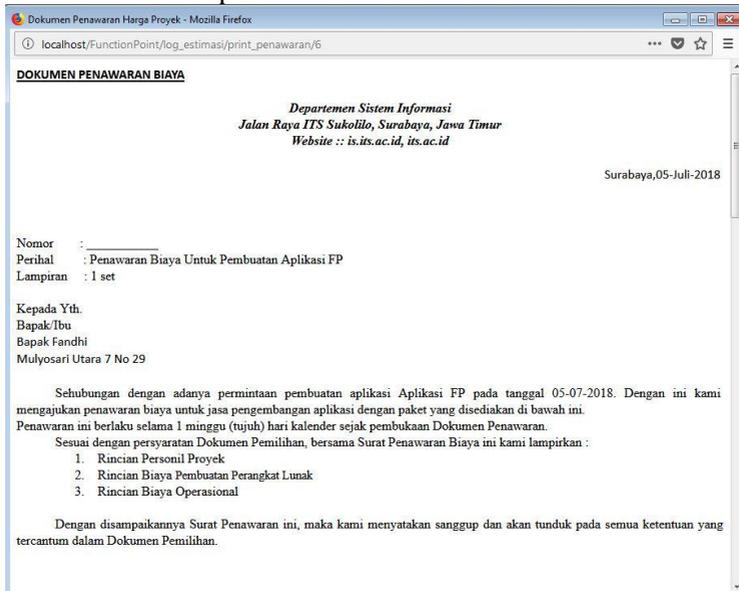
No	Deskripsi Biaya	Jumlah (Rp)
1	Biaya Pokok Pembuatan Perangkat Lunak	53.648.500,00
2	Total Biaya Operasional	0,00
<b>TOTAL KESELURUHAN</b>		<b>53.648.500,00</b>



**Gambar 6. 11 Halaman validasi hasil estimasi**

- **Cetak Dokumen Penawaran**  
Cetak dokumen penawaran digunakan agar estimasi yang sudah dilakukan dapat dibuat dalam bentuk dokumen sehingga dapat segera di cetak dalam bentuk fisik. Fungsi ini berjalan ketika sekretaris menekan tombol *print* pada aplikasi yang sudah di estimasi.

Gambar 6.12 dan 6.13 menampilkan hasil cetak dokumen penawaran dalam bentuk dokumen.



**Gambar 6. 12 Hasil cetak dokumen penawaran halaman 1**

**PELAKSANAAN**

**Rincian Personil Project**

Estimasi dalam menyelesaikan project ini adalah - dilaksanakan oleh 3 orang anggota Project Team dengan rincian, sebagai berikut :

Nama	Job Description
Dewangga Prasetya Praja	Programmer
Naufal Raihan Noly	System Analyst
Fandhi Akhmad	Project Manager

**Rincian Biaya**

**1. Rincian Biaya Pengembangan Perangkat Lunak**

Nama Aktivitas	Nilai Biaya (Rp)
Requirements	2.763.687,50
Specifications & Design	6.448.750,00
Coding	2.632.187,50
Testing	2.948.000,00
Project management	7.370.000,00
Configuration Management	1.105.562,50
Documentation	631.750,00
Training & Support	1.105.562,50
Acceptance & Deployment	5.265.000,00
Quality Assurance & Control	12.992.500,00
Evaluation and Testing	10.385.500,00
<b>Total</b>	<b>Rp 53.648.500,00</b>

**2. Rincian Biaya Operasional**

**Gambar 6. 13 Hasil cetak dokumen penawaran halaman 2**

- Hasil akhir estimasi  
Halaman hasil akhir estimasi muncul ketika seorang analis telah memasukkan nilai *actual effort* sehingga data estimasi tidak dapat diganti atau diubah. Gambar 6.14 menjelaskan tampilan halaman hasil akhir estimasi yang dilakukan.

No	Tanggal estimasi	Nama Client	Nama Aplikasi	CFP	RCAF	Effort Estimate	Effort Real	Biaya Estimasi (Total)	Tim Pengembang	Status	Actions
1	05-07-2018	Biapak Fandhi	Aplikasi FP	96.00	42.00	842.30	504.00	Rp. 53.648.500,00	Tim Aplikasi FP	Done	<a href="#">Lihat Detail</a>

**Gambar 6. 14 Halaman hasil akhir estimasi**

- Halaman *Log Estimasi*  
Halaman *log estimasi* adalah halaman yang digunakan untuk menampilkan data-data estimasi yang sudah pernah dilakukan oleh pengguna. Gambar 6.15 menampilkan hasil akhir estimasi yang tidak bisa diubah atau diganti.

Daftar Log Estimasi Aplikasi

10 records per page Search:

No	Tanggal estimasi	Nama Client	Nama Aplikasi	Biaya Estimasi (Total)	Tim Pengembang	Status	Actions
1	31-05-2018	Fandhi Akhmad	Aplikasi FP	Rp. 34.863.500,00	<a href="#">Tim Aplikasi FP</a>	Belum lengkap	<a href="#">Edit</a>
2	28-05-2018	awdwa	Event Extract	Rp. 30.109.750,00	<a href="#">Tim Event Extract</a>	Belum lengkap	<a href="#">Edit</a>
3	28-05-2018	adwa	Event Extract	Rp. 156.846.437,50	<a href="#">Tim Event Extract</a>	Belum lengkap	<a href="#">Edit</a>
4	28-05-2018	Umar Abdul Aziz	Aplikasi FP	Rp. 21.741.500,00	<a href="#">Tim Aplikasi FP</a>	Belum lengkap	<a href="#">Edit</a>
5	28-05-2018	Oman	Event Extract	Rp. 21.741.500,00	<a href="#">Tim Event Extract</a>	Belum lengkap	<a href="#">Edit</a>

**Gambar 6. 15 Halaman log estimasi**

## 6.2 Hasil Pengujian Aplikasi

Pengujian dilakukan menggunakan *unit testing*. Gambar 6.16 menampilkan hasil menjalankan *unit testing* yang dilakukan menggunakan *command window*. Hasil dari *unit testing* yang dilakukan terdiri dari 29 skenario dimana 25 berhasil cocok antara ekspetasi dengan hasil sedangkan 4 mengalami *error*.

```
←[37;41m←[2KFAILURES!  
←[0m←[37;41m←[2KTests: 22, Assertions: 25, Errors: 4.  
←[0m←[2K  
A PHP Error was encountered  
  
Severity:      Core Warning  
Message:      Module 'openssl' already loaded  
Filename:     Unknown  
Line Number:  0  
  
Backtrace:
```

Gambar 6. 16 Hasil running kode pengujian

Pengujian dilakukan berdasarkan prosedur pengujian yang telah dibuat sebelumnya. Peneliti melakukan pengujian dengan menggunakan *unit testing*. Tabel 6.1 menjelaskan hasil pengujian aplikasi berdasarkan skenario yang telah dibuat pada bagian perancangan.

**Tabel 6. 1 Hasil pengujian aplikasi estimasi biaya perangkat lunak**

Skenario	No	Test Case	Output Ekspetasi	Output	Kesimpulan
Melakukan Login	001	Membuka halaman login aplikasi /FunctionPoint/login	Berhasil ke halaman login	Sistem menampilkan halaman login aplikasi	Berhasil dan sesuai dengan ekspetasi
	002	Login dengan user project manager dengan password yang sesuai	Masuk ke halaman homepage	Sistem menampilkan halaman homepage aplikasi	Berhasil dan sesuai dengan ekspetasi
	003	Login dengan user project manager menggunakan password yang tidak sesuai	Kembali ke halaman login	Sistem menampilkan halaman login aplikasi	Berhasil dan sesuai dengan ekspetasi
	004	Login dengan user projectmanager dengan password yang dikosongkan	Kembali ke halaman login	Sistem menampilkan halaman login aplikasi	Berhasil dan sesuai dengan ekspetasi
Melakukan Logout	001	Melakukan klik tombol logout	Berhasil ke halaman login	Sistem menampilkan	Berhasil dan sesuai dengan ekspetasi

				halaman login aplikasi	
	002	Melakukan logout tanpa login	Kembali ke halaman login	Sistem tidak menampilkan halaman login aplikasi	Tidak berhasil
Membuka halaman homepage aplikasi	001	Membuka halaman awal aplikasi dengan melakukan login	Berhasil masuk ke halaman homepage	Sistem menampilkan halaman homepage	Berhasil dan sesuai dengan ekspetasi
	002	Membuka halaman awal aplikasi tanpa melakukan login	Kembali ke halaman aplikasi	Sistem menampilkan halaman login aplikasi	Berhasil dan sesuai dengan ekspetasi
Masuk ke halaman deskripsi aplikasi	001	Masuk ke halaman deskripsi aplikasi tanpa melakukan login	Kembali ke halaman aplikasi	Sistem menampilkan halaman login aplikasi	Berhasil dan sesuai dengan ekspetasi
	002	Masuk ke halaman deskripsi aplikasi dengan melakukan login	Berhasil masuk ke halaman deskripsi aplikasi	Sistem menampilkan	Berhasil dan sesuai dengan ekspetasi

				halaman deskripsi aplikasi	
Memasukkan data deskripsi aplikasi ke dalam database	001	Memasukkan data deskripsi aplikasi	Menampilkan pesan “Data berhasil disimpan” dan masuk ke halaman edit aplikasi	Sistem menampilkan pesan top up “data berhasil disimpan” dan sistem menampilkan halaman edit aplikasi	Berhasil dan sesuai dengan ekspetasi
Memeriksa data deskripsi aplikasi di dalam database	001	Memeriksa data deskripsi aplikasi	Menampilkan data deskripsi aplikasi dari database	Sistem menampilkan isi data deskripsi aplikasi	Berhasil dan sesuai dengan ekspetasi
Masuk ke halaman edit deskripsi aplikasi	001	Masuk ke halaman edit deskripsi aplikasi tanpa melakukan login	Kembali ke halaman aplikasi	Sistem menampilkan halaman login aplikasi	Berhasil dan sesuai dengan ekspetasi

	002	Masuk ke halaman edit deskripsi aplikasi dengan melakukan login	Berhasil masuk ke halaman edit deskripsi aplikasi	Sistem menampilkan halaman edit deskripsi aplikasi	Berhasil dan sesuai dengan ekspektasi
Melakukan perubahan terhadap data deskripsi aplikasi	001	Mengubah data deskripsi aplikasi	Menampilkan pesan “Data berhasil disimpan” dan masuk ke halaman edit deskripsi aplikasi	Sistem menampilkan pesan top up “data berhasil disimpan” dan menampilkan halaman edit deskripsi aplikasi	Berhasil dan sesuai dengan ekspektasi
Menghapus data deskripsi aplikasi	001	Menghapus data deskripsi aplikasi di dalam database	Menampilkan pesan “Data berhasil dihapus” dan kembali ke halaman deskripsi aplikasi	Sistem menampilkan pesan “data berhasil di hapus”	Berhasil dan sesuai dengan ekspektasi
Masuk ke halaman CFP	001	Masuk ke halaman CFP tanpa melakukan login	Kembali ke halaman login aplikasi	Sistem menampilkan halaman login aplikasi	Berhasil dan sesuai dengan eskpetasi

	002	Masuk ke halaman CFP dengan melakukan login	Berhasil masuk ke dalam aplikasi dan menuju halaman CFP	Sistem menampilkan halaman CFP	Berhasil dan sesuai dengan ekspetasi
Melakukan pengisian indikator CFP	001	Memasukkan nilai CFP	Menampilkan pesan “data berhasil disimpan” kemudian masuk ke halaman edit CFP	Sistem menampilkan pesan top up “data berhasil disimpan”	Berhasil dan sesuai dengan ekspetasi
Memeriksa data CFP yang sudah masuk di dalam database	001	Memeriksa data CFP di database	Menampilkan data penilaian CFP	Sistem menampilkan log penilaian CFP	Berhasil dan sesuai dengan ekspetasi
Masuk ke halaman edit CFP	001	Masuk ke halaman edit CFP tanpa melakukan login	Kembali ke halaman login aplikasi	Sistem menampilkan halaman login aplikasi	Berhasil dan sesuai dengan ekspetasi

	002	Masuk ke halaman edit CFP dengan melakukan login	Berhasil login dan menuju ke halaman edit CFP	Sistem menampilkan halaman edit CFP	Berhasil dan sesuai dengan ekspetasi
Melakukan perubahan terhadap data CFP	001	Mengubah data penilaian CFP yang sudah ada	Menampilkan pesan “Data berhasil disimpan” dan kembali ke halaman edit CFP	Sistem menampilkan pesan top up “data berhasil disimpan”	Berhasil dan sesuai dengan ekspetasi
Menghapus data CFP	001	Menghapus data CFP yang ada di database	Menampilkan pesan “Data berhasil dihapus” dan kembali ke halaman CFP	Sistem menampilkan pesan “data berhasil di hapus”	Berhasil dan sesuai dengan ekspetasi
Masuk ke halaman RCAF	001	Masuk ke halaman RCAF tanpa melakukan login	Kembali ke halaman login aplikasi	Sistem menampilkan halaman login	Berhasil dan sesuai dengan ekspetasi
	002	Masuk ke halaman RCAF dengan melakukan login	Berhasil masuk ke dalam aplikasi dan menuju halaman RCAF	Sistem menampilkan halaman RCAF	Berhasil dan sesuai dengan ekspetasi

Melakukan pengisian nilai RCAF	001	Memasukkan data RCAF	Menampilkan pesan “Data berhasil disimpan” dan kembali ke halaman edit RCAF	Sistem menampilkan pesan top up “data berhasil disimpan”	Berhasil dan sesuai dengan ekspetasi
Memeriksa nilai RCAF	001	Memeriksa nilai RCAF yang sudah masuk di database	Menampilkan data RCAF yang ada di database	Sistem menampilkan log penilaian RCAF	Berhasil dan sesuai dengan ekspetasi
Masuk ke halaman edit RCAF	001	Masuk ke halaman edit RCAF tanpa melakukan login	Kembali ke halaman login aplikasi	Sistem menampilkan halaman login aplikasi	Berhasil dan sesuai dengan ekspetasi
	002	Masuk ke halaman edit RCAF dengan melakukan login	Berhasil login ke aplikasi dan menuju halaman edit RCAF	Sistem menampilkan halaman edit RCAF	Berhasil dan sesuai dengan ekspetasi
Melakukan perubahan nilai RCAF	001	Mengubah data RCAF di database	Menampilkan pesan “Data berhasil disimpan”	Sistem menampilkan	Berhasil dan sesuai dengan ekspetasi

			dan kembali ke halaman edit RCAF	pesan top up “data berhasil di simpan”	
Menghapus nilai RCAF	001	Menghapus nilai RCAF yang ada di database	Menampilkan pesan “Data berhasil di hapus”	Sistem menampilkan pesan top up “data berhasil di hapus”	Berhasil dan sesuai dengan ekspetasi
Masuk ke halaman result	001	Masuk ke halaman result tanpa melakukan login	Kembali kehalaman login	Sistem menampilkan halaman login aplikasi	Berhasil dan sesuai dengan ekspetasi
	002	Masuk ke halaman result dengan melakukan login	Berhasil masuk ke dalam aplikasi dan menuju ke halaman result	Sistem menampilkan halaman result	Berhasil dan sesuai dengan ekspetasi

### 6.3 Validasi Aplikasi

Validasi aplikasi digunakan untuk memastikan bahwa aplikasi sesuai dengan kebutuhan dan perancangan yang sudah dibuat. Validasi ini dilakukan dengan menguji hasil perhitungan metode *function point* pada aplikasi yang sudah dibuat dengan kertas kerja yang sudah dipilih sebelumnya pada tahapan analisis kebutuhan. Validasi dilakukan dengan memasukkan data *dummy* yang dilakukan oleh peneliti untuk mendapatkan hasil *function point* yang sama dengan memasukkan faktor *crude function point* dan *relative complexity adjustment factor* yang sama. Berikut adalah hasil validasi dari aplikasi estimasi biaya perangkat lunak menggunakan metode *function point* dengan kertas kerja *function point* yang sudah ada.

#### a. Kertas Kerja *Function Point*

##### 1. Perhitungan CFP

Perhitungan yang dilakukan oleh peneliti dengan memasukkan angka satu disetiap indikator yang perlu diisi dalam perhitungan CFP.

Item	Item Description	Complexity	Count	Weight	Weighted Count
1	Number of User Inputs	Simple	1	3	3
		Average	1	4	4
		Complex	1	6	6
2	Number of User Oututs	Simple	1	4	4
		Average	1	5	5
		Complex	1	7	7
3	Number of User Inquiries	Simple	1	3	3
		Average	1	4	4
		Complex	1	6	6
4	Number of Files	Simple	1	7	7
		Average	1	10	10
		Complex	1	15	15
5	Number of External Interfaces	Simple	1	5	5
		Average	1	7	7
		Complex	1	10	10
<b>Total Weighted Function Count (FC)</b>					<b>96</b>

**Gambar 6. 17** Hasil perhitungan CFP pada kertas kerja

## 2. Perhitungan RCAF

Perhitungan yang dilakukan oleh peneliti dengan memasukkan angka tiga di setiap indikator yang perlu diisi dalam perhitungan RCAF.

Factor	Description	Rating
		0=Irrelevant
		5=Essential
F1	Reliability and backup recovery	3
F2	Data communications	3
F3	Distributed processing	3
F4	Performance	3
F5	Operate on existing system	3
F6	On-line data entry	3
F7	Data entry over multiple screens	3
F8	Master files updated on-line	3
F9	Complex inputs, outputs, files & inquiries	3
F10	Complex internal processing	3
F11	Code needs to be reusable	3
F12	Need conversion and installation	3
F13	Multiple installations of the system	3
F14	Easy to change and use	3
<b>Complexity Factor (CF) = sum of ratings</b>		<b>42</b>

Gambar 6. 18 Hasil perhitungan RCAF pada kertas kerja

3. Hasil *function point*

Hasil *function point* yang didapatkan oleh peneliti dengan masukkan angka satu di setiap indikator CFP dan angka tiga di setiap indikator RCAF.

Function Points	
<b>Function Points (FP) = FC x (0.65 + 0.01 x CF)</b>	<b>102.72</b>

Gambar 6. 19 Hasil nilai function point pada kertas kerja

b. Aplikasi Estimasi Biaya Perangkat Lunak

1. Perhitungan CFP

Perhitungan yang dilakukan oleh peneliti dengan memasukkan angka satu di setiap indikator yang perlu diisi dalam perhitungan CFP.

11*	Number Of Files	average	<input type="text" value="1"/>
12*	Number Of Files	complex	<input type="text" value="1"/>
13*	Number Of External Interfaces	simple	<input type="text" value="1"/>
14*	Number Of External Interfaces	average	<input type="text" value="1"/>
15*	Number Of External Interfaces	complex	<input type="text" value="1"/>

NB: \* Wajib Diisi

Perbarui

Nilai CFP : 96

**Gambar 6. 20 Perhitungan CFP pada aplikasi estimasi biaya perangkat lunak**

2. Perhitungan RCAF

Perhitungan yang dilakukan oleh peneliti dengan memasukkan angka tiga di setiap indikator yang perlu diisi dalam perhitungan RCAF.

11*	Tingkat Kemudahan Dalam Instalasi	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input checked="" type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5
12*	Tingkat Kemudahan Operasional Software	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input checked="" type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5
13*	Tingkat Software dibuat untuk multi organisasi/perusahaan client	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input checked="" type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5
14*	Tingkat Kompleksitas dalam mengikuti perubahan/fleksibel	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input checked="" type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5

NB: \* Wajib Diisi

Nilai RCAF : 42

**Gambar 6. 21 Perhitungan RCAF pada aplikasi estimasi biaya perangkat lunak**

### 3. Hasil *function point*

Hasil nilai *function point* yang didapatkan oleh peneliti dengan masukan angka satu di setiap indikator CFP dan angka tiga di setiap indikator RCAF.

## Hasil Akhir Perhitungan Nilai FP

Nilai CFP ( <i>Crude Function Point</i> )	:	96.00
Nilai RCAF ( <i>Relative Complexity Adjustment Factor</i> )	:	42.00
Nilai FP $CFP * (0.65 + 0.01 * RCAF)$	:	102.72

**Gambar 6. 22 Hasil nilai function point pada aplikasi function point**

Berdasarkan proses validasi yang dilakukan dengan melakukan perhitungan *function point* menggunakan kertas kerja dan aplikasi yang dibuat telah didapatkan hasil nilai *function point* yang sama. Pada penilaian indikator CFP dimasukkan angka satu di setiap indikator sehingga mendapatkan hasil CFP adalah

96. Pada penilaian indikator RCAF dimasukkan angka tiga di setiap indikator sehingga mendapatkan hasil RCAF adalah 42. Berdasarkan nilai tersebut yang kemudian dimasukkan ke dalam formula metode *function point* sehingga menghasilkan nilai *function point* sebesar 102.72 pada kertas kerja yang telah dipilih dan aplikasi *function point* yang sudah dibuat.

#### 6.4 Data Percobaan

Nama Perusahaan : Dynamic Team Solution

Client : Pemerintah Daerah XYZ

**Tabel 6. 2 Proyek perusahaan DTS**

No.	Nama Proyek Perangkat Lunak	Jenis	Waktu
1.	Tanda Daftar Industri	Perizinan	2 Bulan
2.	Izin Usaha Industri	Perizinan	2 Bulan
3.	Persetujuan Prinsip	Perizinan	2 Bulan
4.	Tanda Daftar Perusahaan	Perizinan	4 Bulan

Keterangan :

1. Nama Perangkat Lunak : Tanda Daftar Industri (TDI)  
Aplikasi pelayanan publik ini digunakan untuk melakukan pendaftaran industri skala kecil dengan modal kurang dari Rp. 10 Miliar. Aplikasi perizinan ini dikelola oleh Dinas Perdagangan dan Perindustrian.
2. Nama Perangkat Lunak : Izin Usaha Industri (IUI)  
Aplikasi pelayanan publik ini digunakan untuk mendaftarkan usaha industri skala besar dengan modal lebih dari Rp. 10 Miliar. Aplikasi perizinan ini dikelola oleh Dinas Perdagangan dan Perindustrian.
3. Nama Perangkat Lunak : Persetujuan Prinsip (PP)  
Aplikasi pelayanan publik ini digunakan untuk mendaftarkan usaha industri yang belum beroperasi sehingga masih dalam tahap pembangunan pabrik. Aplikasi perizinan ini dikelola oleh Dinas Perdagangan dan Perindustrian.

4. Nama Perangkat Lunak : Tanda Daftar Perusahaan (TDP)

Aplikasi pelayanan publik ini digunakan untuk mendaftarkan segala jenis usaha yang telah memiliki izin teknis seperti TDI, IUI, PP, dan lain-lain sehingga TDP merupakan legalitas yang wajib dimiliki setiap pengusaha. Aplikasi perizinan ini dikelola oleh Dinas Perdagangan dan Perindustrian.

Dalam penelitian ini penulis hanya menggunakan satu data proyek yang telah disebutkan diatas yaitu proyek Tanda Daftar Industri. Berikut adalah data dari proyek Tanda Daftar Industri (TDI).

a. Nilai CFP

Tabel 6. 3 Penilaian CFP

Tipe Komponen	Level Kompleksitas									Total
	Low			Average			High			
	Count	Weighting Factor	Point	Count	Weighting Factor	Point	Count	Weighting Factor	Point	
EI	17	3	51	3	4	12	4	6	24	87
EO	5	4	20	1	5	5	0	7	0	25
EQ	5	3	15	0	4	0	1	6	6	21
ILF	5	7	35	2	10	20	0	15	0	55
EIF	2	5	10	0	7	0	0	10	0	10
<b>Total Unadjusted Function Point</b>										<b>198</b>

b. Nilai RCAF

Tabel 6. 4 Penilaian RCAF

No	Characterictics	Project
		TDI
1	Data Communication	5
2	Distributed data processing	4

No	Characteristics	Project
		TDI
3	Performance	4
4	Heavily used configuration	2
5	Transaction rate	1
6	On-line data entry	5
7	End-user efficiency	5
8	On-line update	4
9	Complex processing	3
10	Reusability	5
11	Installation ease	5
12	Operational ease	3
13	Multiple sites	5
14	Facilitate change	3
<b>Total Degree of Influence =</b>		<b>54</b>

## 6.5 Perhitungan Biaya dengan Aplikasi Estimasi Biaya Perangkat Lunak

Setelah data diperoleh tahapan selanjutnya yang dilakukan yaitu memasukkan data kedalam aplikasi ini. Berikut adalah hasil dari memasukkan data ke dalam aplikasi.

### 1. Perhitungan CFP

11*	Number Of Files	average	<input type="text" value="2"/>
12*	Number Of Files	complex	<input type="text" value="0"/>
13*	Number Of External Interfaces	simple	<input type="text" value="2"/>
14*	Number Of External Interfaces	average	<input type="text" value="0"/>
15*	Number Of External Interfaces	complex	<input type="text" value="0"/>

NB: \* Wajib Diisi

Nilai CFP : 198

**Gambar 6. 23 Hasil perhitungan CFP**

Gambar 6.23 menunjukkan nilai dari masing-masing indikator *crude function point* yang sudah dimasukkan berdasarkan proyek aplikasi TDI. Masing-masing nilai yang sudah dimasukkan akan secara otomatis dihitung dengan bobot yang sudah ada di dalam aplikasi. Pada gambar diatas juga menampilkan hasil dari perhitungan CFP berupa nilai CFP.

## 2. Perhitungan RCAF

11*	Tingkat Kemudahan Dalam Instalasi	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input checked="" type="radio"/> 5
12*	Tingkat Kemudahan Operasional Software	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input checked="" type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5
13*	Tingkat Software dibuat untuk multi organisasi/perusahaan client	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input checked="" type="radio"/> 5
14*	Tingkat Kompleksitas dalam mengikuti perubahan/fleksibel	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input checked="" type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5

NB: \* Wajib Diisi

Nilai RCAF : 54

**Gambar 6. 24 Hasil perhitungan RCAF**

Gambar 6.24 menunjukkan nilai dari masing-masing indikator *relative complexity adjustment factor* yang sudah dimasukkan berdasarkan proyek aplikasi TDI. Masing-masing nilai yang sudah dimasukkan akan secara otomatis terhitung ke dalam aplikasi. Gambar diatas juga menunjukkan hasil dari perhitungan RCAF berupa nilai RCAF.

## 3. Rekapitulasi Keseluruhan Perhitungan

**\$ Form Estimasi Harga Perangkat Lunak**

Informasi Client    Deskripsi Aplikasi    CFP    RCAF    **Result**

---

**Hasil Akhir Perhitungan Nilai FP**

Nilai CFP ( <i>Crude Function Point</i> )	: 198.00
Nilai RCAF ( <i>Relative Complexity Adjustment Factor</i> )	: 54.00
Nilai FP $CFP * (0.65 + 0.01 * RCAF)$	: 235.62

**Gambar 6. 25 Hasil perhitungan Function Point**

Gambar 6.25 menunjukkan hasil dari perhitungan formula *function point* yang sudah dilakukan oleh aplikasi. Pada halaman tersebut juga menunjukkan nilai CFP dan nilai RCAF.

---

## Hasil Perhitungan Usaha

Nilai Effort Rate (Konstanta)	:	8.2
Nilai Total Usaha (Hour of Effort)	:	1932.084

**Gambar 6. 26 Hasil perhitungan Effort**

Gambar 6.26 menunjukkan hasil dari perhitungan usaha yang sudah dilakukan oleh aplikasi. Pada halaman tersebut juga menampilkan nilai rata-rata usaha berupa konstanta nilai *man-hour-effort* yaitu sebesar 8.2.

## Rekap Biaya Tiap Aktivitas

 Edit

Aktivitas	Effort		Biaya		
	%	Hour of Effort	Standard gaji/bulan(Rp)	Gaji per Jam (Rp)	Biaya (Rp)
<b>SOFTWARE DEVELOPMENT</b>					
Requirements	7.5 %	144,91	7.000.000,00	43.750,00	6.339.812,50
Specifications & Design	17.5 %	338,11	7.000.000,00	43.750,00	14.792.312,50
Coding	10 %	193,21	5.000.000,00	31.250,00	6.037.812,50
Testing	7 %	135,25	5.000.000,00	31.250,00	4.226.562,50
<b>Sub Total</b>	<b>42,00%</b>	<b>811,48</b>			<b>31.396.500,00</b>
<b>ON GOING ACTIVITY</b>					
Project management	7 %	135,25	8.500.000,00	53.125,00	7.185.156,25
Configuration Management	3 %	57,96	7.000.000,00	43.750,00	2.535.750,00
Documentation	3 %	57,96	4.000.000,00	25.000,00	1.449.000,00
Training & Support	3 %	57,96	7.000.000,00	43.750,00	2.535.750,00

**Gambar 6. 27 Hasil perhitungan rekap biaya**

Gambar 6.27 menunjukkan nilai usaha dan biaya pada masing-masing aktivitas yang ada pada proyek aplikasi TDI. Nilai biayanya yang ada didasarkan pada masing-masing standard gaji yang ada pada masing-masing aktivitas.

## Biaya Keseluruhan

No	Deskripsi Biaya	Jumlah (Rp)
1	Biaya Pokok Pembuatan Perangkat Lunak	77.789.156,25
2	Total Biaya Operasional	0,00
<b>TOTAL KESELURUHAN</b>		<b>77.789.156,25</b>

**Gambar 6. 28 Hasil biaya keseluruhan**

Gambar 6.28 menunjukkan hasil biaya keseluruhan berdasarkan rekap perhitungan biaya aktivitas yang ada pada proyek aplikasi TDI.

### 6.6 Pembahasan

Berdasarkan kondisi sekarang, aplikasi *function point* sudah banyak dibuat oleh *developer* dari mulai bentuk hanya dalam excel hingga bentuk *website*. Aplikasi *function point* yang sekarang memiliki kemampuan sebagai berikut :

1. Aplikasi *function point* dapat menghitung perhitungan *crude function point*.
2. Aplikasi *function point* dapat menghitung perhitungan bobot kompleksitas atau dikenal dengan nilai *relative complexity adjustment factor*.
3. Aplikasi *function point* yang sekarang hanya sebatas mendapatkan nilai *function point*.
4. Aplikasi *function point* yang sekarang hanya menjelaskan mengenai pengertian metode tanpa memberi penjelasan bagaimana cara melakukan pengisian perhitungan *function point*.

Berdasarkan empat poin diatas, dirasa kurang membantu orang dalam melakukan estimasi biaya perangkat lunak dimana peneliti membuat aplikasi estimasi biaya perangkat lunak menggunakan metode *function point* dengan adanya

penambahan kompleksitas proses bisnis secara lebih kompleks dengan memiliki kemampuan sebagai berikut :

1. Aplikasi estimasi biaya perangkat lunak ini dibuat sesuai dengan proses bisnis dari perhitungan *function point*. Penggunaan metode estimasi ini bertujuan untuk memberikan acuan atau landasan bagi tim suatu proyek dalam menentukan estimasi biaya perangkat lunak yang dikerjakan. Selain itu, dengan menggunakan metode ini, estimasi perangkat lunak akan lebih akurat dan mendekati harga yang sebenarnya.
2. Aplikasi estimasi biaya perangkat lunak ini memiliki proses penggunaan aplikasi yang berhubungan dengan alur proses bisnis yang ada di dalam suatu proyek dimana terdiri dari tiga pengguna yaitu *project manager*, analis dan sekretaris yang memiliki peran masing-masing.
3. Aplikasi estimasi biaya perangkat lunak ini menyediakan fitur jika pengguna memiliki *client* serta memasukkan deskripsi aplikasi yang nama aplikasi dan fitur yang ada pada aplikasi tersebut.
4. Aplikasi estimasi biaya perangkat lunak ini menyediakan fitur dimana pengguna dapat memilih metode *function point* yang tidak dimodifikasi atau metode *function point* yang sudah dimodifikasi dengan adanya penambahan tingkat kompleksitas proses bisnis pada karakteristik perangkat lunak sehingga aplikasi ini bisa mengikuti perkembangan pengembangan perangkat lunak di masa sekarang.
5. Aplikasi estimasi biaya perangkat lunak ini menyediakan fitur dimana pengguna dapat memilih *effort rate* yang nantinya digunakan dalam distribusi usaha dengan pilihan untuk proyek kecil-menengah, proyek kecil, menengah dan kompleks sehingga pengguna dapat menyesuaikan *effort rate* yang digunakan dengan jenis proyek yang dikerjakan.

6. Aplikasi estimasi biaya perangkat lunak ini menyediakan fitur dimana pengguna dapat memasukkan fitur yang ada pada aplikasi yang di estimasi sehingga data dari aplikasi lebih detail dan dapat dijadikan acuan sebagai masukan perhitungan nilai *function point*.
7. Aplikasi estimasi biaya perangkat lunak ini dapat melakukan perhitungan *crude function point* dan *relative complexity adjustment factor* dengan menyediakan tombol bantuan bagaimana cara mengisi penilaian kedua hal tersebut.
8. Aplikasi estimasi biaya perangkat lunak ini tidak sekedar hanya mendapatkan nilai *function point* tetapi nilai *function point* yang didapatkan nantinya akan dikonversi ke dalam bentuk harga dengan menggunakan distribusi usaha sehingga pengguna dapat mengetahui harga dari perangkat lunak yang di estimasi.
9. Aplikasi estimasi biaya perangkat lunak ini menyediakan fitur untuk memberikan notifikasi dalam bentuk *email* yang dikirimkan langsung kepada *project manager* setiap kali analis selesai melakukan estimasi harga. Hal ini bertujuan untuk meminimalisir *miss communication* dalam hal validasi hasil estimasi. Selain itu dengan adanya fitur ini seorang *project manager* bisa mengetahui aplikasi apa saja yang sudah diestimasi oleh analis dan status dari aplikasi yang akan di estimasi.
10. Aplikasi estimasi biaya perangkat lunak ini menyediakan fitur *print* dimana hasil estimasi yang sudah dilakukan dalam aplikasi dapat dibentuk ke dalam bentuk dokumen secara terstruktur sehingga dapat digunakan dengan baik.
11. Aplikasi estimasi biaya perangkat lunak ini menyediakan halaman dimana pengguna dapat melihat riwayat estimasi yang sudah pernah dilakukan sehingga

pengguna dapat mengetahui estimasi aplikasi apa saja yang sudah pernah dilakukan.

12. Aplikasi estimasi biaya perangkat lunak ini menyediakan fitur dimana pengguna dapat menambahkan biaya operasional diluar biaya aktivitas proyek yang sudah ada sehingga biaya non proyek yang dikeluarkan dapat dimasukkan juga ke dalam biaya total proyek pengembangan perangkat lunak tersebut.
13. Aplikasi estimasi biaya perangkat lunak ini menyediakan fitur pengelola pengguna, pengelola profesi, pengelola tim dan aktivitas sehingga pengguna dapat menyesuaikan dengan kondisi tim proyek.
14. Aplikasi estimasi biaya perangkat lunak ini menyediakan fitur memasukkan gaji sesuai dengan tim proyek sehingga estimasi harga yang didapatkan berdasarkan *pay rate* yang dimiliki oleh pengguna tetapi aplikasi *function point* ini sudah menyediakan standar gaji yang dikeluarkan oleh Kelly Service pada tahun 2017. Dengan adanya fitur ini, estimasi yang dilakukan oleh suatu tim proyek dapat sesuai dengan kondisi tim proyek tersebut.
15. Aplikasi estimasi biaya perangkat lunak ini menggunakan *user interface* yang mudah bagi pengguna sehingga pengguna tidak mengalami kesulitan dalam menggunakan aplikasi tersebut.

## **BAB VII**

### **KESIMPULAN DAN SARAN**

Bab ini membahas mengenai kesimpulan yang diperoleh dari semua proses yang telah dilakukan dan menguraikan beberapa saran terhadap pengembangan selanjutnya.

#### **7.1 Kesimpulan**

Berdasarkan proses-proses yang telah dilakukan dalam pengerjaan tugas akhir ini maka ada beberapa kesimpulan yang dapat diambil diantaranya sebagai berikut:

1. Spesifikasi kebutuhan perangkat lunak pada aplikasi ini dibuat berdasarkan aktivitas yang ada dalam suatu proyek.
2. Rancangan dari aplikasi ini dibuat berdasarkan hasil analisis kebutuhan pengguna dan kebutuhan fungsional aplikasi.
3. Aplikasi estimasi biaya perangkat lunak ini dibuat berdasarkan alur yang ada pada metode *function point*.
4. Metode pengembangan yang digunakan untuk mengembangkan aplikasi estimasi biaya perangkat lunak ini yaitu dengan metode *waterfall*.
5. Pembangunan aplikasi menggunakan metode *waterfall* karena memiliki struktur kerja yang baik khususnya dalam hal dokumentasi.
6. Tahapan implementasi aplikasi estimasi biaya perangkat lunak menggunakan metode *function point* menghasilkan aplikasi estimasi biaya perangkat lunak berbasis *website* yang sudah diuji dengan menggunakan metode pengujian *whitebox testing* berupa beberapa *test case*.
7. Terdapat beberapa perbaikan dari hasil pengujian dalam hal fungsional aplikasi.
8. Perbaikan pengujian ini dilakukan pada tahapan testing yang ada pada metode *waterfall*.

9. Kesesuaian kebutuhan fungsional sistem dengan implementasi aplikasi dicapai karena menyesuaikan kondisi yang dibutuhkan suatu proyek dalam membangun suatu aplikasi.
10. Keunggulan aplikasi estimasi biaya perangkat lunak ini dibandingkan dengan aplikasi estimasi lainnya yang menggunakan metode yang sama yaitu adanya fitur memilih metode *function point* yang sudah dimodifikasi dengan adanya penambahan karakteristik perangkat lunak yaitu tingkat kompleksitas proses bisnis dan pemilihan *effort rate* yang sudah disediakan berdasarkan model yang digunakan.
11. Aplikasi ini dapat menghasilkan estimasi biaya perangkat lunak dengan menggunakan distribusi usaha suatu proyek serta menggunakan pilihan *effort rate* untuk proyek sederhana, proyek menengah dan proyek kompleks.
12. Dengan adanya penambahan kompleksitas membuat hasil dari *function point* yang berbeda dengan metode *function point* yang sudah ada sebelumnya.
13. Secara *general* kelebihan aplikasi ini dapat membantu suatu tim proyek dalam mendapatkan biaya estimasi perangkat lunak menggunakan metode *function point*.

## 7.2 Saran

Berdasarkan kesimpulan dari tugas akhir ini, maka penulis merekomendasikan beberapa saran yang diberikan untuk pengembangan kedepan sebagai berikut:

1. Rancang bangun aplikasi estimasi biaya perangkat lunak menggunakan metode *function point* berikutnya lebih baik menggunakan kode program yang berbeda karena kode program aplikasi ini sulit dalam hal *maintainability*.
2. *Uni testing* yang dilakukan hanya terbatas pada fitur perhitungan estimasi harga menggunakan *function point* sampai dapat mencetak dokumen penawaran

sehingga perlu dilakukan *unit testing* untuk fitur-fitur yang belum di tes.

3. Aplikasi estimasi biaya perangkat lunak menggunakan metode *function point* ini bisa dikaitkan dengan *time management* berupa estimasi waktu yang dibutuhkan dalam pengerjaan proyek untuk menentukan *timeline* suatu proyek pengembangan perangkat lunak.
4. Fitur *effort rate* pada aplikasi ini bisa dibuat menjadi dinamis sehingga menghasilkan estimasi yang bisa lebih akurat.
5. Notifikasi hasil estimasi yang dikirimkan tidak hanya dalam bentuk *email* tetapi bisa juga dalam bentuk *sms gateway* sehingga pengiriman hasil estimasi bisa lebih cepat sampai ke penerima.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] M. F. FANANI, “Rancang Bangun Aplikasi Use Case Point Untuk Estimasi Harga Perangkat Lunak ( Studi Kasus : Perusahaan DTS ),” 2015.
- [2] Universitas and M. Bandung, “Manajemen Proyek IT,” *Network*, pp. 1–41.
- [3] Tanto, W. W. Wing, and A. R. M, “Analisis Penerapan Sistem Enterprise Resource Planning Pada PT. Aneka Dharma Persada di Yogyakarta,” vol. 8, no. 2, pp. 41–54, 2015.
- [4] N. Balaji, N. Shivakumar, and V. V. Ananth, “Software cost estimation using function point with non algorithmic approach,” *Glob. J. Comput. Sci. Technol.*, vol. 13, no. 8, 2013.
- [5] V. Tunali, “Software Size Estimation Using Function Point Analysis – A Case Study for a Mobile Application,” *7. Mühendislik ve Teknol. Sempozyumu*, no. May, pp. 73–76, 2014.
- [6] Ifpug, “Function Point Counting Practices Manual,” *Group*, vol. on06/23/, p. 02-17-, 2010.
- [7] R. S. Dewi, A. P. Subriadi, and Sholiq, “A Modification Complexity Factor in Function Points Method for Software Cost Estimation Towards Public Service Application,” *Procedia Comput. Sci.*, vol. 124, pp. 415–422, 2017.
- [8] M. A. Al-Hajri, A. A. A. Ghani, M. N. Sulaiman, and M. H. Selamat, “Modification of standard function point complexity weights system,” *J. Syst. Softw.*, vol. 74, no. 2 SPEC. ISS., pp. 195–206, 2005.
- [9] B. Boehm, C. Abts, and S. Chulani, “Software development cost estimation approaches—A survey,” *Ann. Softw. Eng.*, vol. 10, no. 1–4, pp. 177–205, 2000.
- [10] D. Pratiwi, “Implementation of Function Point Analysis in Measuring The Volume Estimation of Software System in Object Oriented and Structural Model of Academic System,” *Int. J. Comput. Appl.*, vol. 70, no.

- 10, pp. 1–4, 2013.
- [11] I. M. S. Raharja and A. Sn, “Perbandingan Proses Pengembangan Perangkat Lunak Model Spiral Dan Cleanroom,” *Semin. Nas. Inform. 2015 (semnasIF 2015)*, vol. 2012, no. semnasIF, p. hh. 103-109, 2012.
- [12] S. Yunis, “Arsitektur Bisnis : Pemodelan Proses Bisnis,” *Seminar*, vol. 2010, no. semnasIF, pp. 167–173, 2010.
- [13] Hamzah, R. Saptono, and A. Rini, “Development of Software Size Estimation Application using Function Point Analysis ( FPA ) Approach with Rapid Application ... Development of Software Size Estimation Application using Function Point Analysis ( FPA ) Approach with Rapid Application Develop,” vol. 5, no. December, pp. 96–103, 2016.
- [14] D. Khairani, “Studi Kasus Pengukuran Sistem Informasi Menggunakan Function Point ( Fp ),” vol. 8, no. 2, pp. 1–7, 2015.
- [15] F. I. AMALIA, “Pembuatan Kertas Kerja Perkiraan Sendiri ( HPS ) Perangkat Menggunakan Metode Cost Model ( COCOMO ) II,” 2017.
- [16] ROCHMA PERMATASARI, *Analisis Perbandingan Estimasi Effort Proyek Perangkat Lunak Menggunakan Metode Function Point Dengan Masukan Dokumen Use Case, Data Flow Diagram dan Natural Language*. 2016.
- [17] M. Bramwell A. Kasaedja, Rizal Sengkey, ST, MT, Oktavian A. Lantang, ST, “Rancang Bangun Web Service Perpustakaan Universitas Sam Ratulangi,” *Ranc. Bangun Web Serv. Perpust. Univ. Sam Ratulangi*, p. 13, 2014.
- [18] C. YUNITA, *Rancang Bangun Website E-Commerce Kustomisasi Produk Handicraft ( Studi Kasus : UMKM NENA NAMO )*. 2017.
- [19] S. Karouw, “Analisa dan Perancangan Sistem Informasi dengan Pendekatan Agile menurut Panduan PAUS,” *J. Cybermatika*, vol. 1, no. 1, 2013.
- [20] A. S. Sopwan and V. Sofica, “Jurnal Sistem Informasi

- Antar Bangsa.”
- [21] E. Quigley and M. Gargenta, *PHP and MySQL by Example*. 2006.
  - [22] R. C. Martin, *UML for Java programmers*. 2003.
  - [23] S. Dharwiyanti and R. S. Wahono, “Pengantar Unified Modeling Language (UML),” *IlmuKomputer.com*, pp. 1–13, 2003.
  - [24] D. Rahmadiansyah and D. Irwan, “Implementasi Metode Model View Controller Menggunakan Framework Code Igniter dalam Pengembangan Aplikasi Manajemen Depo Petikemas pada Unit Usaha Belawan Logistics Center,” *Snastikom*, no. Snastikom, pp. 1–11, 2012.
  - [25] Y. Kurniawan, Sholiq, and F. A. Muqtadiroh, “Rancang Bangun Perangkat Lunak untuk Workflow Pengelolaan Surat Menyurat Dinas Bagian Surat Mask di Kabupaten Buton Utara,” *J. Tek. ITS*, vol. 1, no. August 2015, pp. 272–277, 2012.
  - [26] S. T. Romeo, *Testing dan Implementasi*. .
  - [27] D. Mustaqbal, “Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN),” *J. Ilm. Teknol. Inf. Terap.*, vol. 1, no. 3, pp. 31–36, 2015.
  - [28] M. B. Sharma, M. K. Choudhary, and R. Purohit, “Function Point Analysis : Converting various design elements into Function Points,” vol. 5, no. 7, pp. 1–7, 2017.
  - [29] GSC, “General System haracteristics.”
  - [30] B. Boehm, S. Devnani-Chulani, B. Clark, and B. Steece, “Calibration approach and results of the COCOMO II post–architecture model,” pp. 1–8, 1997.
  - [31] W. Pradani, “Kajian Metode Perhitungan Metrik Function-Point dan Penerapannya pada Dua Perangkat Lunak yang Dipilih,” *Al-Azhar Indones. Seri Sains Dan Teknol.*, vol. 2, no. 1, pp. 28–34, 2013.
  - [32] A. Widodo and B. Prasetyo, “Perancangan Sistem Estimasi Biaya Proyek Pengembangan,” vol. 2006, no.

- Snati, 2006.
- [33] P. Yhurinda, Anggi Putri, “Modifikasi Metode Function Point Dengan Menambahkan Kompleksitas Proses Bisnis Pada General System Characteristics Untuk Estimasi Biaya Perangkat Lunak,” 2018.
- [34] P. Rana, “A Study of Component Based Complexity Metrics,” vol. 9359, no. 11, pp. 159–165, 2014.
- [35] S. Grimstad, “Software Effort Estimation Error,” no. November, p. 145, 2006.
- [36] Sholiq, R. S. Dewi, and A. P. Subriadi, “A Comparative Study of Software Development Size Estimation Method: UCPabc vs Function Points,” *Procedia Comput. Sci.*, vol. 124, pp. 470–477, 2017.
- [37] G. Karner, “Resource Estimation For Objectory Projects,” *Object. Syst. SF AB*, pp. 1–9, 1993.
- [38] A. P. Subriadi, F. A. Muqtadiroh, and R. S. Dewi, “A Model of Owner Estimate Cost for Software Development Project in Indonesia,” no. 1, pp. 1–13.
- [39] M. Ochodek, J. Nawrocki, and K. Kwarciaak, “Simplifying effort estimation based on Use Case Points,” *Inf. Softw. Technol.*, vol. 53, no. 3, pp. 200–213, 2011.
- [40] A. P. Subriadi, Sholiq, and P. A. Ningrum, “Critical review of the effort rate value in use case point method for estimating software development effort,” *J. Theoretical Appl. Inf. Technol.*, vol. 59, no. 3, pp. 735–744, 2014.
- [41] K. Saleh, “Effort and Cost Allocation in Medium to Large Software Development Projects,” *Intenational J. Comput.*, vol. 5, no. 1, pp. 74–79, 2011.
- [42] K. S. Australia, “Salary Guide,” no. February, 2017.
- [43] I. N. K. I. INKINDO, “Pedoman Standar Minimal Tahun 2018 untuk Kegiatan Jasa Konsultansi,” 2018.

## BIODATA PENULIS



Penulis bernama lengkap Fandhi Akhmad, dilahirkan di kota Pekon Baru, 12 Agustus 1996, merupakan anak kedua dari dua bersaudara. Penulis telah menempuh pendidikan formal di SDN 8 Jatiasih Bekasi, SMPN 9 Bekasi, dan SMAN 3 Bekasi. Penulis meneruskan pendidikan tinggi negeri di Departemen Sistem Informasi Institut Teknologi Sepuluh Nopember, Surabaya dan terdaftar dengan NRP 05211440000171. Pengalaman selama menjadi mahasiswa di ITS, penulis aktif berorganisasi di Himpunan Mahasiswa Sistem Informasi sebagai Wakil Ketua Himpunan. Penulis juga pernah melaksanakan kerja praktik selama dua bulan pada tahun 2017 di PT. Pertamina Gas, Jakarta.

Penulis mengambil bidang minat Manajemen Sistem Informasi (MSI) pada penyelesaian Penelitian Tugas Akhir dengan topik Rancang Bangun Aplikasi *Function Point* Untuk Estimasi Harga Perangkat Lunak. Untuk menghubungi penulis, dapat melalui e-mail : [fandiakhmad1996@gmail.com](mailto:fandiakhmad1996@gmail.com).

*Halaman ini sengaja dikosongkan*

## LAMPIRAN A

### A.1 Requirement Analysis

- Kertas Kerja Function Point

**Tabel A. 1 Tabel Perhitungan Crude Function Point**

Item	Item Description	Complexity	Count	Weight	Weighted Count
1	Number of User Inputs	Simple		3	
		Average		4	
		Complex		6	
2	Number of User Outputs	Simple		4	
		Average		5	
		Complex		7	
3	Number of User Inquiries	Simple		3	
		Average		4	
		Complex		6	
4	Number of Files	Simple		7	
		Average		10	
		Complex		15	
5	Number of External Interfaces	Simple		5	
		Average		7	
		Complex		10	
<b>Total Weighted Function Count (FC)</b>					<b>0</b>

**Tabel A. 2 Tabel Perhitungan Relative Complexity Adjustment Factor**

Factor	Description	Rating 0=Irrelevant 5=Essential
F1	Reliability and backup recovery	
F2	Data communications	
F3	Distributed processing	
F4	Performance	
F5	Operate on existing system	
F6	On-line data entry	

F7	Data entry over multiple screens	
F8	Master files updated on-line	
F9	Complex inputs, outputs, files & inquiries	
F10	Complex internal processing	
F11	Code needs to be reusable	
F12	Need conversion and installation	
F13	Multiple installations of the system	
F14	Easy to change and use	
<b>Complexity Factor (CF) = sum of ratings</b>		<b>0</b>

Tabel A. 3 Tabel Formulasi Function Point

<b>Function Points (FP) = FC x (0.65 + 0.01 x CF)</b>	<b>0</b>
---	----------

## A.2 Use Case Description

Tabel A. 4 UC Description Login

Nama Usecase	UCD.01.01 Login		
Aktor	Project Manager, Analis dan Sekretaris		
Deskripsi	<i>Use case</i> ini dimaksudkan aktor melakukan <i>login</i> sesuai dengan sistem		
Precondition	Pengguna berada pada halaman <i>login</i> aplikasi		
Pemicu	Pengguna ingin melakukan <i>login</i> kedalam aplikasi		
Urutan Kejadian	No	Aktor	Sistem
	1.	Pengguna masuk ke halaman <i>login</i>	Sistem menampilkan halaman <i>login</i>
	2.	Pengguna memasukkan	Sistem menampilkan

		<i>username</i> dan <i>password</i>	<i>username</i> dan <i>password</i> yang dimasukkan oleh pengguna
	3.	Pengguna menekan tombol <i>login</i>	Sistem menampilkan halaman <i>homepage</i> dengan tulisan selamat datang pengguna
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika pengguna telah berada di halaman <i>homepage</i>		
Kondisi Setelah	Pengguna bisa melakukan estimasi		
Aturan Proses	Pengguna harus mengisi kolom <i>username</i> dan <i>password</i> yang terdaftar ke dalam sistem.		
Asumsi	<ul style="list-style-type: none"> <li>• Pengguna ingin melakukan <i>login</i> ke dalam aplikasi</li> <li>• Pengguna ingin melakukan estimasi</li> </ul>		

Tabel A. 5 UC Description Logout

Nama Usecase	UCD.01.02 Logout		
Aktor	Project Manager, Analis dan Sekretaris		
Deskripsi	Use case ini dimaksudkan agar aktor bisa keluar dari aplikasi.		
Precondition	Pegguna telah <i>login</i> ke dalam aplikasi		
Pemicu	Pegguna ingin keluar dari aplikasi		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Aktor menekan tombol user di pojok kanan atas	Sistem menampilkan <i>dropdown</i> pilihan edit profile dan logout
	2.	Aktor menekan tombol logout	Sistem menampilkan halaman login aplikasi
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika pengguna sudah berada di halaman <i>login</i> aplikasi		
Kondisi Setelah	Pegguna keluar dari aplikasi dan bisa melakukan <i>login</i> kembali		
Aturan Proses	Pegguna harus <i>login</i> terlebih dahulu		
Asumsi	<ul style="list-style-type: none"> <li>• Pengguna ingin keluar dari aplikasi</li> <li>• Pengguna sudah melakukan estimasi</li> </ul>		

Tabel A. 6 UC Description Membuka Halaman Awal

Nama Usecase	UCD.01.03 Membuka Halaman Awal		
Aktor	Project Manager, Analis dan Sekretaris		
Deskripsi	Use case ini dimaksudkan agar aktor masuk ke dalam tampilan awal aplikasi		
Precondition	Pengguna telah melakukan <i>login</i> kedalam aplikasi		
Pemicu	Pengguna ingin menggunakan aplikasi <i>function point</i>		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Aktor menekan tombol homepage pada daftar menu	Sistem menampilkan halaman homepage yang berisi penjelasan aplikasi dan ucapan selamat datang pengguna
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika pengguna dapat mengakses aplikasi <i>function point</i>		
Kondisi Setelah	Pengguna dapat masuk halaman sesuai peran masing-masing		
Aturan Proses	-		
Asumsi	Pengguna ingin masuk dan menjalankan aplikasi sesuai peran masing-masing		

Tabel A. 7 UC Description Memasukkan Fitur Aplikasi

Nama Usecase	UCD.01.06 Memasukkan Fitur Aplikasi		
Aktor	Project Manager, Analis dan Sekretaris		
Deskripsi	Use case ini dimaksudkan agar aktor bisa memasukkan fitur apa saja yang ada pada aplikasi yang akan di estimasi		
Precondition	Pegguna telah memasukkan deskripsi aplikasi		
Pemicu	Pegguna ingin melakukan estimasi		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Aktor mengisikan kolom nama fitur	Sistem menampilkan data yang diisikan oleh pengguna
	2.	Aktor menekan tombol tambah	Sistem menampilkan fitur yang telah di masukkan ke dalam tabel yang sudah ada
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika pengguna dapat memasukkan fitur aplikasi		
Kondisi Setelah	Pegguna dapat melihat daftar fitur aplikasi		
Aturan Proses	Pegguna telah memasukkan deskripsi aplikasi		
Asumsi	Pegguna ingin melakukan estimasi		

**Tabel A. 8 UC Description Melihat Daftar Fitur Aplikasi**

Nama Usecase	<b>UCD.01.07 Melihat Daftar Fitur Aplikasi</b>		
Aktor	Project Manager, Analis dan Sekretaris		
Deskripsi	Use case ini dimaksudkan agar aktor dapat melihat daftar fitur yang ada pada aplikasi yang akan di estimasi		
Precondition	Pegguna telah memasukkan fitur aplikasi yang di estimasi		
Pemicu	Pegguna ingin melakukan estimasi		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Aktor menekan tab deskripsi aplikasi	Sistem menampilkan daftar fitur dalam bentuk tabel
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika pengguna dapat melihat daftar fitur aplikasi		
Kondisi Setelah	<ul style="list-style-type: none"> <li>• Pengguna dapat menambah atau menghapus fitur aplikasi</li> <li>• Pengguna dapat masuk ke dalam halaman penilaian CFP</li> </ul>		
Aturan Proses	Pegguna berada di halaman form aplikasi		
Asumsi	Pegguna ingin memasukan informasi aplikasi		

**Tabel A. 9 UC Description Mengubah Biaya Tiap Aktivitas**

Nama Usecase	<b>UCD.01.12 Mengubah Biaya Tiap Aktivitas</b>
Aktor	Project Manager, Analis dan Sekretaris

Deskripsi	Use case ini dimaksudkan agar aktor dapat mengubah biaya tiap aktivitas		
Precondition	Pengguna telah melakukan estimasi dan sudah mendapatkan biaya aktivitas		
Pemicu	Pengguna ingin merubah biaya yang telah di estimasi		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Aktor menekan tombol edit	Sistem membuat kolom biaya menjadi editable
	2.	Aktor memasukkan biaya yang ingin dimasukkan	Sistem menampilkan biaya yang dimasukkan oleh pengguna
	3.	Aktor menekan tombol simpan	Sistem menyimpan data terbaru yang diisi oleh pengguna serta menampilkan data yang sudah disikan pengguna
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika tabel aktivitas biaya dapat diubah		
Kondisi Setelah	Pengguna dapat menyimpan biaya aktivitas yang baru		

Aturan Proses	Pengguna berada di halaman estimasi
Asumsi	Hasil estimasi tidak sesuai dengan ekspektasi pengguna

**Tabel A. 10 UC Description Menambah Biaya Operasional**

Nama Usecase	<b>UCD.01.13 Menambah Biaya Operasional</b>		
Aktor	Project Manager, Analis dan Sekretaris		
Deskripsi	Use case ini dimaksudkan agar aktor dapat menambah biaya operasional dari proyek pengembangan perangkat lunak		
Precondition	Pengguna telah melakukan estimasi dan sudah mendapatkan harga perangkat lunak		
Pemicu	Pengguna memiliki biaya di luar aktivitas biaya yang sudah ada di dalam sistem		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Aktor menekan tombol tambah	Sistem menampilkan kolom nama biaya operasional dan biaya operasional
	2.	Aktor mengisi kolom nama biaya dan biaya operasional	Sistem menampilkan data yang diisikan oleh pengguna
	3.	Aktor menekan tombol simpan	Sistem menyimpan data yang diisikan serta menampilkan dalam bentuk tabel yang sudah ada

Langkah Alternatif	-
Kesimpulan	Usecase berhasil jika pengguna berhasil menambahkan biaya operasional
Kondisi Setelah	<ul style="list-style-type: none"> <li>• Pengguna dapat melihat dan menghapus biaya operasional</li> <li>• Pengguna mendapat biaya total keseluruhan yang baru</li> </ul>
Aturan Proses	Pengguna berada di halaman estimasi
Asumsi	Pengguna memiliki biaya di luar biaya aktivitas yang tersedia di sistem

**Tabel A. 11 UC Description Melihat Biaya Operasional**

Nama Usecase	<b>UCD.01.14 Melihat Biaya Operasional</b>		
Aktor	Project Manager, Analis dan Sekretaris		
Deskripsi	Use case ini dimaksudkan agar aktor dapat melihat biaya operasional yang sudah ditambahkan		
Precondition	Pengguna telah menambahkan biaya operasional		
Pemicu	Pengguna ingin mengetahui biaya operasional		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Aktor menekan tombol tab result	Sistem menampilkan halaman hasil estimasi
	2.	Aktor mengarahkan	Sistem menampilkan

		pada tabel biaya operasional	tabel biaya operasional
Langkah Alternatif	-		
Kesimpulan	Use case berhasil jika pengguna melihat biaya operasional		
Kondisi Setelah	Pengguna dapat melihat, memperbaharui dan menghapus biaya operasional		
Aturan Proses	Pengguna berada di halaman hasil estimasi		
Asumsi	Pengguna ingin mengetahui biaya operasional yang di tambahkan		

**Tabel A. 12 UC Description Mengirim Hasil Estimasi**

Nama Usecase	UCD.01.15 Mengirim Hasil Estimasi		
Aktor	Analisis		
Deskripsi	Use case ini dimaksudkan agar aktor analisis bisa mengirimkan hasil estimasi kepada project manager		
Precondition	Analisis sudah melakukan estimasi		
Pemicu	Analisis ingin memvalidasi hasil estimasi		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Analisis menekan tombol tab result	Sistem menampilkan halaman hasil estimasi
	2.	Analisis menekan tombol kirim validasi	Sistem mengirim hasil estimasi menggunakan mailer dan menampilkan pop

			up email telah tersimpan.
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika sistem menampilkan pesan estimasi terkirim		
Kondisi Setelah	<ul style="list-style-type: none"> <li>• <i>Project</i> manager mendapat informasi estimasi aplikasi yang dikerjakan</li> </ul>		
Aturan Proses	<ul style="list-style-type: none"> <li>• Analis berada di halaman estimasi</li> <li>• Hasil estimasi terkirim ke aktor <i>Project Manager</i></li> </ul>		
Asumsi	<ul style="list-style-type: none"> <li>• Analis ingin memvalidasi hasil estimasi yang sudah dilakukan</li> </ul>		

**Tabel A. 13 UC Description Validasi Hasil Estimasi**

Nama Usecase	UCD.01.16 Validasi Hasil Estimasi		
Aktor	<i>Project Manager</i>		
Deskripsi	Use case ini dimaksudkan agar project manager dapat memvalidasi hasil estimasi yang dikirimkan		
Precondition	Analis telah mengirimkan hasil estimasi		
Pemicu	Pengguna mendapat kiriman hasil estimasi		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Project manager menekan tombol edit pada log estimasi	Sistem menampilkan halaman estimasi dari aplikasi yang dipilih
	2.	Project manager	Sistem menampilkan

		menekan tombol validasi hasil estimasi	pop up validasi telah dilakukan lalu kembali ke halaman log estimasi
Langkah Alternatif	-		
Kesimpulan	Use case ini berhasil jika muncul tombol cetak dokumen pada aktor analis		
Kondisi Setelah	Pengguna dapat mencetak dokumen penawaran		
Aturan Proses	Pengguna <i>login</i> sebagai <i>project manager</i>		
Asumsi	Project Manager mendapat kiriman hasil estimasi		

Tabel A. 14 UC Description Mencetak Dokumen Penawaran

Nama Usecase	UCD.01.17 Mencetak Dokumen Penawaran		
Aktor	Sekretaris		
Deskripsi	Use case ini dimaksudkan agar aktor dapat mencetak dokumen penawaran yang telah dibuat		
Precondition	Sekretaris telah menerima dokumen estimasi yang sudah divalidasi		
Pemicu	Pengguna ingin mencetak dokumen		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Sekretaris menekan tombol print	Sistem menampilkan hasil dokumen penawaran

	2.	Sekretaris melakukan cetak dokumen	Sistem mencetak dokumen
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika sistem menampilkan dokumen penawaran yang siap dicetak		
Kondisi Setelah	Sekretaris mendapat dokumen fisik penawaran		
Aturan Proses	Sekretaris berada di halaman log estimasi dan <i>login</i> sebagai sekretaris		
Asumsi	Pengguna ingin mencetak dokumen		

**Tabel A. 15 UC Description Melihat Log Estimasi**

Nama Usecase	UCD.01.18 Melihat Log Estimasi		
Aktor	<i>Project Manager</i> , analis dan sekretaris		
Deskripsi	Use case ini dimaksudkan agar aktor dapat melihat data estimasi yang sudah pernah dilakukan		
Precondition	Pengguna telah melakukan <i>login</i> dan berada di halaman awal aplikasi		
Pemicu	Pengguna ingin mengetahui daftar estimasi yang sudah dilakukan		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Aktor menekan tombol log estimasi di daftar menu	Sistem menampilkan halaman log estimasi dalam bentuk tabel yang berisi estimasi yang

			sudah pernah dilakukan
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika sistem menampilkan daftar estimasi		
Kondisi Setelah	Pengguna dapat memilih daftar estimasi		
Aturan Proses	Pengguna berada di halaman log estimasi		
Asumsi	Pengguna ingin mengetahui daftar estimasi yang sudah dilakukan		

**Tabel A. 16 UC Description Menambah Daftar Pengguna**

Nama Usecase	UCD.01.19 Menambah Daftar Pengguna		
Aktor	<i>Project Manager</i>		
Deskripsi	Use case ini dimaksudkan agar aktor dapat menambah data pengguna		
Precondition	Pengguna telah <i>login</i> sebagai <i>project manager</i>		
Pemicu	Pengguna ingin menambah <i>user</i> yang ada dalam proyek tersebut		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Project manager menekan tombol tambah pengguna pada daftar menu	Sistem menampilkan halaman tambah pengguna
	2.	Project manager mengisi semua kolom yang ada	Sistem menampilkan data yang

			disikan oleh project manager
	3.	Project manager menekan tombol simpan	Sistem menyimpan data pengguna dan menuju ke halaman daftar pengguna
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika sistem menampilkan halaman tambah pengguna		
Kondisi Setelah	Project Manager dapat memasukkan data pengguna		
Aturan Proses	Pengguna <i>login</i> sebagai <i>project manager</i>		
Asumsi	<i>Project manager</i> ingin menambah personel baru di proyek pengembangan aplikasi		

**Tabel A. 17 UC Description Melihat Daftar Pengguna**

Nama Usecase	<b>UCD.01.20 Melihat Daftar Pengguna</b>
Aktor	<i>Project manager</i>
Deskripsi	Use case ini dimaksudkan agar aktor dapat melihat daftar <i>user</i> yang ada pada proyek tersebut
Precondition	Pengguna telah melakukan <i>login</i> dan berada di halaman awal aplikasi
Pemicu	Pengguna ingin melihat daftar <i>user</i> yang ada pada proyek tersebut

Urutan Kejadian	No.	Aktor	Sistem
	1.	Project Manager menekan tombol daftar pengguna pada daftar menu	Sistem menampilkan halaman daftar pengguna dalam bentuk tabel yang berisi daftar pengguna
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika sistem menampilkan daftar pengguna yang ada pada proyek		
Kondisi Setelah	-		
Aturan Proses	Pengguna telah <i>login</i> sebagai <i>project manager</i> dan berada di halaman daftar pengguna		
Asumsi	Pengguna ingin mengetahui daftar <i>user</i> yang ada pada proyek tersebut		

Tabel A. 18 UC Description Menambah Daftar Tim Anggota

Nama Usecase	UCD.01.21 Menambah Daftar Tim Anggota
Aktor	<i>Project Manager</i>
Deskripsi	Use case ini dimaksudkan agar project manager dapat menambah tim di dalam proyeknya
Precondition	Pengguna telah melakukan <i>login</i> dan berada di halaman awal aplikasi
Pemicu	<i>Project manager</i> ingin menambah anggota tim

Urutan Kejadian	No.	Aktor	Sistem
	1.	Project manager menekan tombol tambah anggota pada daftar menu	Sistem menampilkan halaman tambah anggota
	2.	Project manager mengisi semua kolom yang ada	Sistem menampilkan data yang diisikan oleh project manager
	3.	Project manager menekan tombol simpan	Sistem menyimpan data yang diisikan dan menuju ke halaman daftar anggota tim
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika sistem menampilkan halaman tambah daftar tim anggota		
Kondisi Setelah	Pegguna dapat mengisi anggota tim baru		
Aturan Proses	Pegguna telah <i>login</i> sebagai <i>project manager</i>		
Asumsi	Pegguna ingin menambahkan anggota dari tim		

Tabel A. 19 UC Description Melihat Daftar Tim Anggota

Nama Usecase	UCD.01.22 Melihat Daftar Tim Anggota		
Aktor	<i>Project Manager</i>		
Deskripsi	Use case ini dimaksudkan agar <i>project manager</i> dapat melihat daftar anggota yang ada pada proyek		
Precondition	Pengguna telah melakukan <i>login</i> dan berada di halaman awal aplikasi		
Pemicu	<i>Project manager</i> ingin mengetahui daftar anggota tim pada proyek tersebut		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Project manager menekan tombol daftar anggota tim pada daftar menu	Sistem menampilkan halaman daftar anggota dalam bentuk tabel yang berisi daftar anggota tim
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika sistem menampilkan halaman daftar tim anggota		
Kondisi Setelah	Pengguna dapat menghapus anggota tim pengembang		
Aturan Proses	Pengguna telah <i>login</i> sebagai <i>project manager</i>		
Asumsi	Pengguna ingin mengetahui daftar anggota tim yang ada pada proyek		

Tabel A. 20 UC Description Melihat Daftar Aktivitas

Nama Usecase	UCD.01.23 Melihat Daftar Aktivitas		
Aktor	<i>Project Manager</i>		
Deskripsi	Use case ini dimaksudkan agar <i>project manager</i> dapat melihat daftar aktivitas yang ada		
Precondition	Pengguna telah melakukan <i>login</i> dan berada di halaman awal aplikasi		
Pemicu	Pengguna ingin melihat daftar aktivitas		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Project manager menekan tombol daftar aktivitas	Sistem menampilkan halaman daftar aktivitas dalam bentuk tabel yang berisi aktivitas proyek
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika sistem menampilkan daftar aktivitas proyek		
Kondisi Setelah	Pengguna dapat mengubah dan menghapus aktivitas yang ada		
Aturan Proses	Pengguna telah <i>login</i> sebagai <i>project manager</i>		
Asumsi	Pengguna ingin mengetahui daftar aktivitas yang ada pada proyek		

Tabel A. 21 UC Description Menambah Daftar Profesi

Nama Usecase	UCD.01.24 Menambah Daftar Profesi		
Aktor	<i>Project Manager</i>		
Deskripsi	Use case ini dimaksudkan agar <i>project manager</i> dapat menambah data profesi		
Precondition	Pengguna telah <i>login</i> dan berada di halaman awal aplikasi		
Pemicu	Pengguna ingin menambah profesi yang ada di dalam proyek pengembangan aplikasi		
Urutan Kejadian	No.	Aktor	Sistem
	1.	Project manager menekan tombol tambah profesi pada menu	Sistem menampilkan halaman tambah profesi
	2.	Project manager mengisi semua kolom pada form tambah profesi	Sistem menampilkan data yang diisikan oleh project manager
	3.	Project manager menekan tombol simpan	Sistem menyimpan data dan menuju ke halaman daftar profesi
Langkah Alternatif	-		

Kesimpulan	Usecase berhasil sistem menampilkan halaman tambah profesi
Kondisi Setelah	Pengguna dapat menambah profesi
Aturan Proses	Pengguna telah <i>login</i> sebagai <i>project manager</i>
Asumsi	<i>Project manager</i> ingin menambah profesi yang ada dalam proyek

**Tabel A. 22 UC Description Melihat Daftar Profesi**

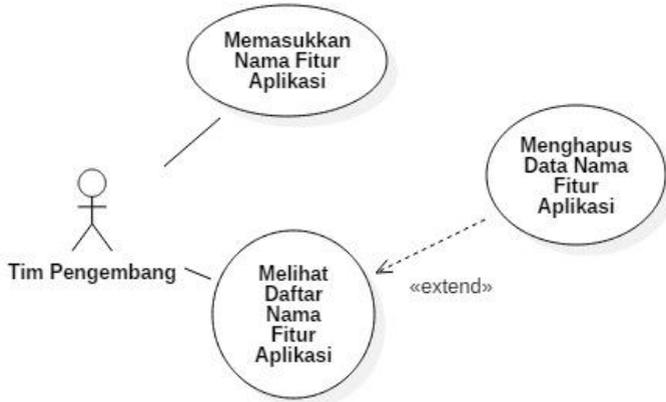
Nama Usecase	UCD.01.25 Melihat Daftar Profesi		
Aktor	<i>Project Manager</i>		
Deskripsi	Use case ini dimaksudkan agar <i>project manager</i> dapat melihat data profesi yang ada di dalam proyek		
Precondition	Pengguna telah melakukan <i>login</i> dan berada di halaman awal aplikasi		
Pemicu	<i>Project manager</i> ingin mengetahui daftar profesi yang ada pada proyek		
Urutan Kejadian	No	Aktor	Sistem
	1.	Project manager menekan tombol daftar profesi pada daftar menu	Sistem menampilkan halaman daftar profesi dalam bentuk tabel yang berisi profesi yang ada
Langkah Alternatif	-		
Kesimpulan	Usecase berhasil jika sistem menampilkan halaman daftar profesi		

Kondisi Setelah	Pengguna dapat mengubah dan menghapus profesi yang ada pada proyeknya
Aturan Proses	Pengguna telah <i>login</i> sebagai <i>project manager</i>
Asumsi	Project manager ingin mengetahui profesi yang ada pada proyeknya

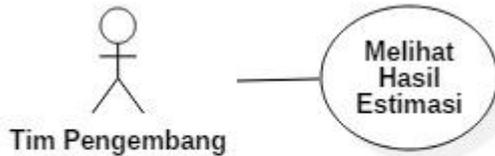
*Halaman ini sengaja dikosongkan*

## LAMPIRAN B

### B.1. Use Case Diagram Aplikasi Estimasi Biaya Perangkat Lunak

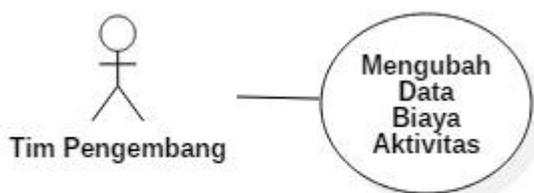


Gambar B. 1 Use case memasukkan nama fitur aplikasi

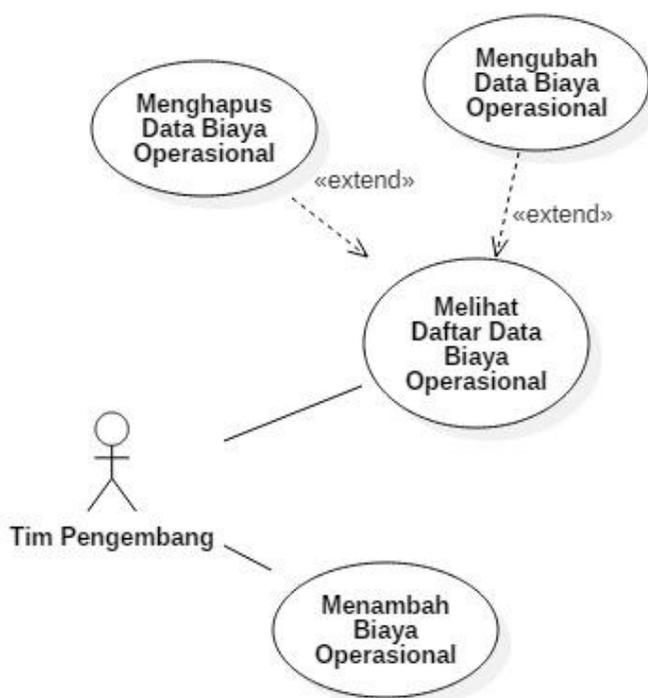


Gambar B. 2 Use case melihat hasil perhitungan estimasi

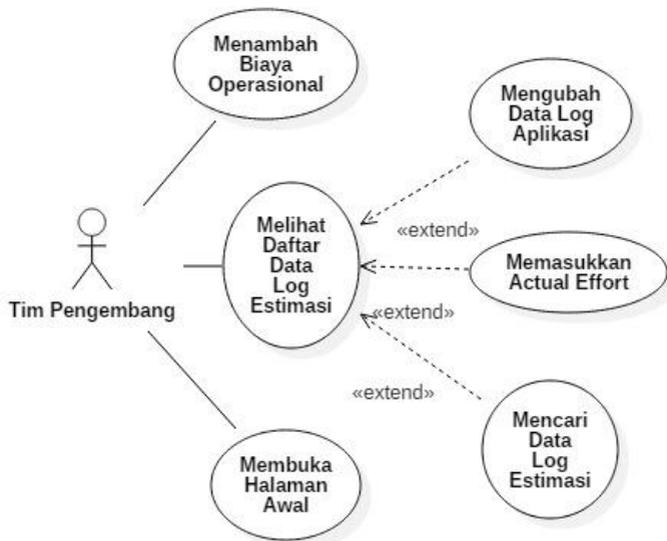
B - 2 -



Gambar B. 3 Use case mengubah biaya aktivitas



Gambar B. 4 Use case biaya operasional

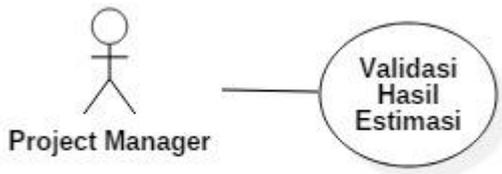


Gambar B. 5 Use case log estimasi aplikasi

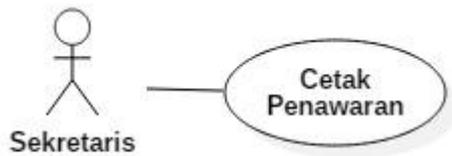


Gambar B. 6 Use case mengirim hasil estimasi

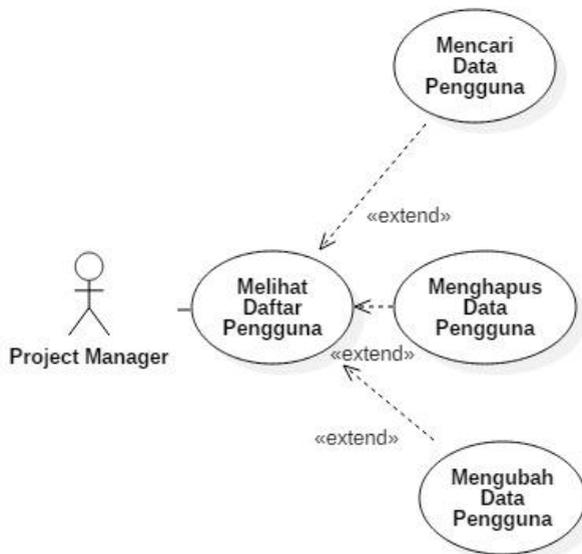
B - 4 -



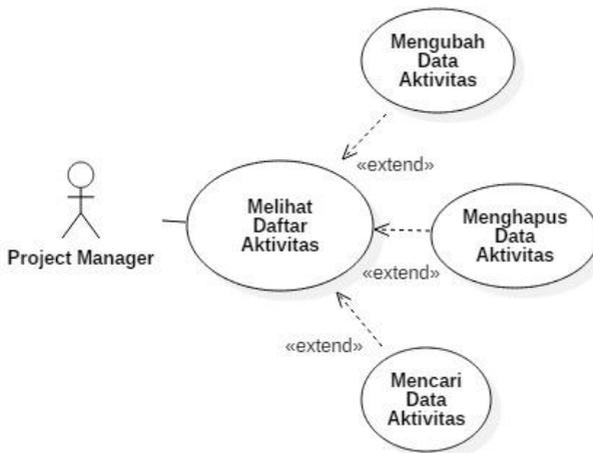
**Gambar B. 7 Use case validasi estimasi**



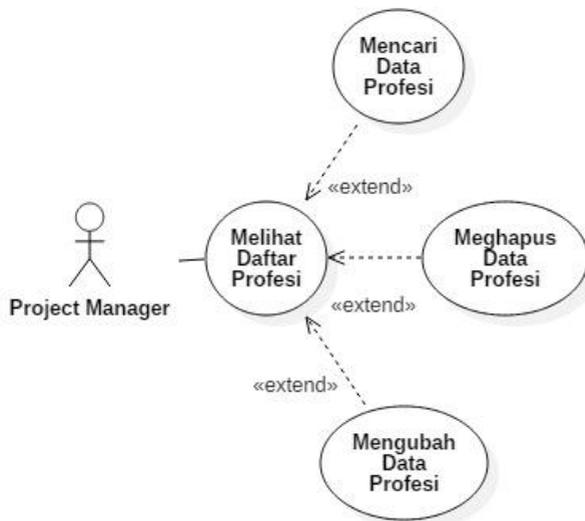
**Gambar B. 8 Use case cetak penawaran**



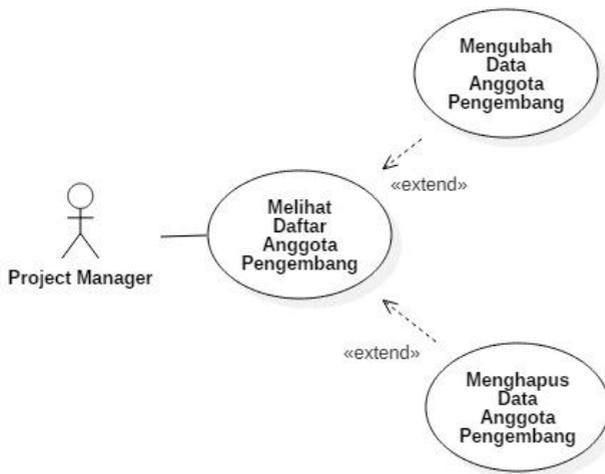
**Gambar B. 9 Use case pengelola pengguna**



**Gambar B. 10 Use case pengelola aktivitas**

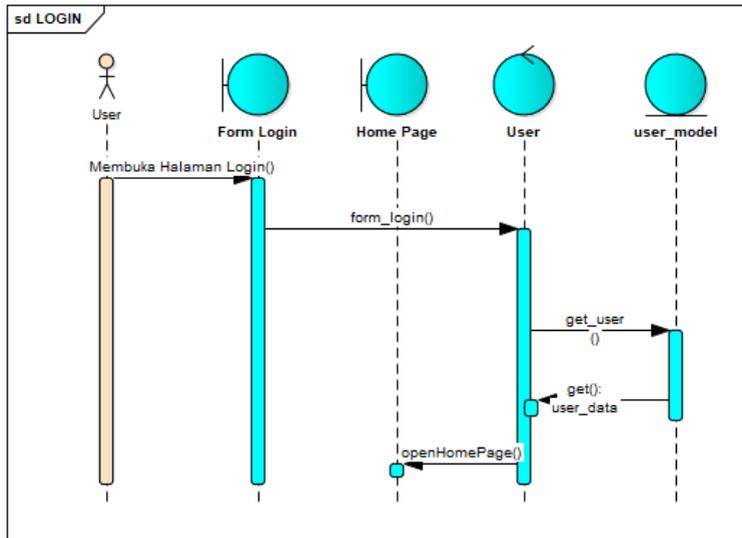


Gambar B. 11 Use case pengelola profesi

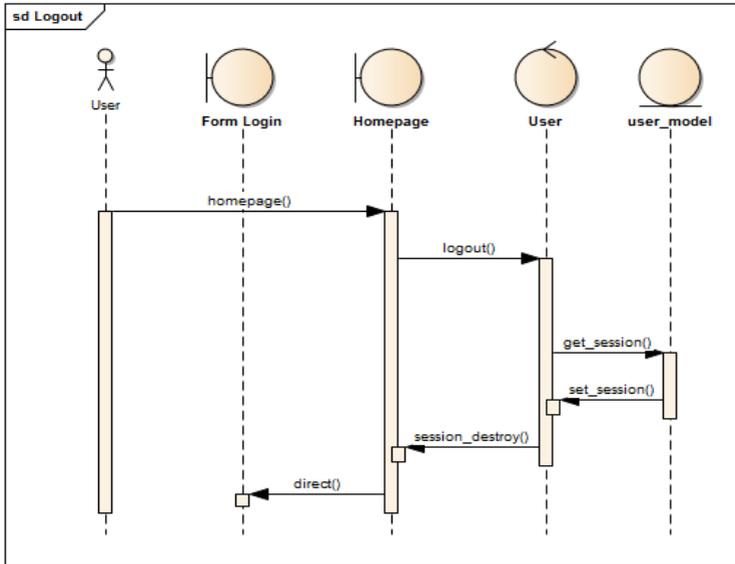


Gambar B. 12 Use case pengelola anggota pengembang

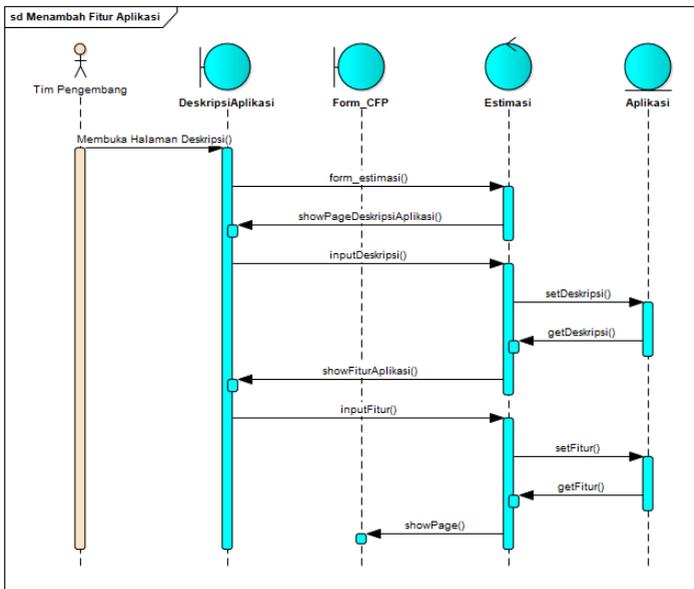
## B.2. Sequence Diagram Aplikasi Estimasi Biaya Perangkat Lunak



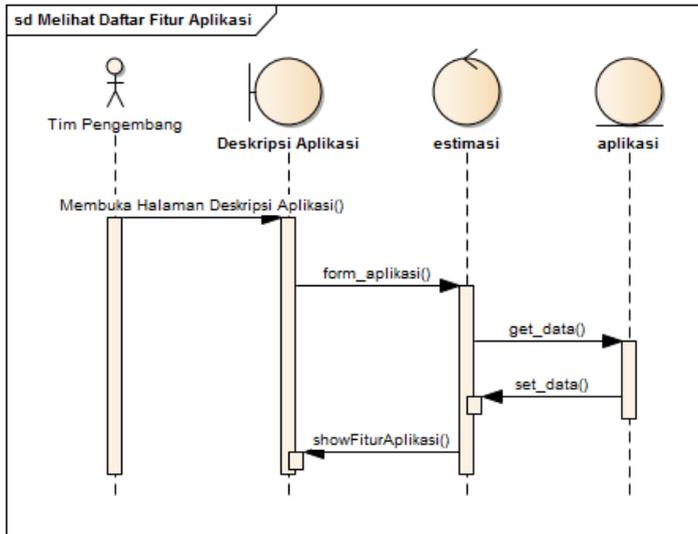
Gambar B. 13 Sequence Login



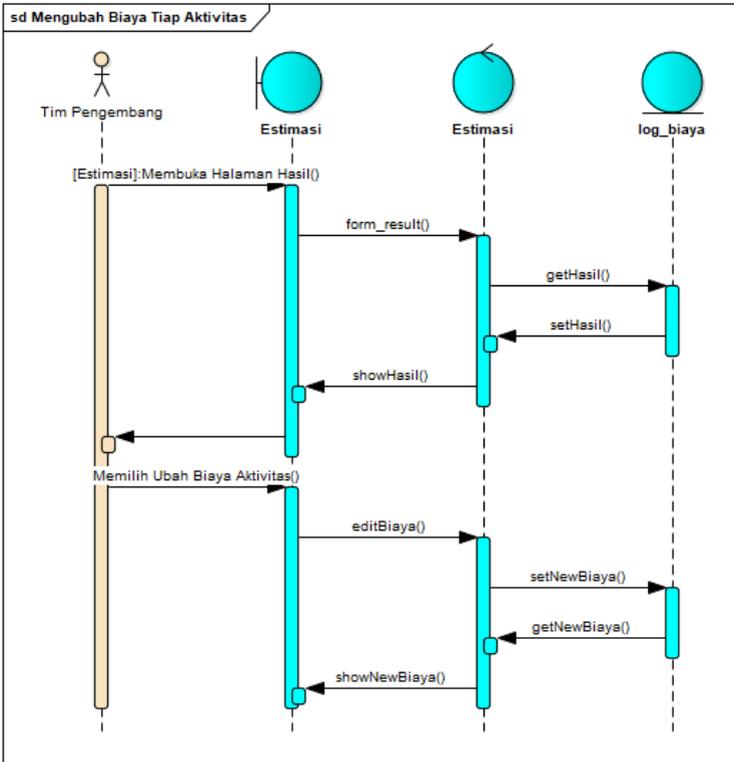
Gambar B. 14 Sequence Logout



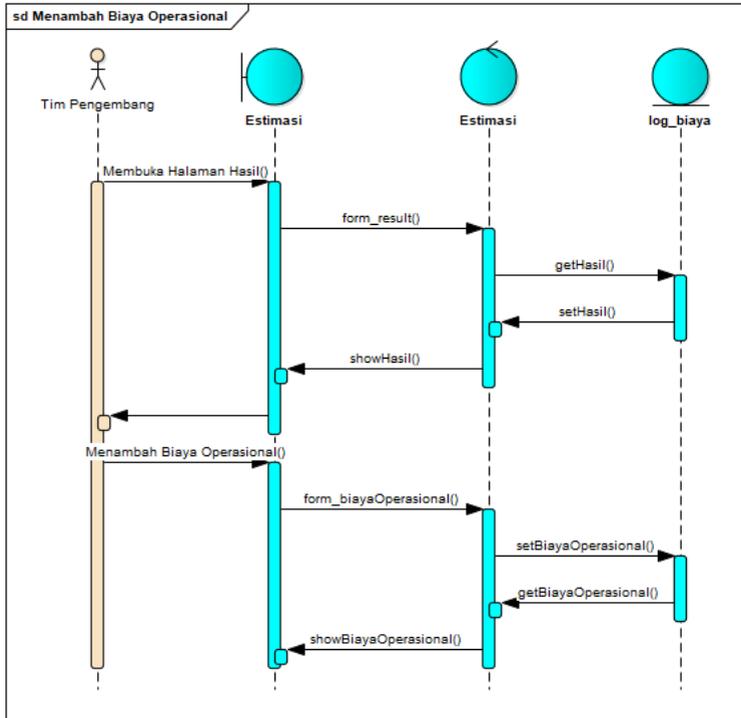
Gambar B. 15 Sequence menambah fitur aplikasi



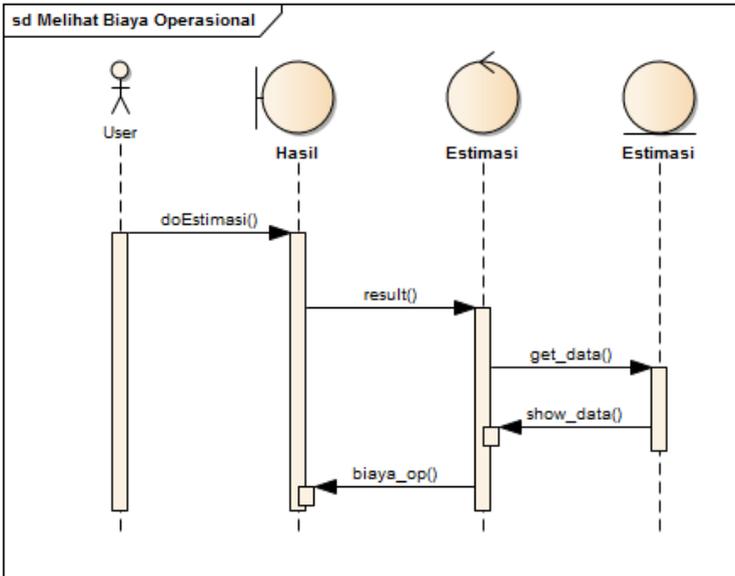
Gambar B. 16 Sequence melihat daftar fitur aplikasi



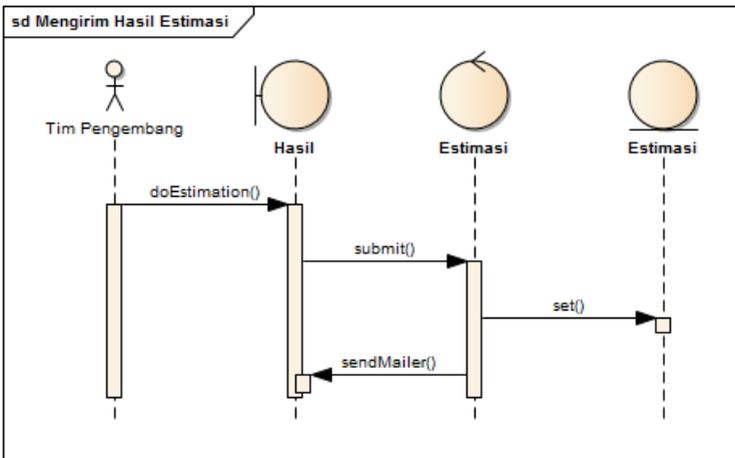
Gambar B. 17 Sequence mengubah biaya tiap aktivitas



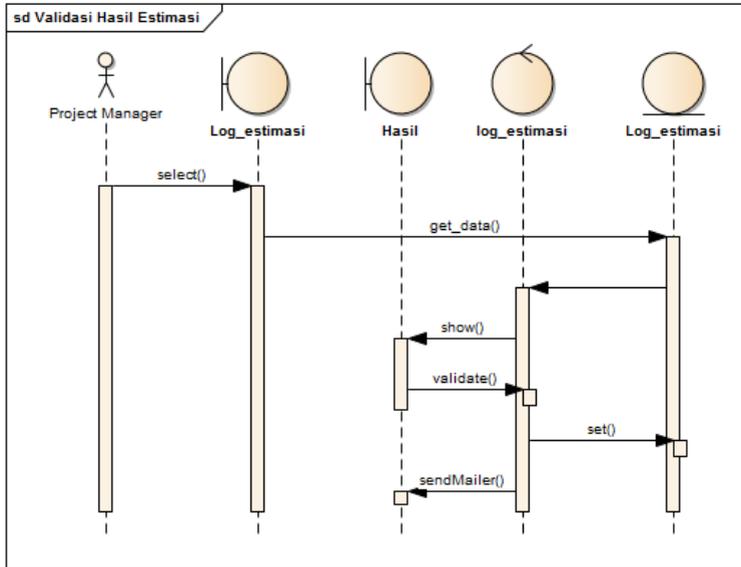
Gambar B. 18 Sequence menambah biaya operasional



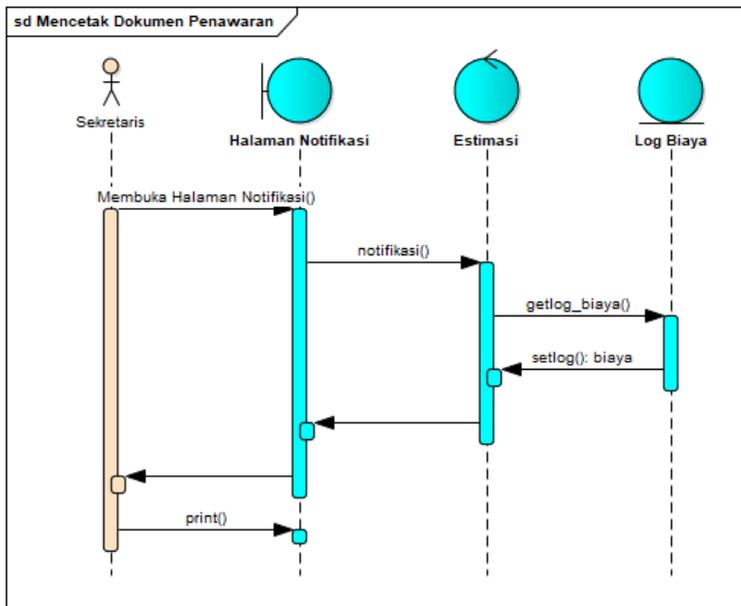
Gambar B. 19 Sequence melihat biaya operasional



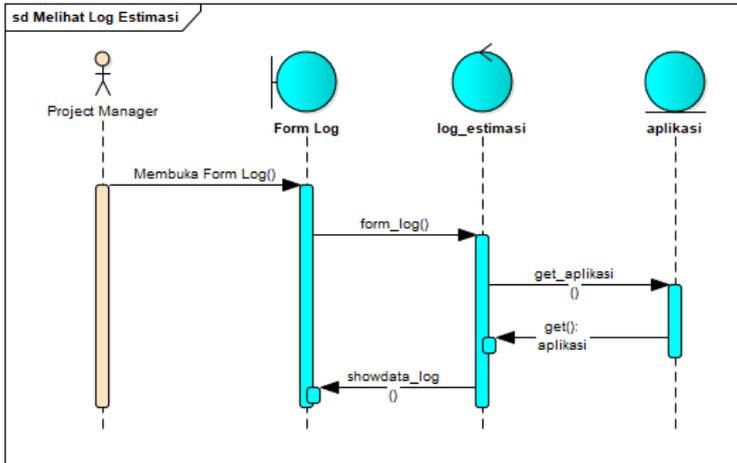
Gambar B. 20 Sequence mengirim hasil estimasi



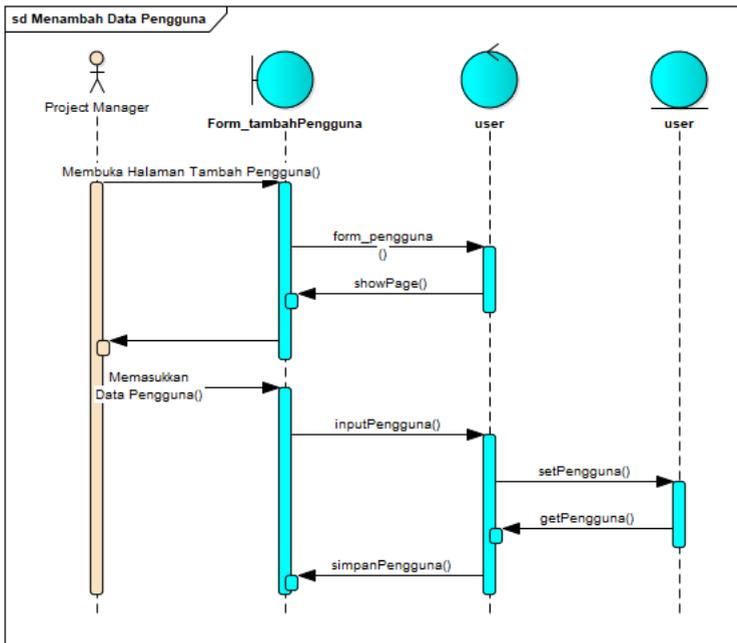
Gambar B. 21 Sequence validasi hasil estimasi



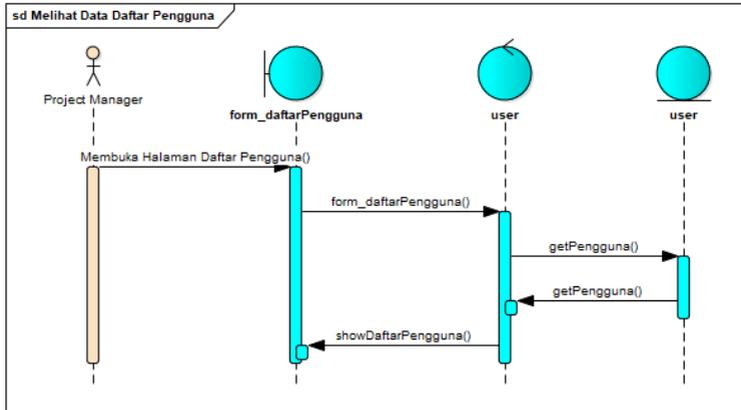
Gambar B. 22 Sequence mencetak dokumen penawaran



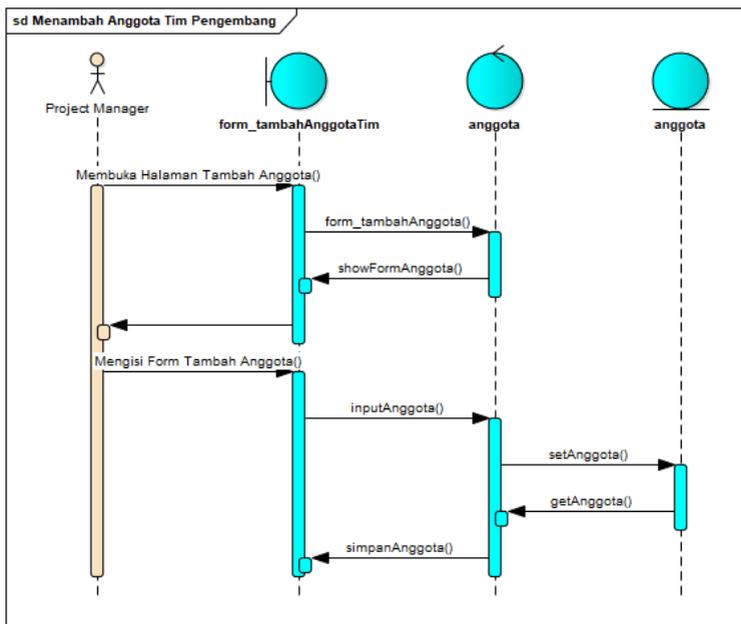
Gambar B. 23 Sequence melihat log estimasi



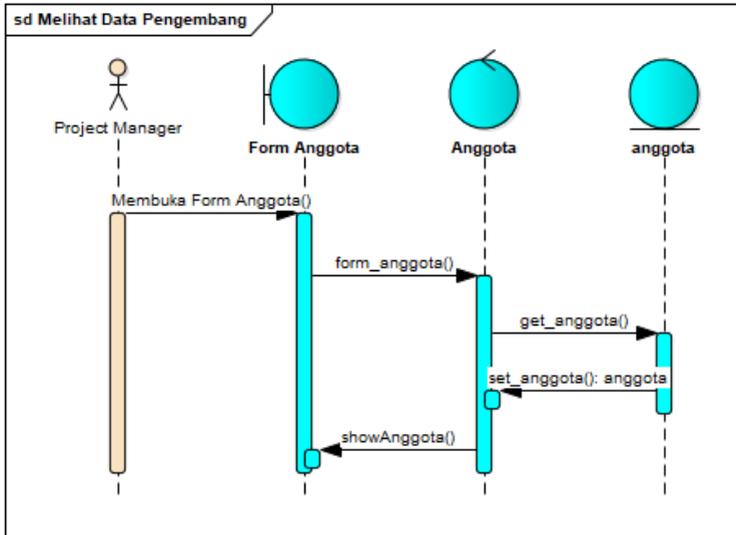
Gambar B. 24 Sequence menambah daftar pengguna



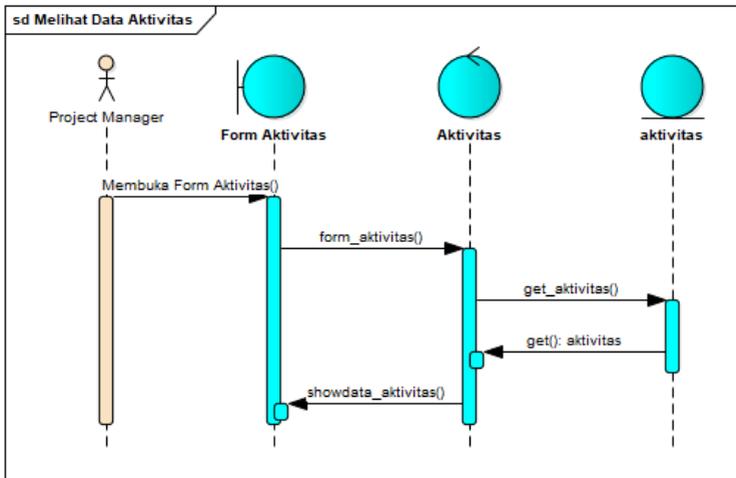
Gambar B. 25 Sequence melihat daftar pengguna



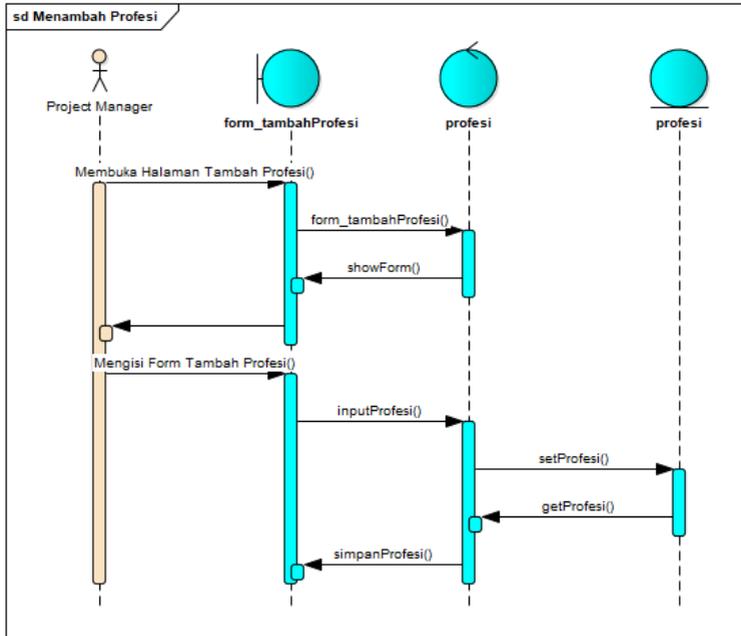
Gambar B. 26 Sequence menambah anggota tim pengembang



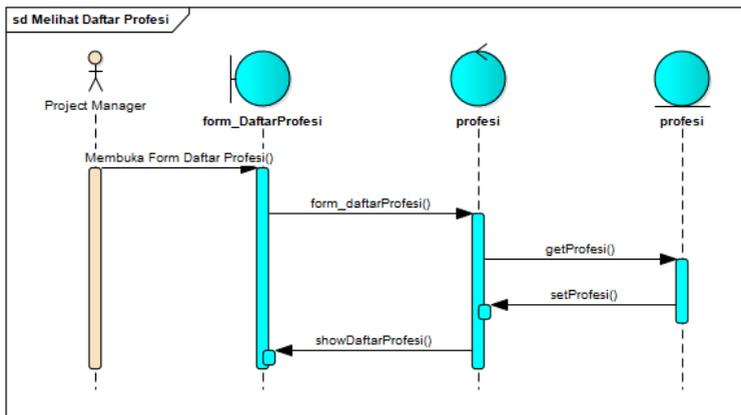
Gambar B. 27 Sequence melihat daftar anggota tim pengembang



Gambar B. 28 Sequence melihat daftar aktivitas

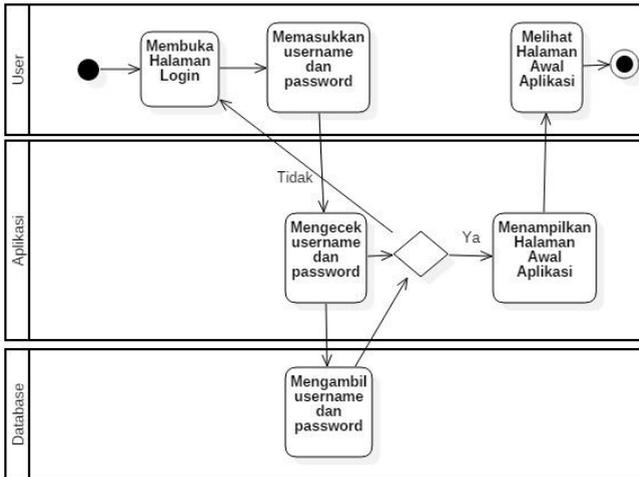


Gambar B. 29 Sequence menambah daftar profesi

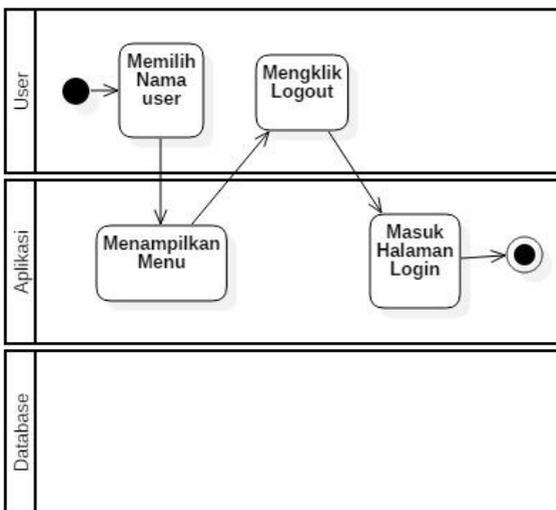


Gambar B. 30 Sequence melihat daftar profesi

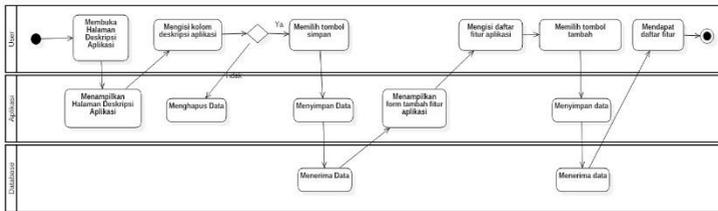
### B.3. Activity Diagram Aplikasi Estimasi Biaya Perangkat Lunak



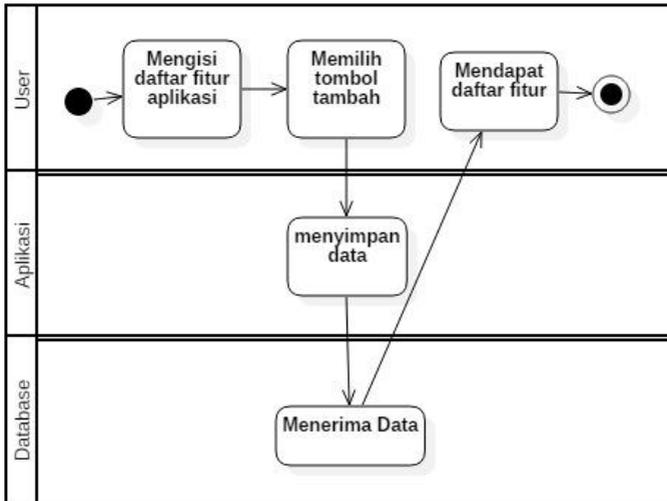
Gambar B. 31 Activity Login



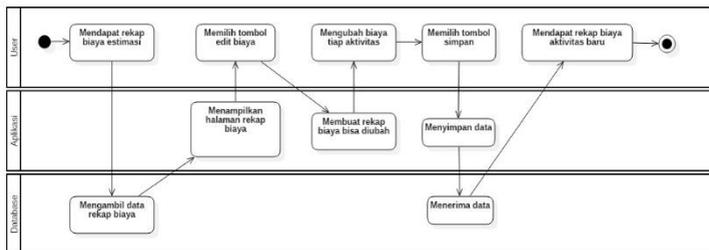
Gambar B. 32 Activity Logout



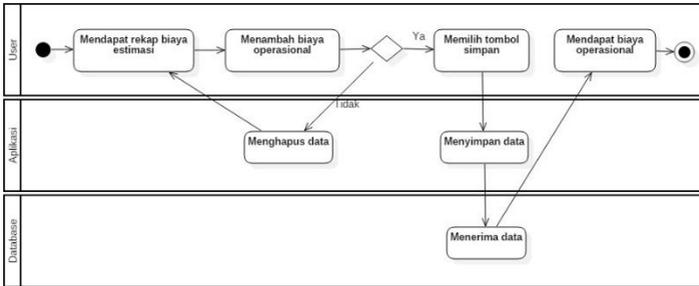
Gambar B. 33 Activity menambah fitur aplikasi



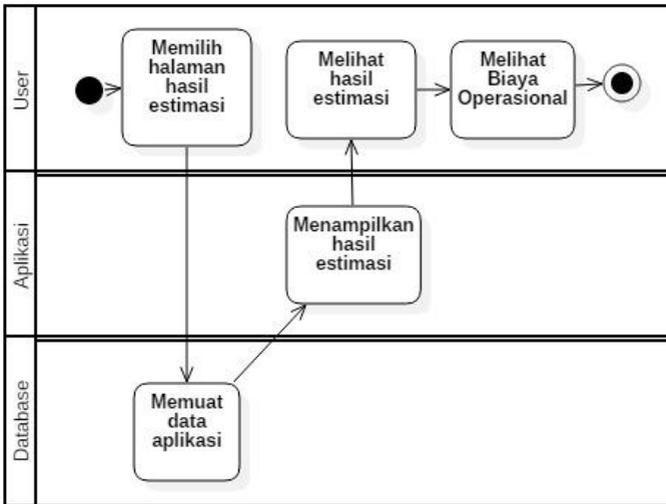
Gambar B. 34 Activity melihat daftar fitur aplikasi



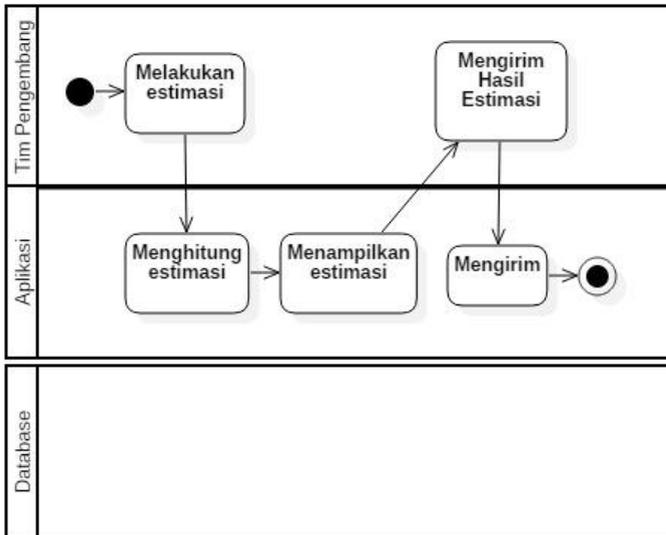
Gambar B. 35 Activity mengubah biaya tiap aktivitas



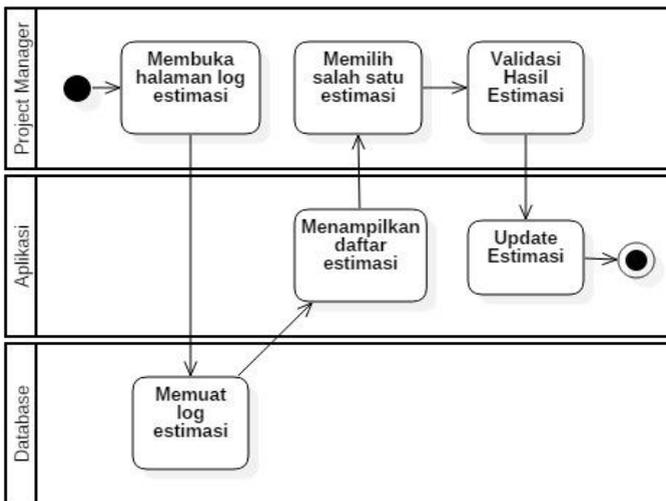
Gambar B. 36 Activity menambah biaya operasional



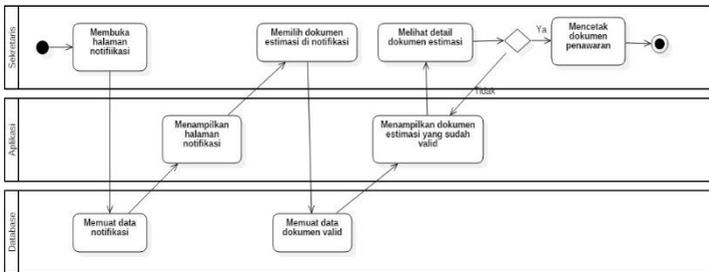
Gambar B. 37 Activity melihat biaya operasional



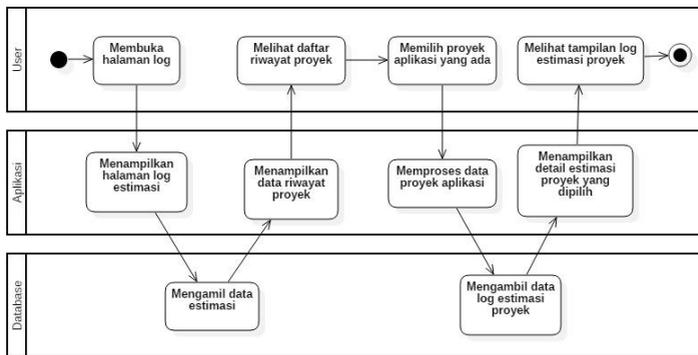
Gambar B. 38 Activity mengirim hasil estimasi



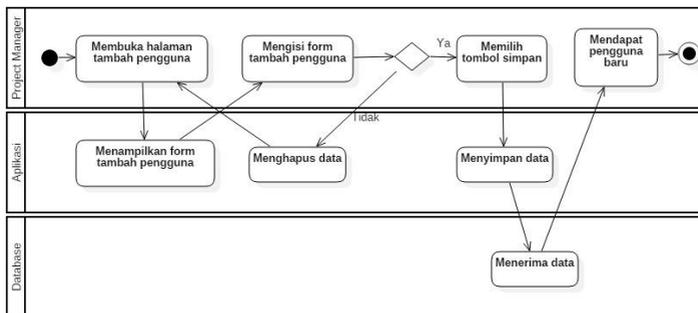
Gambar B. 39 Activity validasi hasil estimasi



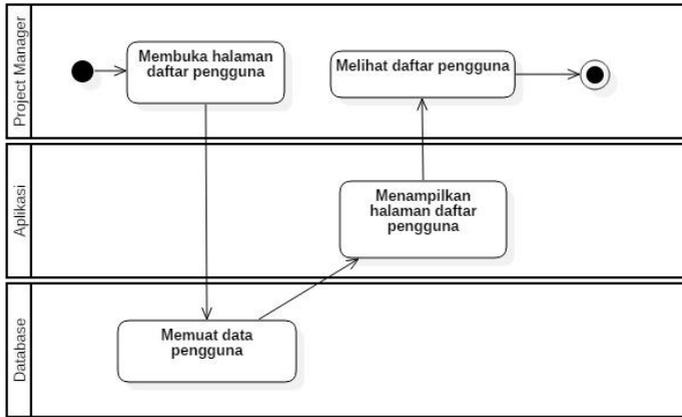
**Gambar B. 40 Activity mencetak dokumen penawaran**



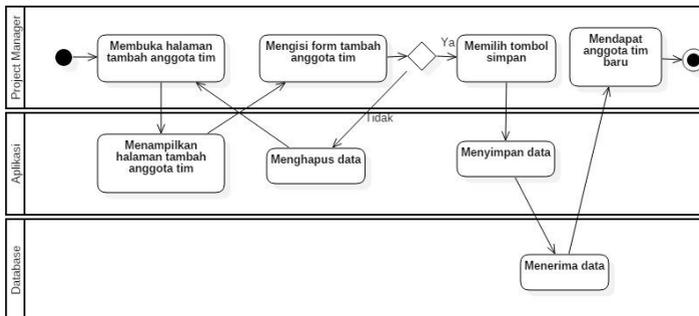
**Gambar B. 41 Activity melihat log estimasi**



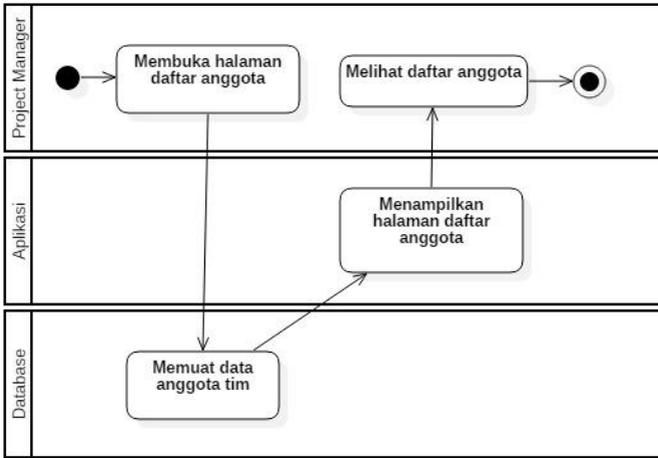
**Gambar B. 42 Activity menambah daftar pengguna**



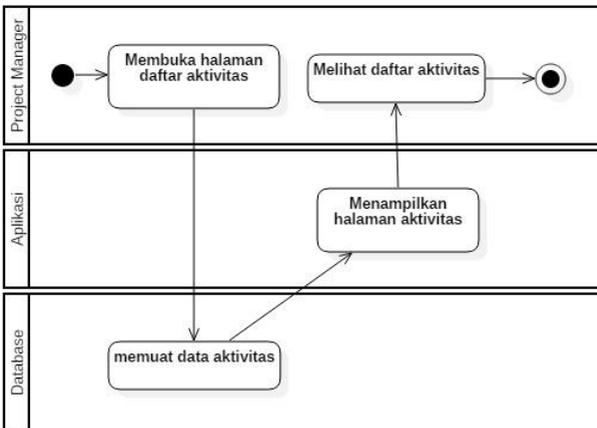
**Gambar B. 43 Activity melihat daftar pengguna**



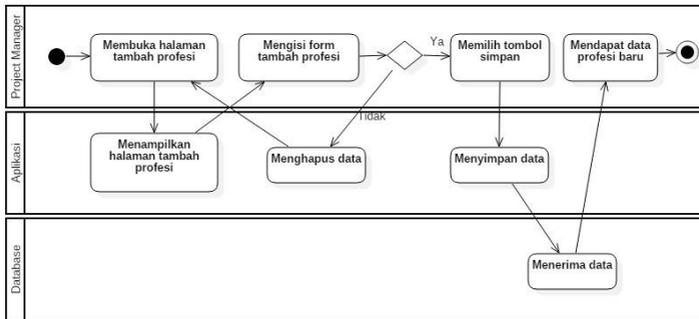
**Gambar B. 44 Activity menambah anggota tim pengembang**



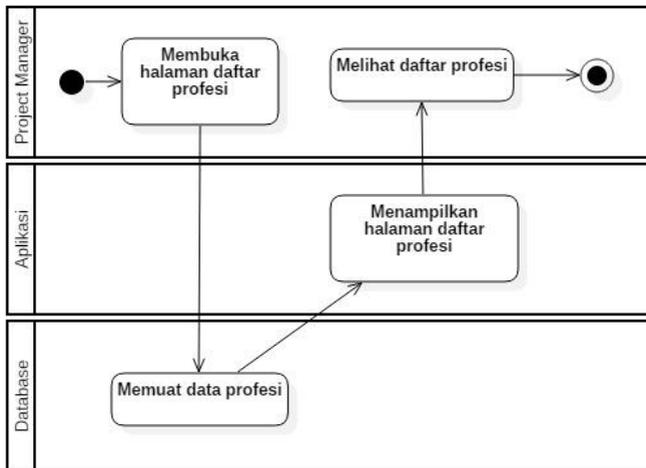
Gambar B. 45 Activity melihat daftar anggota tim pengembang



Gambar B. 46 Activity melihat daftar aktivitas



**Gambar B. 47** Activity menambah daftar profesi



**Gambar B. 48** Activity melihat daftar profesi

*Halaman ini sengaja dikosongkan*

## LAMPIRAN C

### C.1 Hasil Aplikasi Estimasi Biaya Perangkat Lunak

#### Rekap Biaya Tiap Aktivitas

 Edit

Aktivitas	Effort		Biaya		
	%	Hour of Effort	Standard gaji/bulan(Rp)	Gaji per Jam (Rp)	Biaya (Rp)
<b>SOFTWARE DEVELOPMENT</b>					
Requirements	7.5 %	63,17	7.000.000,00	43.750,00	2.763.687,50
Specifications & Design	17.5 %	147,40	7.000.000,00	43.750,00	6.448.750,00
Coding	10 %	84,23	5.000.000,00	31.250,00	2.632.187,50
Testing	7 %	58,96	5.000.000,00	31.250,00	1.842.500,00
Sub Total	42,00%	353,76			13.687.125,00
<b>ON GOING ACTIVITY</b>					
Project management	7 %	58,96	8.500.000,00	53.125,00	3.132.250,00
Configuration Management	3 %	25,27	7.000.000,00	43.750,00	1.105.562,50

Gambar C. 1 Halaman biaya estimasi

## Biaya Operasional

No	Deskripsi	Biaya (Rp)	Actions
Tidak ada data			
<b>TOTAL</b>		<b>0,00</b>	

**+ Tambah**

**Gambar C. 2 Halaman biaya operasional**

 **Form Pengguna** [Tambah Data](#)

**Nama\***

**Username\***

**Email\***

**New Password\***

**Peran\***

Project Manager
  Sistem Analis
  Sekretaris

NB: \* Wajib Diisi

[Kembali](#)
[Simpan](#)

**Gambar C. 3 Halaman tambah pengguna**

**Daftar Pengguna**

10 records per page Search:

No	Nama	Username	Email	Role	Actions
1	Administrator	admin		Admin	No Action
2	Project Manager	projectmanager		Project Manager	<a href="#">Edit</a> <a href="#">Delete</a>
3	System Analyst	analis		Sistem Analis	<a href="#">Edit</a> <a href="#">Delete</a>
4	Secretary	sekretaris		Sekretaris	<a href="#">Edit</a> <a href="#">Delete</a>

**Gambar C. 4 Halaman daftar pengguna**

 **Tambah Data**

**Nama\***

**Profesi\***

**Pengalaman\***

NB: \* Wajib Disi

[← Kembali](#) [Simpan](#)

**Gambar C. 5 Halaman tambah anggota tim**

**Daftar anggota tim**

10 records per page Search:

No	Nama Anggota	Profesi	Pengalaman	Actions
1	Dewangga Prasetya Praja	Programmer	2 tahun	<a href="#">Edit</a>
2	Naufal Raihan Noly	System Analyst	2 tahun	<a href="#">Edit</a>
3	Fandhi Akhmad	Project Manager	2 tahun	<a href="#">Edit</a>
4	Brillianto WS	Software QA	2 tahun	<a href="#">Edit</a>
5	M Iqbal Imaduddin	Secretary	2 Tahun	<a href="#">Edit</a>

**Gambar C. 6 Halaman daftar tim pengembang**

 **Tambah Data**

**Nama profesi\***

**Gaji /bulan\***

NB: \* Wajib Disi

[← Kembali](#) [Simpan](#)

**Gambar C. 7 Halaman tambah profesi**

Daftar profesi			
10	records per page	Search:	<input type="text"/>
No	Nama Profesi	Gaji/bulan	Actions
1	Programmer	Rp. 5.000.000	<a href="#">Edit</a> <a href="#">Delete</a>
2	System Analyst	Rp. 7.000.000	<a href="#">Edit</a> <a href="#">Delete</a>
3	Project Manager	Rp. 8.500.000	<a href="#">Edit</a> <a href="#">Delete</a>
4	Software QA	Rp. 5.000.000	<a href="#">Edit</a> <a href="#">Delete</a>
5	Secretary	Rp. 4.000.000	<a href="#">Edit</a> <a href="#">Delete</a>

**Gambar C. 8 Halaman daftar profesi**

Daftar distribusi aktivitas				
10	records per page	Search:	<input type="text"/>	
No	Nama Aktivitas	Pelaku Aktivitas	Presentase Usaha	Actions
1	Acceptance & Deployment	Project Manager	5%	<a href="#">Ubah Pelaku Aktivitas</a>
2	Acceptance & Deployment	Project Manager	5%	<a href="#">Ubah Pelaku Aktivitas</a>
3	Coding	Programmer	10%	<a href="#">Ubah Pelaku Aktivitas</a>
4	Coding	Programmer	10%	<a href="#">Ubah Pelaku Aktivitas</a>
5	Configuration Management	System Analyst	3%	<a href="#">Ubah Pelaku Aktivitas</a>
6	Configuration Management	System Analyst	3%	<a href="#">Ubah Pelaku Aktivitas</a>

**Gambar C. 9 Halaman daftar aktivitas**

## LAMPIRAN D

### D.1 Penjelasan Penilaian Indikator RCAF

Tabel D. 1 Penjelasan bobot tingkat kompleksitas komunikasi data

Nilai	Deskripsi
0	Aplikasi merupakan kumpulan proses yang diproses dalam satu perangkat yang berdiri sendiri
1	Aplikasi merupakan kumpulan proses yang memiliki entri data ATAU proses <i>printing</i> secara <i>remote</i>
2	Aplikasi merupakan kumpulan proses yang memiliki entri data DAN proses <i>printing</i> secara <i>remote</i>
3	Aplikasi termasuk pada pengumpulan data secara <i>online</i> atau TP ( <i>teleprocessing</i> ) tampilan antarmuka ke kumpulan proses atau <i>query</i> sistem
4	Aplikasi lebih dari sekedar tampilan antarmuka tapi mendukung hanya satu tipe protocol komunikasi TP
5	Aplikasi lebih dari sekedar tampilan antarmuka dan mendukung lebih dari satu tipe protocol komunikasi TP

**Tabel D. 2 Penjelasan bobot tingkat kompleksitas pemrosesan terdistribusi**

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

**Tabel D. 3 Penjelasan bobot tingkat kompleksitas performance**

Nilai	Deskripsi
0	Tidak ada persyaratan kinerja khusus yang dinyatakan oleh <i>user</i>
1	Persyaratan kinerja dan desain dinyatakan dan ditinjau tapi tidak ada tindakan khusus yang diperlukan
2	Waktu respon sangat penting selama waktu sibuk. Tidak ada desain khusus untuk penggunaan CPU yang dibutuhkan. Batas waktu proses adalah untuk hari kerja berikutnya

3	Waktu respon sangat penting setiap jam bisnis. Tidak ada desain khusus untuk penggunaan CPU yang dibutuhkan. Memproses tenggang waktu diperlukan dengan sistem yang terhambat
4	Persyaratan kinerja pengguna cukup ketat untuk membutuhkan tugas analisis kinerja di fase desain
5	Alat analisis kinerja digunakan di tahap desain, pengembangan dan / atau implementasi untuk mendapatkan persyaratan kinerja yang dinyatakan oleh pengguna

**Tabel D. 4 Penjelasan bobot tingkat kompleksitas konfigurasi**

Nilai	Deskripsi
0	Tidak ada batasan operasional baik secara eskplisit maupun implisit yang disertakan
1	Pembatasan operasional memang ada, tapi kurang ketat dibanding sebuah aplikasi yang memiliki khas lain. Tidak ada upaya khusus yang diperlukan untuk memenuhi pembatasan.
2	Beberapa pertimbangan keamanan atau waktu disertakan
3	Kebutuhan prosesor khusus untuk bagian tertentu dari suatu aplikasi
4	Pembatasan operasi yang ditetapkan memerlukan batasan khusus aplikasi di pusat prosesor
5	Adanya kendala khusus pada aplikasi yang terdistribusi pada komponen sistem

**Tabel D. 5 Penjelasan bobot tingkat frekuensi pengguna perangkat lunak**

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

**Tabel D. 6 Penjelasan bobot tingkat frekuensi input data**

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

**Tabel D. 7 Penjelasan bobot tingkat kemudahan penggunaan bagi user**

Faktor	<p>Fungsi online yang diberikan menekankan desain untuk efisiensi pengguna akhir dimana desainnya meliputi :</p> <ul style="list-style-type: none"><li>• Bantuan navigasi (misalnya tombol fungsi, lompatan, menu yang dibuat secara dinamis)</li><li>• Menu</li><li>• Bantuan dan dokumen online</li><li>• Gerakan kursor otomatis</li><li>• Menggulir</li><li>• Pencetakan jarak jauh (melalui transaksi online)</li><li>• Tombol fungsi yang telah ditentukan sebelumnya</li><li>• Sistem batch dikirim dari transaksi online</li><li>• Pemilihan kursor data pada layar</li><li>• Penggunaan video terbalik, sorotan, warna yang menggarisbawahi dan indicator lainnya</li><li>• Dokumentasi pengguna hardcopy transaksi online</li><li>• Antarmuka mouse</li><li>• Pop-up jendela</li><li>• Sedikit layar untuk menyelesaikan fungsi bisnis</li><li>• Dukungan bilingual (dua bahasa)</li></ul>
--------	--

Nilai	Deskripsi
	<ul style="list-style-type: none"> <li>Dukungan multibahasa (lebih dari dua bahasa)</li> </ul>
0	Bukan dari salah satu diatas
1	1 sampai 3 dari diatas
2	4 sampai 5 dari diatas
3	6 atau lebih hal diatas tapi tidak ada persyaratan khusus oleh pengguna yang berkaitan dengan efisiensi
4	6 atau lebih persyaratan diatas dan dinyatakan untuk efisiensi pengguna akhir membutuhkan tugas desain (misalnya memaksimalkan <i>default</i> , penggunaan <i>template</i> )
5	6 atau lebih persyaratan diatas dan dinyatakan untuk efisiensi pengguna akhir yang membutuhkan pemakaian alat khusus dan proses untuk menunjukkan bahwa tujuan telah tercapai

**Tabel D. 8 Penjelasan bobot tingkat frekuensi update data**

Nilai	Deskripsi
0	Tidak ada
1	Pembaruan <i>online</i> menyertakan satu sampai tiga file. Volume pembaruan rendah dan pemulihan mudah dilakukan
2	Pembaruan <i>online</i> menyertakan empat atau lebih file. Volume pembaruan rendah dan pemulihan mudah

3	Pembaruan <i>online</i> file logis internal utama disertakan
4	Perlindungan terhadap data yang hilang sangat penting dan telah dirancang dan diprogram secara khusus dalam sistem
5	Volume tinggi membawa pertimbangan biaya ke dalam proses pemulihan yang otomatis dengan menyertakan intervensi operator

**Tabel D. 9 Penjelasan bobot tingkat kompleksitas processing data**

Faktor	<p>Proses yang kompleks merupakan karakter dari suatu aplikasi. Berikut merupakan komponennya kompleksitas :</p> <ul style="list-style-type: none"> <li>• Sensitive (contoh yaitu proses audit) dan /atau proses aplikasi yang spesifik</li> <li>• Proses logis yang ekstensif</li> <li>• Proses matematika yang ekstensif</li> <li>• Banyak sekali proses yang dihindari sehingga menghasilkan transaksi yang tidak lengkap dan harus diolah kembali</li> <li>• Pengolahan yang kompleks untuk menangani beberapa kemungkinan masukan atau keluaran</li> </ul>
Nilai	Deskripsi
0	Bukan dari salah satu di atas
1	Salah satu di atas
2	Dua di atas

3	Tiga di atas
4	Empat di atas
5	Kelima hal di atas

**Tabel D. 10 Penjelasan bobot tingkat kemungkinan penggunaan kembali**

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

**Tabel D. 11 Penjelasan bobot tingkat kemudahan dalam instalasi**

Faktor	Installation Ease
Nilai	Deskripsi
0	Tidak ada pertimbangan khusus yang dikemukakan oleh pengguna dan tidak perlu persiapan khusus untuk instalasi

1	Tidak ada pertimbangan khusus yang dikemukakan oleh pengguna tapi perlu pengaturan khusus untuk instalasi
2	Persyaratan konversi dan pemasangan dinyatakan oleh pengguna dan panduan konversi dan pemasangan disediakan dan diuji. Dampak konversi pada proyek tidak dianggap penting
3	Persyaratan konversi dan pemasangan dinyatakan oleh pengguna dan panduan konversi dan pemasangan disediakan dan diuji. Dampak konversi pada proyek ini dianggap penting
4	Selain 2 diatas, konversi dan instalasi otomatis disediakan dan diuji
5	Selain 3 diatas, otomatis konersi dan instalasi otomatis disediakan dan diuji

**Tabel D. 12 Penjelasan bobot tingkat kemudahan operasional perangkat lunak**

Faktor	Operational Ease
Nilai	Deskripsi
0	Tidak ada pertimbangan operasional khusus selain yang normal. Prosedur <i>back up</i> dinyatakan oleh pengguna
1 - 4	Satu, beberapa atau semua item berlaku untuk aplikasi. Setiap item ada benarnya kecuali yang dinyatakan lain. Proses <i>start up</i> , <i>back up</i> dan <i>recovery</i> yang efektif tapi membutuhkan intervensi operator

	<p>Proses <i>start up</i>, <i>back up</i> dan <i>recovery</i> yang efektif tapi tidak memerlukan intervensi operator</p> <p>Aplikasi ini meminimalkan kebutuhan akan rekaman</p> <p>Aplikasi ini meminimalkan kebutuhan penanganan kertas</p>
5	<p>Aplikasi dirancang untuk operasi tanpa pengawasan. Operasi tanpa pengawasan berarti tidak adanya intervensi operator dalam pengoperasian sistem selain untuk <i>start up</i> atau <i>shut down</i> aplikasi. Pemulihan kesalahan otomatis adalah fitur dari aplikasi</p>

**Tabel D. 13 Penjelasan bobot tingkat perangkat lunak dibuat untuk multi-organisasi**

Faktor	Multiple Sites
Nilai	Deskripsi
0	Persyaratan pengguna tidak perlu mempertimbangkan kebutuhan lebih dari satu pengguna/situs instalasi
1	Kebutuhan beberapa situs dipertimbangkan dalam desain dan aplikasi dirancang untuk beroperasi hanya dibawah lingkungan perangkat keras dan lunak
2	Kebutuhan beberapa situs dipertimbangkan dalam desain dan aplikasi dirancang untuk beroperasi hanya dibawah yang serupa lingkungan perangkat keras dan /atau perangkat lunak

3	Kebutuhan beberapa situs dipertimbangkan dalam desain dan aplikasi ini dirancang untuk beroperasi di bawah yang berbeda dari perangkat keras dan /atau perangkat lunak
4	Dokumentasi dan rencana dukungan disediakan dan diuji untuk mendukung aplikasi di beberapa situs seperti nomor 1 dan 2
5	Dokumentasi dan rencana dukungan disediakan dan diuji untuk mendukung aplikasi di beberapa situs dan aplikasi seperti nomor 3

**Tabel D. 14 Penjelasan bobot tingkat kompleksitas dalam mengikuti perubahan**

Faktor	<p>Aplikasi ini telah dirancang secara khusus, dikembangkan dan didukung untuk memudahkan perubahan. Karakteristik berikut dapat mengajukan permohonan untuk aplikasi:</p> <ul style="list-style-type: none"> <li>• Permintaan dan fasilitas laporan yang fleksibel yang dapat menangani permintaan sederhana sebagai contoh logika hanya diterapkan pada satu file logis internal</li> <li>• Permintaan dan fasilitas laporan yang fleksible yang dapat menangani permintaan dengan kompleksitas rata-rata sebagai contoh lebih dari satu file logis internal</li> <li>• Permintaan dan fasilitas laporan yang fleksibel yang dapat menangani permintaan yang rumit misalnya pada satu atau lebih file logis internal</li> </ul>
--------	---

	<ul style="list-style-type: none"> <li>• Data kontrol bisnis disimpan dalam table yang dikelola oleh pengguna secara online tapi perubahannya berlaku pada hari kerja berikutnya</li> <li>• Data kontrol bisnis disimpan dalam table yang dikelola oleh pengguna secara online dan perubahan berlaku segera</li> </ul>
Nilai	Deskripsi
0	Bukan dari salah satu di atas
1	Salah satu diatas
2	Dua diatas
3	Tiga diatas
4	Empat diatas
5	Semua hal diatas

## LAMPIRAN E

### E.1 Penjelasan Aktivitas

Tabel E. 1 Penjelasan aktivitas

<b>Fase</b>	<b>Variabel Tahapan</b>	<b>Deskripsi Aktivitas</b>	<b>Jenis Aktivitas</b>
Software Development	Requirement	Aktivitas ini merupakan aktivitas yang dilakukan untuk merumuskan kebutuhan fungsional dan non-fungsional, gambaran antarmuka dan melakukan prioritasi kebutuhan yang harus dikerjakan terlebih dahulu. Keluaran dari aktivitas ini adalah ruang lingkup pada dokumen proyek, spesifikasi kebutuhan perangkat lunak dan dokumentasi perencanaan pengujian sesuai dengan keadaan lapangan	<ul style="list-style-type: none"><li>• Survey</li><li>• Wawancara</li><li>• Rapat kick-off</li></ul>

Fase	Variabel Tahapan	Deskripsi Aktivitas	Jenis Aktivitas
	Spesifications	Keluaran dari proses penggalian kebutuhan yang dikerjakan terlebih dahulu untuk digunakan sebagai bahan dasar dalam pembangunan perangkat lunak	<ul style="list-style-type: none"> <li>• Rapat hasil analisis dengan tim pengembang</li> <li>• Analisis proses bisnis aplikasi berdasarkan hasil analisa kebutuhan</li> </ul>
	Design	Rincian aktivitas ini antara lain merumuskan arsitektur teknologi, arsitektur tingkat tinggi, basis data, antarmuka	<ul style="list-style-type: none"> <li>• Penyusunan dokumen SKPL</li> <li>• Rapat hasil analisis dengan stakeholder</li> <li>• Perbaikan dokumen SKPL</li> </ul>
			<ul style="list-style-type: none"> <li>• Sosialisasi hasil analisis kebutuhan ke internal tim pengembang</li> </ul>

Fase	Variabel Tahapan	Deskripsi Aktivitas	Jenis Aktivitas
		pengguna dan perancangan detail lainnya.	<ul style="list-style-type: none"> <li>• Pembagian SDM dan jobdesk</li> <li>• <i>Benchmark template</i> aplikasi</li> <li>• Pembuatan <i>mockup/user interface</i></li> </ul>
	Coding	Aktivitas ini merupakan aktivitas untuk mengeksekusi dokumen perancangan menjadi sebuah kode-kode perangkat lunak.	<ul style="list-style-type: none"> <li>• Sosialisasi UI ke programmer</li> <li>• Pembelian <i>template</i> dan <i>plugin</i></li> <li>• Eksekusi kode program</li> <li>• Revisi program</li> <li>• Rapat dengan stakeholder internal atau external</li> </ul>

Fase	Variabel Tahapan	Deskripsi Aktivitas	Jenis Aktivitas
			<ul style="list-style-type: none"> <li>• <i>Release</i> aplikasi/sistem versi beta</li> <li>• Pembuatan user guide versi beta</li> </ul>
	Testing	<p>Aktivitas ini dilakukan merupakan aktivitas pengujian (<i>testing</i>) yang bertujuan mengetahui apakah aplikasi yang dibangun sudah terintegrasi dengan gambaran modul, sistem hingga komponen penyusunnya dengan mengetahui kesalahan atau ketidaksesuaian perangkat lunak dengan kebutuhan yang sudah dispesifikasikan. Keluaran dari aktivitas ini berupa perangkat lunak yang sudah teruji dan terintegrasi.</p>	<ul style="list-style-type: none"> <li>• Pembuatan checklist pengujian sistem</li> <li>• Pembuatan User Acceptance Test Plan</li> <li>• Rapat koordinasi dengan stakeholder</li> <li>• Pengujian dan integrasi dengan metode whitebox</li> </ul>

Fase	Variabel Tahapan	Deskripsi Aktivitas	Jenis Aktivitas
			<ul style="list-style-type: none"> <li>• Penyesuaian SKPL</li> <li>• Perbaikan dan melengkapi user guide</li> </ul>
On Going Activity	Project Management	Aktivitas ini dimulai dari <i>kick-off</i> hingga penutupan proyek. Proses yang dilakukan dimulai dari pengelolaan sumberdaya hingga risiko yang mungkin ditimbulkan.	<ul style="list-style-type: none"> <li>• Penentuan target proyek</li> <li>• Rekrutmen tim</li> <li>• Penghitungan gaji/upah anggota tim</li> <li>• Penyusunan dokumen penawaran</li> <li>• Penyusunan dokumen perencanaan proyek</li> <li>• Evaluasi tengah proyek</li> </ul>

Fase	Variabel Tahapan	Deskripsi Aktivitas	Jenis Aktivitas
			<ul style="list-style-type: none"> <li>• Rapat penutupan proyek</li> <li>• Penentuan target proyek</li> </ul>
	Configuration Management	Aktivitas pengelolaan konfigurasi dilakukan secara terus-menerus selama fase pengembangan perangkat lunak berlangsung. Keluaran dari konfigurasi ini berupa revisi, penentuan revisi, pembangunan dan manajemen perubahan.	<ul style="list-style-type: none"> <li>• Pengaturan konfigurasi server internal</li> <li>• Uji coba akses aplikasi ke server</li> <li>• Penyusunan dokumen konfigurasi</li> </ul>
	Documentation	Aktivitas ini merupakan akhir dari seluruh aktivitas yaitu mencatat seluruh aktivitas yang ada, mulai dari dokumen internal, manual pengguna, manual instalasi, dan lain-lain.	<ul style="list-style-type: none"> <li>• Pembuatan user guide aplikasi</li> <li>• Pembuatan video tutorial</li> <li>• Penataan laporan akhir proyek</li> </ul>

Fase	Variabel Tahapan	Deskripsi Aktivitas	Jenis Aktivitas
		Aktivitas ini dilakukan selama proyek berlangsung.	
	Training & Support	Aktivitas ini dilakukan apabila pemrograman membutuhkan bantuan tambahan untuk mengatasi permasalahan tertentu maupun pelatihan terhadap pihak-pihak yang menggunakan aplikasi ( <i>user</i> ).	<ul style="list-style-type: none"> <li>• Pemberian materi pelatihan</li> <li>• Pembelian konsumsi pelatihan</li> </ul>
Quality and Testing	Penjaminan Mutu	Aktivitas ini melakukan pengecekan ulang terhadap sistem yang telah dibangun di setiap fase pengembangan perangkat lunak. Pelaporan mutu dan log disertai dengan adanya metric pengukuran mutu yang diharapkan oleh tim manajemen proyek. Selain itu, aktivitas ini juga menghasilkan rekomendasi selanjutnya yang	<ul style="list-style-type: none"> <li>• Pemeriksaan aplikasi oleh pengguna</li> <li>• Penandatanganan software quality</li> <li>• Maintenance aplikasi secara berkala sesuai kesepakatan</li> </ul>

Fase	Variabel Tahapan	Deskripsi Aktivitas	Jenis Aktivitas
		dapat dilakukan oleh tim proyek.	antara PM dengan client
	Evaluasi dan Pengujian	Aktivias ini termaksud dalam kegiatan verifikasi dan validasi. Pada akhir tahapan pengembangan perangkat lunak, diperlukan persetujuan dari pihak-pihak yang terlibat dalam proyek. Evaluasi dilakukan melalui 2 cara yaitu informal (melalui proses audit) dan formal (melalui tools evaluasi)	<ul style="list-style-type: none"> <li>• Maintenance data</li> <li>• Maintenance hardware internal</li> </ul>