

**TUGAS AKHIR - KS141501**

**RANCANG BANGUN APLIKASI USE CASE POINT  
UNTUK ESTIMASI HARGA PERANGKAT LUNAK  
(STUDI KASUS: PERUSAHAAN DTS)**

**MUKHAMAD FAIZ FANANI  
NRP 5211 100 043**

**Dosen Pembimbing  
Sholih, S.T ,M.Kom, M.SA  
Feby Artwodini Muqtadiroh, S.Kom, MT**

**JURUSAN SISTEM INFORMASI  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015**

**FINAL PROJECT - KS141501**

**DESIGN AND IMPLEMENTATION OF USE CASE POINT  
SOFTWARE FOR ESTIMATING SOFTWARE  
DEVELOPMENT COST (CASE STUDY: DTS  
CORPORATION)**

**MUKHAMAD FAIZ FANANI  
NRP 5211 100 043**

**Supervisors  
Sholiq, S.T ,M.Kom, M.SA  
Feby Artwodini Muqtadiroh, S.Kom, MT**

**INFORMATION SYSTEM DEPARTMENT  
Faculty of Information Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015**

**LEMBAR PERSETUJUAN**

**RANCANG BANGUN APLIKASI USE CASE POINT  
UNTUK ESTIMASI HARGA PERANGKAT LUNAK  
(STUDI KASUS: PERUSAHAAN DTS)**

**TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**MUKHAMAD FAIZ FANANI**

NRP 5211 100 043

Disetujui Tim Penguji : Tanggal Ujian : Juni 2015  
Periode Wisuda : September 2015

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

(Pembimbing I)

**Feby Artwodini Muqtadiroh, S.Kom, MT**

(Pembimbing II)

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

(Penguji I)

**Bekti Cahyo Hidayanto, S.Si., M.Kom**

(Penguji II)

# **RANCANG BANGUN APLIKASI USE CASE POINT UNTUK ESTIMASI HARGA PERANGKAT LUNAK (STUDI KASUS: PERUSAHAAN DTS)**

**Nama Mahasiswa : Mukhamad Faiz Fanani**  
**NRP : 5211 100 043**  
**Jurusan : Sistem Informasi FTIf-ITS**  
**Dosen Pembimbing : Sholiq, S.T ,M.Kom, M.SA**  
**Feby Artwodini Muqtadiroh, S.Kom,**  
**MT**

## **ABSTRAK**

*Perusahaan DTS merupakan perusahaan yang bergerak dalam bidang pengembangan perangkat lunak berbasis proyek. Sebelum proyek pengembangan perangkat lunak dikerjakan, tim pengembang di DTS melakukan estimasi nilai harga proyek perangkat lunak terlebih dahulu untuk kemudian dibuatkan dokumen penawaran. Namun, sebelum dokumen penawaran dibuat, tim pengembang harus melaporkan dulu kepada direktur untuk dilakukan validasi. Setelah itu, dokumen penawaran dicetak oleh sekretaris.*

*Selama proses estimasi, tim pengembang sering menemui beberapa permasalahan. Pertama, dalam proses estimasi harga perangkat lunak, tim pengembang berbeda pendapat antara anggota satu dengan anggota lainnya. Hal ini karena dalam menentukan harga, masih menggunakan pendapat masing-masing individu tanpa ada dasar yang jelas. Kedua, dalam melakukan validasi proyek, direktur kesulitan dalam memberikan keputusan apakah nilai proyek yang diajukan oleh tim pengembang terlalu murah atau terlalu mahal. Ketiga, terkadang dalam melakukan estimasi harga, direktur dan tim pengembang sering salah paham karena tim tidak melaporkan hasil estimasi untuk dilakukan validasi.*

*Oleh karena itu, aplikasi Use Case Point sangat dibutuhkan untuk estimasi harga perangkat lunak berdasarkan metode Use Case Point (UCP) dan proses bisnis perusahaan DTS. UCP merupakan metode untuk menentukan usaha (Effort) berdasarkan Use Case. Nilai Effort kemudian diubah kedalam bentuk nilai biaya. Sebenarnya, terdapat beberapa metode dalam estimasi usaha (Effort), salah satunya yaitu Metode Use Case Point (UCP). Metode estimasi menggunakan Use Case Point dipilih karena metode ini memiliki keakuratan lebih tinggi dibanding dengan metode lainnya.*

*Metode pengembangan perangkat lunak untuk membuat aplikasi Use Case Point adalah metode incremental model. Metode incremental model dipilih karena dapat meminimalisir ketidaksesuaian antara kebutuhan pengguna dengan aplikasi yang sudah dibuat.*

*Hasil dari pengerjaan tugas akhir peneliti ini berupa Aplikasi Use Case Point melalui 3 kali increment. Increment pertama ditambahkan fitur estimasi usaha. Increment kedua ditambahkan fitur biaya, increment ketiga ditambahkan fitur-fitur untuk melakukan kalibrasi perhitungan estimasi. Selain itu, dihasilkan dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL), Deskripsi Perancangan Perangkat Lunak (DPPL), dokumen pengujian, dan dokumen panduan.*

**Kata kunci— UCP, Effort, Incremental Model, SKPL, DPPL, dan Aplikasi Use Case Point**

# **DESIGN AND DEVELOP USE CASE POINT SOFTWARE FOR ESTIMATING SOFTWARE DEVELOPMENT COST (CASE STUDY: DTS CORPORATION)**

**Neme** : Mukhamad Faiz Fanani  
**NRP** : 5211 100 043  
**Departement** : Sistem Informasi FTIf-ITS  
**Supervisors** : Sholiq, S.T ,M.Kom, M.Sa  
Feby Artwodini Muqtadiroh, S.Kom,  
MT

## **ABSTRACT**

*DTS Corporation is a company that focused in developing application based project. Before start to be developed, DTS's developer team estimate about the price of the project. After that, they submit bidding document to client for approving. And then, printed out by secretary.*

*During estimation process, developer team often faced some problems. First, Developer Team have difficulties to estimate about the price of the project dan sometimes having different argument with team member. It's happened because they have no standard or rigid procedure to estimate the price of the software, So they estimate the software price by feeling of each other. Secound, director feel difficult to decide or validate the result of estimation reported by team. Director feel afraid wheter the estimation result done by developer team is higer or lower. Third, sometimes director have miss communication with developer team because developer team forget to report result of the estimation.*

*Because of that, researcher develop Use Case Point Cost Estimate Software based Use Case Point (UCP) method and business process of the cost estimation in DTS Corporation. UCP is method that used for estimate Effort. The Effort and then will be converted to form of cost value. Actually, there are some Effort estimate methods, Use Case Point (UCP) is one of them. Use Case Point method is chosen because the accuration of the estimate result is higher than another method.*

*The Development methodology that chosen for develop Use Case Point Application is incremental model. The reason for chosing this methodology is because incremental model can avoid incompatibility between application and user needs.*

*The result of this final project is Use Case Point Cost Estimate Software through three times of increment. The first increment added feature Effort estimate, second increment added feature cost estimate, third increment added feature that perform calibrate estimation result. Additionally, generated documents Software Requirement Specification (SRS), Software Design Description (SDD), Testing Documentation, and User Guide*

**Keywords— UCP, Incremental Model, SRS, SDD, and Use Case Point Cost Estimate Software**

## **KATA PENGANTAR**

Alhamdulillah atas karunia, rahmat, berkah, dan jalan kemudahan yang diberikan oleh Allah SWT, penulis mendapat kelancaran dalam menyelesaikan tugas akhir dengan judul:

### **RANCANG BANGUN APLIKASI USE CASE POINT UNTK ESTIMASI HARGA PERANGKAT LUNAK (STUDI KASUS: PERUSAHAAN DTS)**

yang merupakan salah satu syarat kelulusan untuk mendapatkan gelar sarjana di Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Pada kesempatan ini, saya ingin mengucapkan banyak terima kasih kepada pihak-pihak yang telah memberikan saran, dukungan, motivasi, semangat dan bantuan, baik dalam bentuk bantuan tenaga, spirit maupun bantuan spiritual demi terselesainya pengerjaan tugas akhir ini. Terima kasih secara khusus penulis berikan kepada:

- 1) Keluarga yang selalu memberikan dukungan tenaga dan materi demi terselesainya
- 2) Bapak Sholiq, S.T ,M.Kom, M.SA dan Ibu Feby Artowordini Muqtadiroh, S.Kom, MT selaku dosen pembimbing yang memberikan banyak waktu, tenaga, dan ilmu kepada penulis dalam menyelesaikan tugas akhir ini.
- 3) Renny Pradina Kusumawardhani,. ST.MT dan Hanim Maria Astuti, S.Kom., M.Sc selaku dosen wali yang selama ini sudah memberikan motivasi kepada penulis dalam menyelesaikan perkuliahan di Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya

- 4) Seluruh dosen Jurusan Sistem Informasi ITS yang telah memberikan ilmu yang sangat berharga dan bermanfaat bagi penulis.
- 5) Mas Faturrahman yang sudah membantu dalam memberikan data terkait topic pengerjaan tugas akhir peneliti
- 6) Seleuruh anggota perusahaan Dynamic Team Solution (DTS) yang sudah mensupport dan mendukung penulis dalam penyelesaian tugas akhir
- 7) Sahabat terbaik yang selalu mendukung dan memotivasi selama perkuliahan, Afildawina Mas Yordanis, Adit, Huda, Mas affas, mas fatur, Acip, dan Aula.
- 8) Teman-teman Lab. PPSI Stephen, Angga, Zaki, Morgan, Idam, Sita,Linda, dan lain-lain yang tidak dapat saya sebutkan satu per satu serta temen-temen SPK dan e-Bisnis
- 9) Seluruh temen-temen BASILISK, mbak/mas FOXIS dan adik-adik SOLARIS yang memberikan banyak pembelajaran selama ini
- 10) Berbagai pihak yang turut serta membantu dalam pengerjaan tugas akhir ini

Terima kasih ata segala bantuan, dukungan, dan doa yang diberikan selama ini. Semoga Allah SWT memberikan rahmat, dimudahkan segala urusan dan membalas kebaikan yang selama ini diberikan oleh penulis. Wassalamu'alaikum Wr.Wb.

Surabaya, Juni 2015

Penulis

## DAFTAR ISI

ABSTRAK .....	v
KATA PENGANTAR.....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xix
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah .....	3
1.3. Batasan Pengerjaan Tugas Akhir .....	3
1.4. Tujuan Tugas Akhir .....	3
1.5. Manfaat Tugas Akhir .....	4
1.6. Relevansi.....	4
BAB II TINJAUAN PUSTAKA .....	5
2.1. Studi Sebelumnya .....	5
2.1.1. Hasil Penelitian Terkait UCP .....	5
2.1.2. Aplikasi Use Case Point Calculator Excel .....	6
2.1.3. Website Perhitungan Use Case Point .....	7
2.2. Use Case Point (UCP).....	8
2.2.1. Pengertian.....	8
2.2.2. Perhitungan UCP.....	8
2.3. Mengubah Nilai UCP Kedalam Bentuk <i>Effort</i> .....	14
2.4. Distribusi Usaha Kedalam Aktivitas.....	14
2.5. Menghitung Nilai Biaya.....	15
2.6. Profil Perusahaan DTS.....	16
2.7. Estimasi Harga di Perusahaan DTS .....	20
2.8. Metode Pengembangan Perangkat Lunak .....	21
2.9. <i>Incremental Model</i> .....	22
2.10. Keuntungan Model <i>Increment</i> .....	22
2.11. Kekurangan Model <i>Increment</i> .....	23
2.12. Kesesuaian Metodologi Untuk Tugas Akhir.....	23
2.13. Unified Modeling Language (UML).....	24
2.14. Diagram pada UML .....	24

2.15.	Tahapan Desain Dengan UML.....	25
2.16.	Use Case.....	26
2.17.	Use Case Diagram.....	27
2.18.	<i>Activity Diagram</i> .....	27
2.19.	<i>Squence Diagram</i> .....	28
2.20.	<i>Class Diagram</i> .....	29
2.21.	Pengujian.....	30
2.22.	Metode Uji Coba Black Box ( <i>Blackbox Testing</i> ).....	30
2.23.	Pengujian Antarmuka Dengan Aturan Shneiderman ...	31
2.24.	PHP ( <i>Hypertext Preprocessor</i> ).....	32
2.25.	Codeigniter (CI).....	33
BAB III METODOLOGI PENELITIAN.....		35
BAB IV PERANCANGAN.....		41
4.1.	Tahapan Perancangan .....	41
4.1.1.	Melakukan Wawancara .....	41
4.1.2.	Mendapatkan Kebutuhan Pengguna .....	42
4.1.3.	Pemodelan Proses Bisnis Aplikasi .....	42
4.1.4.	Melakukan Desain <i>Use Case</i> .....	43
4.1.5.	Desain <i>Squence Diagram</i> .....	43
4.1.6.	Desain <i>Class Diagram</i> .....	43
4.1.7.	Pembuatan Model Data Konseptual .....	43
4.1.8.	Pembuatan Model Data Fisik .....	44
4.1.9.	Melakukan Desain Antarmuka .....	44
4.1.10.	Perancangan Pengujian.....	44
4.2.	Perancangan Tahap <i>Increment 1</i> .....	44
4.2.1.	Penggalian Kebutuhan.....	44
4.2.2.	Analisa Kebutuhan Aplikasi.....	49
4.2.3.	Desain Use Case Diagram .....	55
4.2.4.	<i>Use Case Description</i> .....	58
4.2.5.	Desain <i>Squence Diagram</i> .....	62
4.2.6.	Desain <i>Class Diagram</i> .....	63
4.2.7.	Conseptual Data Model(CDM) .....	64
4.2.8.	Physical Data Model (PDM) .....	65
4.2.9.	Desain Antarmuka Pengguna .....	66
4.2.10.	Perancangan Pengujian.....	69

4.3.	Perancangan Tahap <i>Increment</i> 2.....	75
4.3.1.	Penggalian Kebutuhan.....	75
4.3.2.	Analisa Kebutuhan.....	78
4.3.3.	Hak Akses Pengguna.....	84
4.3.4.	Desain Use Case Diagram.....	85
4.3.5.	<i>Use Case Description</i> .....	87
4.3.6.	Desain Sequence Diagram.....	90
4.3.7.	Desain <i>Class Diagram</i> .....	91
4.3.8.	Conceptual Data Model(CDM).....	92
4.3.9.	Physical Data Model (PDM).....	93
4.3.10.	Desain Antarmuka Pengguna.....	94
4.3.11.	Perancangan Pengujian.....	97
4.4.	Perancangan Tahap <i>Increment</i> 3.....	99
4.4.1.	Penggalian Kebutuhan.....	99
4.4.2.	Analisa Kebutuhan Aplikasi.....	100
4.4.3.	Desain <i>Use Case Diagram</i> .....	106
4.4.4.	<i>Use Case Description</i> .....	107
4.4.5.	Desain <i>Sequence Diagram</i> .....	109
4.4.6.	Desain <i>Class Diagram</i> .....	110
4.4.7.	Conceptual Data Model(CDM).....	111
4.4.8.	Physical Data Model (PDM).....	112
4.4.9.	Desain Antarmuka Pengguna.....	113
4.4.10.	Perancangan Pengujian.....	119
BAB V IMPLEMENTASI.....		125
5.1.	Tahapan Implementasi.....	125
5.1.1.	Pembuatan Database.....	125
5.1.2.	Pengkodean/Implementasi.....	125
5.1.3.	Pengujian.....	126
5.2.	Implementasi Tahap <i>Increment</i> 1.....	126
5.2.1.	Desain Database.....	126
5.2.2.	Sampel Implementasi Aplikasi.....	127
5.2.3.	Pengujian.....	131
5.3.	Implementasi Tahap <i>Increment</i> 2.....	133
5.3.1.	Desain Database.....	133
5.3.2.	Sampel Implementasi Aplikasi.....	133

5.3.3.	Pengujian.....	137
5.4.	Implementasi Tahap <i>Increment</i> 3.....	139
5.4.1.	Desain Database.....	139
5.4.2.	Sample Implementasi Aplikasi.....	140
5.4.3.	Pengujian.....	142
BAB VI	HASIL DAN PEMBAHASAN.....	149
6.1.	Aplikasi Use Case Point (UCP).....	149
6.2.	Verifikasi.....	151
6.3.	Validasi Aplikasi.....	151
6.4.	Data Percobaan.....	152
6.5.	Perhitungan Biaya Dengan Aplikasi.....	157
6.6.	Hasil Percobaan Menggunakan Aplkasi UCP.....	157
6.7.	Pembahasan.....	161
BAB VI	PENUTUP.....	163
7.1.	Kesimpulan.....	163
7.2.	Saran.....	165
DAFTAR	PUSTAKA.....	167
LAMPIRAN A-	List Pertanyaan awal.....	A- 1 -
LAMPIRAN B-	Rekaman Wawancara.....	B- 1 -
LAMPIRAN C-	Interview Result.....	C- 1 -
LAMPIRAN D-	Acceptance checklist.....	D- 1 -
LAMPIRAN E-	Hasil Verifikasi Aplikasi.....	E- 1 -
LAMPIRAN F-	Data Use Case Percobaan.....	F- 1 -
LAMPIRAN G-	Lampiran Penjelasan indikator TCF.....	G- 1 -
LAMPIRAN H-	Lampiran Penjelasan Indikator ECF.....	H- 1 -
LAMPIRAN I-	Lampiran Penjelasan Aktivitas.....	I- 1 -
LAMPIRAN J-	Lampiran Struktur Organisasi DTS.....	J- 1 -
RIWAYAT	PENULIS.....	191

## DAFTAR TABEL

Tabel 2.1 Deskripsi use case .....	9
Tabel 2.2 Klasifikasi Bobot Use Case.....	10
Tabel 2.3 Klasifikasi Bobot Aktor .....	11
Tabel 2.4 Daftar indikator TCF.....	12
Tabel 2.5 Daftar indikator ECF.....	13
Tabel 2.6 Distribusi Usaha dari masing-masing aktivitas .....	15
Tabel 2.7 Daftar Tarif/Gaji Tenaga TI.....	16
Tabel 2.8 Kesesuaian Metodologi Model Increment .....	23
Tabel 3.1 Detail Alur Pengerjaan Tugas Akhir .....	37
Tabel 4.1 Translasi kebutuhan.....	45
Tabel 4.2 Daftar kebutuhan aplikasi UCP.....	47
Tabel 4.3 Daftar Kebutuhan fungsional aplikasi.....	48
Tabel 4.4 Daftar tools pengujian kualitas aplikasi .....	49
Tabel 4.5 Daftar kalasifikasi use case .....	51
Tabel 4.6 Klasifikasi pembobotan aktor.....	52
Tabel 4.7 Pembobotan masing-masing indikator TCF.....	53
Tabel 4.8 Daftar indiaktor beserta bobot.....	54
Tabel 4.9 <i>Use Case Description</i> menambah nilai UUCW .....	59
Tabel 4.10 Identifikasi Pengujian kebutuhan fungsional <i>increment 1</i> .....	69
Tabel 4.11 Prosedur Pengujian kebutuhan fungsional <i>increment 1</i> .....	71
Tabel 4.12 Identifikasi Pengujian kebutuhan non- fungsional <i>increment 1</i> .....	72
Tabel 4.13 Prosedur pengujian kebutuhan non-fungsional <i>increment 1</i> .....	73
Tabel 4.14 Translasi kebutuhan.....	75
Tabel 4.15 Tabel kebutuhan fungsional sistem.....	77
Tabel 4.16 Tim pengembang perangkat lunak .....	80
Tabel 4.17 Distribusi Usaha dari masing-masing aktivitas .....	81
Tabel 4.18 Daftar personel yang ada pada masing-masing aktivitas .....	82
Tabel 4.19 Pemetaan jabatan tim dengan jabatan standard Kelly	83

Tabel 4.20 Standard gaji Kelly setelah diturunkan 50% .....	83
Tabel 4.21 Peran pengguna .....	84
Tabel 4.22 <i>Use case diagram</i> memasukan deskripsi aplikasi .....	88
Tabel 4.23 Identifikasi Pengujian kebutuhan fungsional <i>increment 2</i> .....	97
Tabel 4.24 Prosedur Pengujian kebutuhan fungsional <i>increment 2</i> .....	98
Tabel 4.25 Translasi kebutuhan.....	99
Tabel 4.26 Daftar kebutuhan fungsional aplikasi UCP <i>increment 3</i> .....	100
Tabel 4.27 Daftar kebutuhan non-fungsional <i>increment 3</i> .....	100
Tabel 4.28 Distribusi Aktivitas pembangunan aplikasi dengan CMS .....	102
Tabel 4.29 Formulir memasukan actual <i>Effort</i> .....	103
Tabel 4.30 Daftar tools pengujian kualitas aplikasi.....	105
Tabel 4.31 Memasukan <i>actual Effort</i> .....	107
Tabel 4.32 Identifikasi Pengujian kebutuhan fungsional <i>increment 3</i> .....	119
Tabel 4.33 Prosedur Pengujian kebutuhan fungsional <i>increment 3</i> .....	120
Tabel 4.34 Identifikasi Pengujian kebutuhan non-fungsional <i>increment 3</i> .....	122
Tabel 4.35 Prosedur Pengujian kebutuhan non-fungsional <i>increment 3</i> .....	123
Tabel 5.1 Pelaporan hasil pengujian tahap <i>increment 2</i> .....	131
Tabel 5.2 Pelaporan hasil pengujian tahap <i>increment 2</i> .....	137
Tabel 5.3 Pelaporan pengujian <i>incement 3</i> .....	143
Tabel 5.4 Hasil pengujian pada tahap <i>increment 3</i> .....	145
Tabel 5.5 Indikator pengujian <i>Strive for Consistency</i> .....	147
Tabel 5.6 Indiaktor pengujian <i>Cater to Universal Usability</i> .....	147
Tabel 6.1 Deskripsi umum proyek .....	152
Tabel 6.2 Data Actual <i>Effort</i> aplikasi TDI .....	153
Tabel 6.3 Data transaksi pada setiap use case .....	153
Tabel 6.4 Data aktor aplikasi sim pelayanan publik.....	155
Tabel 6.5 Data Penilaian TCF .....	156

Tabel 6.6 Data hasil penilaian ECF.....	156
Tabel 6.7 Perbandingan hasil estimasi dengan aktual .....	162

*“Halaman ini sengaja dikosongkan”*

## DAFTAR GAMBAR

Gambar 1.1 Roadmap Laboraturium (PPSI) .....	4
Gambar 2.1 Aplikasi UCP versi excel.....	6
Gambar 2.2 Aplikasi <i>UCP</i> berbasis website .....	7
Gambar 2.3 Struktur organisasi DTS .....	17
Gambar 2.4 Alur proses estimasi harga Perusahaan DTS .....	20
Gambar 2.5 bentuk dari model <i>Increment</i> [28] .....	22
Gambar 2.6 Contoh use case diagram .....	27
Gambar 2.7 Contoh dari activity diagram [36]. .....	28
Gambar 2.8 Contoh Sequence Diagram .....	29
Gambar 2.9 Contoh <i>Class Diagram</i> .....	30
Gambar 2.10 Sistem Kerja Codeigniter (Ardhana, 2013) .....	33
Gambar 3.1 Metodologi Penelitian .....	35
Gambar 4.1 Alur estimasi perhitungan usaha metode UCP .....	50
Gambar 4.2 Desain <i>use case</i> pada aktor tim pengembang .....	56
Gambar 4.3 Desain <i>use case</i> fitur autentifikasi .....	57
Gambar 4.4 Desain use case fitur perhitungan nilai UUCW .....	57
Gambar 4.5 Desain <i>use case</i> memasukan deskripsi aplikasi.....	58
Gambar 4.6 <i>Squence Diagram</i> menambah nilai UUCW.....	62
Gambar 4.7 Desain <i>Class Diagram</i> tahap <i>increment</i> 1 .....	63
Gambar 4.8 Desain konseptual data model .....	64
Gambar 4.9 Pysical Data Model(PDM aplikasi UCP .....	65
Gambar 4.10 Desain halaman login .....	66
Gambar 4.11 <i>Form</i> deskripsi aplikasi .....	67
Gambar 4.12 Form penilaian Use Case (UUCW) .....	67
Gambar 4.13 Popup informasi perhitungan UUCW .....	68
Gambar 4.14 Proses penentuan harga perangkat lunak.....	79
Gambar 4.15 Desain <i>use case</i> aktor tim pengembang.....	85
Gambar 4.16 Desain <i>use case</i> aktor direktur.....	86
Gambar 4.17 Desain <i>use case</i> aktor sekretaris .....	87
Gambar 4.18 <i>Squence Diagram</i> memasukan deskripsi aplikasi..	90
Gambar 4.19 Desain <i>Class Diagram</i> tahap <i>increment</i> 2 .....	91
Gambar 4.20 <i>Conseptual Data Model (CDM)</i> .....	92

Gambar 4.21 Physical Data Model (PDM) .....	93
Gambar 4.22 Desain form deskripsi aplikasi.....	94
Gambar 4.23 <i>Report</i> estimasi distribusi aktivitas dan biaya.....	95
Gambar 4.24 Desain tampilan daftar pengguna .....	96
Gambar 4.25 Desain Use Case pada tahap <i>increment 3</i> .....	106
Gambar 4.26 <i>Sequence Diagram</i> Memasukan actual <i>Effort</i> .....	109
Gambar 4.27 Desain <i>Class Diagram</i> tahap <i>increment 3</i> .....	110
Gambar 4.28 Conceptual Data Model (CDM) <i>Increment 3</i> .....	111
Gambar 4.29 Physical Data Model (PDM) <i>Increment 3</i> .....	112
Gambar 4.30 Desain form informasi client .....	113
Gambar 4.31 Form deskripsi aplikasi.....	114
Gambar 4.32 Desain form entry actual <i>Effort</i> .....	115
Gambar 4.33 Desain rekapitulasi akhir perhitungan .....	116
Gambar 5.1 Desain database aplikasi UCP <i>increment 1</i> .....	127
Gambar 5.2 Interface form deskripsi aplikasi.....	127
Gambar 5.3 sample kode user_model.....	128
Gambar 5.4 Sample kode controller estimasi .....	129
Gambar 5.5 Sampel kode view daftar pengguna.....	129
Gambar 5.6 Desain database aplikasi UCP <i>increment 2</i> .....	133
Gambar 5.7 interface hasil estimasi biaya .....	134
Gambar 5.8 Sampel kode <i>view</i> hasil estimasi.....	134
Gambar 5.9 kode perhitungan biaya pada <i>controller</i> estimasi ..	135
Gambar 5.10 sampel kode pada <i>model</i> log_biaya_model .....	135
Gambar 5.11 Desain database pada tahap <i>increment 3</i> .....	139
Gambar 5.12 Antarmuka informasi pelanggan.....	140
Gambar 5.13 Sampel kode <i>view</i> informasi pelanggan.....	141
Gambar 5.14 Sampel kode <i>controller</i> Log_estimasi .....	141
Gambar 5.15 Sampel kode model Client_model.....	142
Gambar 6.1 Halaman awal ( <i>Homepage</i> ) aplikasi UCP .....	149
Gambar 6.2 <i>form</i> informasi pelanggan .....	150
Gambar 6.3 Halaman daftar log estimasi .....	150
Gambar 6.4 Perhitungan UUCW dengan aplikasi.....	157
Gambar 6.5 Hasil perhitungan UAW .....	158
Gambar 6.6 Hasil perhitungan <i>Effort</i> .....	158
Gambar 6.7 Perhitungan rekap biaya .....	159

Gambar 6.8 Notifikasi e-mail .....	160
Gambar 6.9 e-mail masuk .....	160
Gambar 6.10 Dokumen penawaran .....	161

*Halaman ini sengaja dikosongkan*

## DAFTAR LAMPIRAN

LAMPIRAN A-List Pertanyaan awal .....	A- 1 -
LAMPIRAN B-Rekaman Wawancara .....	B- 1 -
LAMPIRAN C-Interview Result.....	C- 1 -
LAMPIRAN D-Acceptance checklist .....	D- 1 -
LAMPIRAN E-Hasil Verifikasi Aplikasi .....	E- 1 -
LAMPIRAN F-Data Use Case Percobaan .....	F- 1 -
LAMPIRAN G-Lampiran Penjelasan indikator TCF.....	G- 1 -
LAMPIRAN H-Lampiran Penjelasan Indikator ECF .....	H- 1 -
LAMPIRAN I-Lampiran Penjelasan Aktivitas .....	I- 1 -
LAMPIRAN J-Lampiran Struktur Organisasi DTS .....	J- 1 -

# 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 relevansi pengerjaan 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 (*third party*) dalam bentuk sebuah 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]. Perusahaan DTS merupakan perusahaan yang bergerak dalam bidang pengembangan perangkat lunak berbasis proyek. Sebelum proyek pengembangan perangkat lunak dikerjakan, tim pengembang di DTS melakukan estimasi nilai harga dari proyek perangkat lunak terlebih dahulu untuk kemudian dikirimkan dokumen penawaran ke calon *client*-nya. Selama ini, proses estimasi menggunakan perkiraan yang didasarkan atas diskusi dari tim yang akan melakukan pengerjaan proyek tersebut. Setelah estimasi dilakukan, tim proyek melaporkan kepada direktur untuk mendapatkan persetujuan. Setelah persetujuan dilakukan, tim mengirimkan dokumen penawaran yang dikirimkan ke calon pelanggan (*client*). Kemudian tim menunggu tanggapan dari *client* mengenai penawaran yang dikirim. Jika penawaran disetujui, maka akan dilakukan penandatanganan kontrak sebagai awal pengerjaan proyek.

Selama proses menentukan harga estimasi tersebut, Perusahaan DTS sering menemui beberapa masalah. Pertama, dalam proses estimasi harga proyek sebelum dilaporkan ke atasan(Direktur), tim proyek kesulitan dalam menentukan nilai dari proyek itu sendiri. Hal ini dikarenakan untuk menentukan harga proyek masih tergantung dari pendapat subjektif individu masing-masing anggota dalam tim. Kedua, dalam melakukan validasi nilai proyek, atasan/direktur tidak bisa menentukan apakah nilai yang diajukan terlalu murah atau terlalu mahal. Jika terlalu murah, akan merugikan perusahaan dan tim sendiri. Jika terlalu mahal, direktur takut calon *client* tidak setuju dan pada akhirnya proyek akan gagal dikerjakan. Ketiga, dalam penyampaian ke direktur, tim pengembang terkadang salah paham dengan direktur karena tim pengembang tidak melaporkan hasil estimasi harga.

Oleh karena itu, peneliti membuat aplikasi Use Case Point untuk melakukan estimasi harga perangkat lunak berdasarkan metode Use Case Point dan alur bisnis dari Perusahaan DTS . UCP adalah metode untuk mengukur besarnya *Effort* yang harus dilakukan dalam pembuatan perangkat lunak berdasarkan use case yang ada [3]. Alasan peneliti menggunakan metode UCP ini karena tingkat akurasi metode UCP dalam menentukan estimasi usaha (*Effort*) ini lebih tinggi daripada metode estimasi harga lainnya [4].

Sebenarnya, sudah ada aplikasi estimasi usaha dalam pengembangan proyek dengan metode UCP, tapi dalam aplikasi tersebut tidak menunjukkan nilai biaya yang harus dikeluarkan dalam pengembangan perangkat lunak itu sendiri. Aplikasi yang akan peneliti buat nanti, selain menentukan nilai *Effort*, juga memberikan biaya /cost dalam pengembangan perangkat lunak yang disesuaikan dengan alur estimasi harga perangkat lunak yang ada di Perusahaan DTS.

Tahapan yang peneliti lakukan dalam pembuatan aplikasi ini yaitu: merancang spesifikasi kebutuhan perangkat lunak, merancang perangkat lunak, membangun perangkat lunak dan melakukan *testing* 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 latar belakang tersebut, maka rumusan masalah yang diambil sebagai berikut.

1. Apa saja spesifikasi kebutuhan perangkat lunak yang diperlukan sebagai pedoman perancangan dan pembangunan Aplikasi Use Case Point Untuk Estimasi Perangkat Lunak?
2. Bagaimana hasil perancangan Aplikasi Use Case Point Untuk Estimasi Perangkat Lunak ?
3. Bagaimana hasil pengembangan Aplikasi Use Case Point Untuk Estimasi Perangkat Lunak?

### **1.3. Batasan Pengerjaan Tugas Akhir**

Batasan masalah yang terdapat dalam pengerjaan tugas akhir ini antara lain sebagai berikut.

1. Metode estimasi yang digunakan dalam estimasi harga yaitu metode Use Case Point
2. Use Case yang digunakan untuk perhitungan nilai usaha ini yaitu use case pada level user goal
3. Implementasi aplikasi hanya terbatas pada pemasangan aplikasi di pada web hosting dan pembuatan user guide di lingkup Perusahaan DTS.
4. Pengerjaan tugas akhir ini tidak meliputi bahasan keamanan infrastruktur teknologi informasi.
5. Tahapan pengerjaan tugas akhir ini dikerjakan mulai tahap persiapan hingga tahap implementasi aplikasi.
6. Proses pengaman aplikasi hanya terbatas pada pengamanan dengan username dan password.

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

Tujuan dari tugas akhir ini adalah sebagai berikut:

Menghasilkan aplikasi Use Case Point untuk mengestimasi harga perangkat lunak pada studi kasus perusahaan DTS.

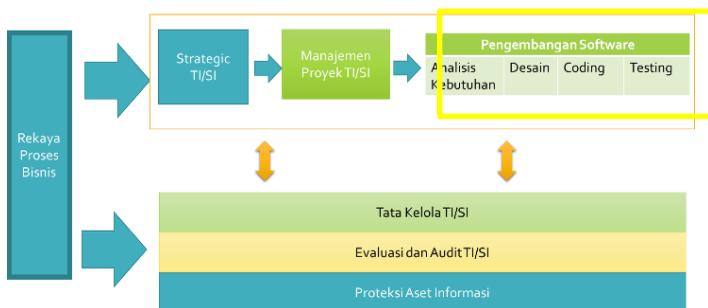
## 1.5. Manfaat Tugas Akhir

Tugas akhir ini diharapkan dapat memberikan manfaat antara lain sebagai berikut.

1. Membantu Perusahaan DTS dalam menentukan Harga Perkiraan Sementara (HPS) atau harga estimasi awal perangkat lunak dengan metode Use Case Point
2. Memberikan kemudahan dan kejelasan tim proyek Perusahaan DTS dalam menentukan nilai proyek pengembangan perangkat lunak.
3. Memberikan kemudahan bagi atasan perusahaan DTS dalam melakukan validasi terhadap nilai proyek pengembangan perangkat lunak.

## 1.6. Relevansi

Relevansi pengerjaan tugas akhir ini yaitu terkait dengan pengembangan software. Pengembangan software dimulai dari tahap analisa kebutuhan, desain, pengkodean hingga pengujian (*Testing*). Gambar 1.1 berikut ini adalah roadmap penelitian dari Laboraturium Perencanaan dan Pengembangan Sistem informasi (PPSI) ITS. Pada gamabr tersebut terlihat bahwa relevansi tugas akhir ini.



**Gambar 1.1 Roadmap Laboraturium (PPSI)**

Gambar 1.1 menunjukkan bahwa pengembangan perangkat lunak merupakan kelanjutan dari manajemen proyek Teknologi Informasi dan Sistem Informasi (SI/TI).

## **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**

Pada bagian ini akan dijelaskan mengenai studi sebelumnya tentang aplikasi perhitungan dengan menggunakan metode Use Case Point (UCP) sebagai berikut.

##### **2.1.1. Hasil Penelitian Terkait UCP**

Berikut ini merupakan hasil penelitian yang dilakukan oleh ahli yang menyebutkan bahwa estimasi dengan menggunakan metode UCP ini mendekati dengan estimasi sebenarnya.

- UCP mempunyai deviasi sebesar 6% (Nageswaran, 2001) [9]  
UCP menghasilkan nilai estimasi *Effort* 367 *man-days*, sedangkan hasil perhitungan terhadap *actual Effort* sebesar 390 *man-days*. Penyimpangan yang terjadi yaitu sebesar 6%.
- UCP memiliki deviasi sebesar 19%, sedangkan untuk estimasi para ahli (*Expert Judgement*) memiliki rata-rata deviasi sebesar 20% (Anda, 2002) [10]
- UCP memiliki nilai deviasi sebesar 9% (Carroll, 2005) [11].  
Penelitian ini menghasilkan temuan dari 200 proyek perangkat lunak yang diteliti membuktikan akurasi yang diperkirakan kurang dari 9% dari

deviasi sebenarnya untuk biaya diperkirakan pada 95% dari proyek-proyek perangkat lunak yang sudah dikerjakan.

### 2.1.2. Aplikasi Use Case Point Calculator Excel

Aplikasi Use Case Point Calculator (UCP) Excel ini dikembangkan oleh Scott Sehlhorst dan *dipublish* pada tahun 2007 [5]. Aplikasi ini berupa file excel yang diberikan rumus-rumus Use Case sehingga pengguna hanya memasukan variable-variable langsung. Output dari aplikasi ini yaitu berupa nilai usaha dari pengembangan perangkat lunak. Kelemahan dari aplikasi ini yaitu aplikasi hanya menampilkan nilai usaha saja, tidak sampai kedalam bentuk nilai biaya. Gambar 2.1 berikut ini adalah bentuk aplikasi UCP excel. Selain itu, aplikasi ini masih tergantung dengan aplikasi lain, yaitu aplikasi Microsoft Excel.

Calculations From Other Tabs		
TCF	Technical Complexity Factor	0.6
EF	Environmental Factor	1.4
UUCP	Unadjusted Use Case Points	0
AW	Actor Weighting	0
Calculation of Use Case Points		
UCP	Use Case Points	0.0
Calculation of Estimated Effort		
Ratio	Hours of Effort per Use Case Point	26
<b>Hours of Effort</b>		-
<p>For additional guidance with this page, check out the following articles at Tyner Blain</p> <p><a href="#">Software Cost Estimation With Use Case Points - Introduction</a></p> <p><a href="#">Software Cost Estimation With Use Case Points - Final Calculations Spreadsheet</a></p> <p><a href="#">Software Cost Estimation With Use Case Points - Free Excel Spreadsheet</a></p>		
Steps to Calculate Use Case Points		
0	For all tabs, enter values only in the highlighted cells	
1	Enter Technical Complexity Factors on the Technical tab	
2	Enter Environmental Factors on the Environmental tab	
3	Identify Use Cases on the Use Case tab	

Gambar 2.1 Aplikasi UCP versi excel

Jadi, aplikasi UCP ini tidak akan jalan apabila tidak ada aplikasi pendukung, yaitu dalam hal ini aplikasi Microsoft Excel.

### 2.1.3. Website Perhitungan Use Case Point

Aplikasi Use Case Point Calculator ini dikembangkan oleh Mechal Kochi [6]. Aplikasi ini melakukan perhitungan estimasi durasi sebuah proyek berdasarkan *Use Cases*, *Actors*, *Technical and Enviromental Factors*. Adapun contoh dari aplikasi ini ada pada Gambar 2.2 berikut ini.

Gambar 2.2 Aplikasi UCP berbasis website

Kelemahan dari aplikasi ini yaitu hanya menampilkan hour/durasi dari proyek, tapi tidak menghasilkan nilai biaya dari proyek tersebut. Kelebihan aplikasi yang akan dibuat oleh peneliti dibandingkan dengan aplikasi ini adalah penentuan nilai dari perangkat lunak yang akan dibuat. Jadi, selain memberikan keluaran jumlah usaha yang diperlukan, juga memberikan keluaran berupa nilai total biaya yang diperlukan untuk mengembangkan perangkat lunak itu sendiri.

## 2.2. Use Case Point (UCP)

### 2.2.1. Pengertian

Use Case point adalah teknik untuk melakukan estimasi ukuran suatu proyek pengembangan perangkat lunak. UCP ini didesain untuk kebutuhan spesifik dari sistem yang berdasarkan Use Case [3]. UCP ini menggunakan use case yang merupakan bagian utama dari UML. Metode Use Case Point (UCP) mempunyai kemampuan dalam menentukan man-hour dari proyek pengembangan perangkat lunak [7]. Adapun keunggulan menggunakan metode UCP ini yaitu:

1. Metode UCP ini memberikan estimasi yang lebih akurat dari beberapa metode tradisional yang ada.
2. Metode ini akan sangat cocok jika digunakan untuk project yang menggunakan use case sebagai informasi utama yang ada saat awal proyek dimulai
3. UCP mempertimbangkan faktor teknis (tecnicaal) dan faktor lingkungan (environmental) yang bisa menambah kekuratan dalam melakukan estimasi
4. Tidak membutuhkan detail kebutuhan yang lebih mendalam dalam melakukan estimasi [8]

### 2.2.2. Perhitungan UCP

Metode UCP menganalisa use case, actor dari use case, secenario, faktor lingkungan dan faktor technical untuk kemudian dibuatkan dalam bentuk sebuah persamaan. UCP disusun atas 3 variabel:

- a. *Unadjusted Use Case Points (UUCP)*.
- b. *The Technical Complexity Factor (TCF)*.
- c. *The Environment Complexity Factor (ECF)*.

Variabel-variabel tersebut didefinisikan dan dilakukan perhitungan secara berbeda-beda dengan menggunakan nilai

bobot pada setiap masing-masing variabel, nilai *subjective*, dan batasan konstanta.

a. *Unadjusted Use Case Points* (UUCP)

UUCP didapatkan melalui dua perhitungan yaitu sebagai berikut.

1. *Unadjusted Use Case Weight* (UUCW)

UUCW didapatkan dari total aktivitas yang ada dalam setiap scenario/transaksi use case. Use Case yang akan dilakukan pembobotan harus ditulis dalam pendekatan level yang sama, yaitu dalam level User-goals dari 5 level Use Case yang diidentifikasi oleh Alistair Cockburn [12]. Tabel 2.1 berikut ini adalah contoh Use Case yang ada pada level use goal.

**Tabel 2.1 Deskripsi use case**

<b>UC.07   Memverifikasi Data Pra-pemohonan dengan Dokumen Asli</b>	
<b>Ringkasan :</b>	<i>Use case</i> ini berfungsi untuk memverifikasi data <i>softcopy</i> yang diunggah dengan dokumen aslinya ( <i>hardcopy</i> ).
<b>Direct Actor:</b>	UPTSA
<b>Prioritas :</b>	<i>Essential</i>
<b>Pre Conditions:</b>	Pemohon telah melakukan pencetakan bukti pendaftaran permohonan.
<b>Skenario Sukses Utama :</b>	<ol style="list-style-type: none"> <li>1. Aktor melakukan pengecekan antara data-data yang diisikan maupun diunggah ke sistem dengan dokumen asli, kemudian memberikan status verifikasi dengan menekan tombol Centang.</li> <li>2. Sistem akan menampilkan pesan “Apakah anda yakin ingin memverifikasi data pra-pemohonan ini?”.</li> <li>3. Aktor menekan tombol Ya,</li> <li>4. Sistem otomatis men-<i>generate</i> nomor permohonan masuk untuk Disperdagin.</li> </ol>

<b>Skenario Alternatif :</b>	<ul style="list-style-type: none"> <li>• Jika aktor menekan tombol Silang, maka sistem akan menampilkan pesan “Apakah anda yakin ingin membatalkan data pra-permohonan ini?”</li> <li>• Jika aktor menekan tombol Ya, maka secara otomatis keluar pop-up untuk pemilihan tindakan selanjutnya: <u>Perpanjangan</u> atau <u>Perubahan</u>.</li> </ul>
------------------------------	--

Dalam UUCW ini, ada 3 kategori dalam menentukan nilai use case, yaitu: simple, average, dan kompleks. Setiap use case dikategorikan dari jumlah sekenario/transaksi yang terkandung didalamnya. Transaksi menurut, Ivar Jacobson, didefinisikan sebuah “round trip” dari pengguna kedalam system, kemudian system kedalam pengguna. Sebuah transaksi akan selesai jika system menunggu inputan baru/stimulus baru [13]. Transaksi dalam use case ini diidentifikasi dalam penggunaan textual, yaitu dalam bentuk sebuah text. Transaksi yang digunakan untuk menentukan kompleksitas dari use case ini merupakan scenario sukses dari sebuah use case [14]. Berdasarkan tabel diatas, terlihat bahwa jumlah scenario sukses yang ada pada Use Case “Memverifikasi Data Pra-permohonan dengan Dokumen Asli” mempunyai adalah 4 buah. Sehingga jumlah transaksi dalam use case ini adalah 2 buah. Setelah dilakukan pengkategorian, dilakukan kalkulasi dengan cara mengalikan jumlah use case yang ada dalam setiap kategori dengan bobot di masing-masing kategori. Hasilnya kemudian dijumlahkan. Adapun klasifikasi use case beserta pembobotannya ada pada Tabel 2.2 dibawah ini.

**Tabel 2.2 Klasifikasi Bobot Use Case**

<b>Tipe</b>	<b>Jumlah Transaksi</b>	<b>Bobot</b>
Simple	1 sampai 3 transaksi	5
Average	4 sampai 7 transaksi	10

Complex	8 transaksi atau lebih	15
---------	------------------------	----

Nilai UUCW didapatkan dengan rumus berikut ini.

$$\text{UUCW} = (\text{Total No. of Simple Use Cases} \times 5) + (\text{Total No. Average Use Case} \times 10) + (\text{Total No. Complex Use Cases} \times 15)$$

## 2. *Unadjusted Actor Weight (UAW)*

UAW didapatkan berdasarkan kompleksitas dari semua actor yang ada di semua use case. Mirip dengan UUCW, UAW mengkategorikan actor berdasarkan kompleksitas dari actor itu sendiri. Actor dalam UAW ini diklasifikasikan kedalam bentuk Simple, Average dan Complex. Setiap kategori tersebut, masing-masing terdapat bobot nilai. Setelah dilakukan pengkategorian, langkah selanjutnya sama dengan UUCW yaitu mengalikan jumlah actor yang ada di setiap kategori dengan masing-masing bobot pada setiap kategori. Untuk klasifikasi faktor beserta pembobotnya ada pada Tabel 2.3 berikut ini.

**Tabel 2.3 Klasifikasi Bobot Actor**

<b>Klasifikasi Actor</b>	<b>Tipe Actor</b>	<b>Bobot</b>
<b>Simple</b>	Berinteraksi melalui baris perintah atau <i>Command Prompt</i>	1
<b>Average</b>	Berinteraksi dengan protokol komunikasi seperti (e.g. TCP/IP, FTP, HTTP, database)	2
<b>Complex</b>	Berinteraksi dengan GUI atau web page	3

Adapun rumus untuk perhitungan UAW ini sebagai berikut.

$$\text{UAW} = (\text{Total No. of Simple actors} \times 1) + (\text{Total No. Average actors} \times 2) + (\text{Total No. Complex actors} \times 3)$$

Perhitungan total dari UUCP ini didapatkan dari rumus berikut ini.

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

b. *Technical Complexity Factor*(TCF)

*Technical Complexity Factor* (TCF) ini digunakan untuk melakukan penilaian terhadap faktor-faktor teknis yang berdampak terhadap produktivitas dari pengerjaan proyek. Untuk setiap proyek, setiap faktor teknis dievaluasi dengan tim pengembang berdasarkan tingkat kompleksitasnya. Untuk penilaian terhadap tingkat kompleksitas dari setiap faktor, diberikan bobot dari 0 sampai 5. Setelah itu, bobot yang ada tadi dikalikan dengan nilai masing-masing kompleksitas faktor. Kemudian, dilakukan perhitungan secara keseluruhan terhadap faktor-faktor tersebut menghasilkan nilai TF. Untuk perhitungan nilai TCF-nya menggunakan rumus berikut ini.

$$\text{TCF} = 0,6 + (0,01 * \text{TF})$$

Adapun Tabel pembobotan dari setiap kompleksitas faktor yaitu pada Tabel 2.4 sebagai berikut.

**Tabel 2.4 Daftar indikator TCF**

Code	Indicator	Weight
T1	Distributed System Required	2.0
T2	Response Time Is Important	1.0
T3	End User Efficiency	1.0
T4	Complex Internal Processing Required	1.0
T5	Reusable Code Must Be A Focus	1.0
T6	Installation Easy	0.5
T7	Usability	0.5

Code	Indicator	Weight
T8	Cross-Platform Support	2.0
T9	Easy To Change	1.0
T10	Highly Concurrent	1.0
T11	Custom Security	1.0
T12	Dependence On Thrid-Part Code	1.0
T13	User Training	1.0

c. Enviromental Complexity Factor(ECF)

Selain faktor kompleksitas teknis, faktor kompleksitas lingkungan (Enviromental Complexity Factor) juga digunakan dalam menentukan estimasi harga proyek. Sama halnya dengan TCF, ECF ini juga ada pembobotan dari setiap faktor yang ada. Jangkauan (range) pembobotann yang ada yaitu antara 0(nol) sampai 5. Perhitunganya yaitu dengan mengalihkan bobot dengan nilai di masing-masing faktor kemudian nilainya dijumlahkan menjadi nilai EF. Untuk menghitung ECF, menggunakan rumus berikut.

$$ECF = 1.4 + (-0.03 \times EF)$$

Adapun factor-faktor dalam kompleksitas lingkungan ini ada pada Tabel 2.5 berikut ini.

**Tabel 2.5 Daftar indikator ECF**

Code	Indicator	Weight
E1	Familiarity with the Project	1.5
E2	Application Experience	0.5
E3	OO Programming Experience	1.0
E4	Lead Analyst Capability	0.5
E5	Motivation	1.0
E6	Stable Requirements	2.0
E7	Part Time Staff	-1.0
E8	Difficult Programming Language	-1.0

Setelah tiga faktor terebut dilakukan kalkulasi, tahapan selanjutnya yaitu menghitung nilai dari UCP sendiri.

Adapun perhitungan untuk nilai UCP ini menggunakan rumus berikut.

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{ECF}$$

### 2.3. Mengubah Nilai UCP Kedalam Bentuk *Effort*

UCP masih belum merepresentasikan nilai *Effort* dalam sebuah pengembangan proyek, untuk mengetahui nilai *Effort*, nilai UCP dikalikan dengan staff-hour per use case point. Adapun acuan dalam perhitungan staff-hour ini yaitu berasal dari penelitian yang dilakukan oleh Puji. Puji mengemukakan untuk nilai staff-hour per use case point sebesar **8.2** [15].

Nilai tersebut akan mengubah secara langsung nilai UCP menjadi nilai usaha (*Effort*) yaitu dalam bentuk *Hour of Effort*. Adapun rumus perhitungannya adalah sebagai berikut.

$$\text{Hours of Effort} = \text{UCP} * \text{Staff-hour per use case point}$$

### 2.4. Distribusi Usaha Kedalam Aktivitas.

Setelah nilai usaha total didapatkan, langkah selanjutnya yang dilakukan yaitu mendistribusikan usaha kedalam masing-masing aktivitas yang ada dalam pengembangan perangkat lunak. Berdasarkan penelitian yang dilakukan oleh Kassem Shaleh (2011) [10], aktivitas yang ada pada pengembangan perangkat lunak dibedakan menjadi 3 bagian, yaitu: *Software Development*, *On Going Activity*, dan *Quality and Testing*. Berikut ini adalah tabel distribusi aktivitas yang ada dalam pengembangan perangkat lunak [16]. Tabel 2.6 berikut ini merupakan tabel distribusi aktivitas yang ada pada perangkat lunak.

Tabel 2.6 Distribusi Usaha dari masing-masing aktivitas

NO.	Aktivitas	Presentase (%)
<b>1</b>	<b>Software Development</b>	
a	Requirement	7.50%
b	Specifications & Design	17.50%
c	Coding	10.00%
d	Testing	7.00%
<b>2</b>	<b>On Going Activity</b>	
a	Project Management	7.00%
b	Configuration Management	3.00%
c	Documentation	3.00%
d	Training & Support	3.00%
e	Acceptance & Deployment	5.00%
<b>3</b>	<b>Quality and Testing</b>	
a	Quality Assurance & Control	12.34%
b	Evaluation and Testing	24.66%
	Total	<b>100%</b>

## 2.5. Menghitung Nilai Biaya

Untuk menentukan nilai biaya suatu proyek, digunakan pendekatan estimasi biaya tiap aktivitas yang didapatkan dari nilai Hours of *Effort*. Hourse of *Effort* dari masing-masing aktivitas ini dikalikan dengan standard gaji dari personel yang ada di masing-masing aktivitas tersebut. Untuk standard gaji sendiri, mengacu pada Indonesian Sallary Guide yang dikeluarkan oleh Kelly Services [17]. Berikut ini adalah daftar rata-rata gaji yang dikeluarkan oleh Kelly 2013.

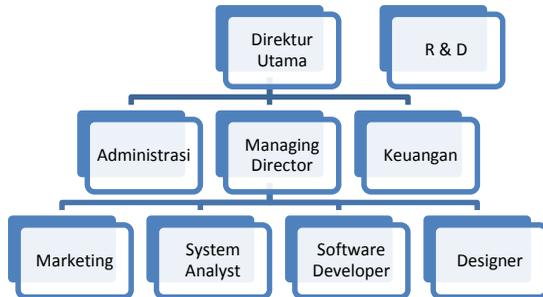
**Tabel 2.7 Daftar Tarif/Gaji Tenaga TI**

<b>Jabatan</b>	<b>Pendidikan/ pengalaman</b>	<b>Gaji /bulan(Rp)</b>
<i>Analyst Programmer/ Software Engineer</i>	Sarjana (min 1 th)	5.000.000
<i>Business/ system analyst</i>	Sarjana (min 2 th)	7.000.000
<i>Project Manager</i>	Sarjana (min 3 th)	8.500.000
<i>Software QA/test analyst</i>	Sarjana (3-6 th)	5.000.000
<i>Secretary</i>	Sarjana (min 1 th)	4.000.000

Untuk mengalikan dengan nilai *Effort*, gaji tiap bulan pada masing-masing profesi diubah kedalam bentuk jam. Setelah didapatkan dalam bentuk jam. Nilai gaji tersebut kemudian dikalikan dengan nilai *Effort* untuk mengetahui nilai biaya. Berdasarkan penelitian yang dilakukan oleh Linda, 2014 bahwa standard yang dikeluarkan oleh Kelly terlalu tinggi. Oleh karena itu, perlu dilakukan penurunan hingga 55% untuk mendekati keakuratan yang lebih baik [18].

## **2.6. Profil Perusahaan DTS**

Perusahaan DTS merupakan perusahaan yang didirikan atas inisiatif mahasiswa aktif jurusan Sistem Informasi ITS. Perusahaan DTS mulai beroperasi pada pertengahan bulan september 2011. Perusahaan DTS ini bergerak di bidang jasa pembuatan website profil pribadi maupun organisasi. Selain website, juga melayani pembuatan aplikasi berbasis web yang bisa menunjang dan mempercepat proses bisnis yang ada di perusahaan/organisasi [19]. Adapun struktur organisasi yang ada pada Perusahaan DTS ini adalah pada gambar 2.3 sebagai berikut [20].



**Gambar 2.3 Struktur organisasi DTS**

Berikut ini adalah *Job Description* dari masing-masing peran yang ada di Perusahaan DTS ini yaitu:

- **Direktur**

Bertugas sebagai pemimpin tunggal dari perusahaan yang memiliki hak penuh dalam pengambilan keputusan dan bertanggungjawab atas seluruh proses bisnis yang ada di perusahaan.

- **Managing Director (Direktur Program)**

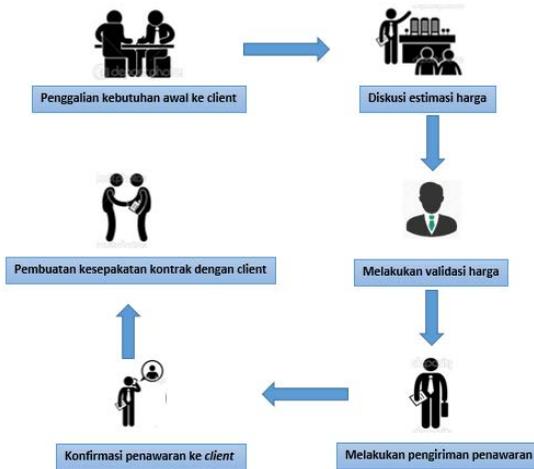
1. Bertugas sebagai tangan kanan dari Direktur dalam membantu mengkoordinasikan setiap Divisi yaitu Divisi Administrasi, Keuangan, Divisi Marketing, Divisi Software Developer dan Divisi Designer
2. Bertugas mengkoordinasikan project manager dan tim yang sedang menjalankan atau akan menjalankan sebuah project.
3. Mencari peluang kerjasama dengan membangun relasi antar individu atau perusahaan guna membangun jaringan bisnis yang lebih luas
4. Mencari dan merekomendasikan calon sdm yang tepat untuk di rekrut menjadi anggota DTS kepada direktur

- **Administrasi dan Keuangan**
  1. Bertugas membuat dan menyiapkan laporan-laporan dan dokumen-dokumen yang diperlukan oleh Direktur.
  2. Bertugas melakukan pencatatan agenda rapat dan hasil rapat saat rapat DTS bulanan berlangsung
  3. Bertugas melakukan pencatatan keluar masuk keuangan perusahaan. Membuat perencanaan harga produk dengan berkonsultasi dengan programmer.
  4. Bertugas untuk membayar pajak setiap bulan di bank dan menyiapkan dokumen pelaporan pajak setiap bulannya.
- **Marketing**
  1. Bertugas menentukan strategi pemasaran, dan produk-produk yang akan dijual atau ditawarkan kepada customer guna meningkatkan omzet dari perusahaan.
  2. Mencari dan Membuat resume profil tentang perusahaan yang akan ditarget untuk proyek DTS.
  3. Melakukan kontak dan penawaran kepada klien sesuai dengan SOP marketing DTS.
  4. Melakukan tindaklanjut terkait klien melalui User Requirement.
  5. Melakukan promo secara online dengan skala waktu tertentu melalui media social network DTS minimal 2 kali update status dalam sehari.
  6. Membuat laporan perkembangan marketing yang didaftarkan setiap minggu kepada koordinator (Managing Director) DTS.
- **System Analyst**
  1. Bertugas untuk mendampingi Managing Director, dan marketing untuk menganalisa kebutuhan yang diinginkan oleh customer.
  2. Bertugas untuk membantu Managing Director dalam menyiapkan data SKPL awal untuk kebutuhan dokumen penawaran project.

- **Software Developer**
  1. Merekomendasikan anggota tim developer yang siap untuk mengikuti project DTS kepada managing director.
  2. Membantu meningkatkan kemampuan sdm developer dengan memberikan studi kasus untuk diselesaikan.
- **Tim Developer**
  1. Bertugas menangani dalam pembuatan aplikasi dan melayani dalam memberikan solusi IT yang tepat kepada customer.
  2. Memiliki tanggungjawab untuk terus belajar dengan software Developer dan aktif serta bertanya kepada divisi R&D untuk diadakan tutorial/pelatihan.
- **Designer**
  1. Bertugas membuat desain aplikasi/web yang bekoordinasi dengan divisi software developer untuk membuat tampilan desain yang dibutuhkan.
  2. Bertugas membantu desain yang diperlukan untuk menunjang tim marketing DTS, seperti; desain kartu nama, brosur, poster dsb.
- **Riset & Development (R&D)**
  1. Bersama sama direktur dan wakil direktur untuk mentransfer knowledge serta pengalaman yang dimiliki kepada seluruh anggota DTS guna kesinambungan jalannya perusahaan.
  2. Mengagendakan program training/pelatihan minimal sebulan sekali untuk tim DTS yang memerlukannya untuk menunjang kinerja DTS Team.
  3. Melakukan riset dan monitoring riset DTS yang sedang dan akan berjalan.
  4. Mengusulkan kepada direktur/wakil untuk produk/jasa baru yang bisa diambil sesuai kebutuhan pasar terkini.

## 2.7. Estimasi Harga di Perusahaan DTS

Selama ini, perusahaan DTS dalam menjalankan bisnisnya masih berbasis proyek. Nilai proyek pengembangan perangkat lunak didasarkan atas kompleksitas dan besar ruang lingkup dari proyek tersebut. Gambar 2.4 berikut ini adalah alur proses penentuan estimasi harga proyek perangkat lunak Perusahaan DTS.



**Gambar 2.4** Alur proses estimasi harga Perusahaan DTS

Perusahaan mencari calon client yang membutuhkan aplikasi atau website. Setelah calon *client* didapatkan, tim proyek menemui calon client untuk melakukan analisa kebutuhan awal. Setelah itu, tim proyek melakukan diskusi untuk melakukan estimasi nilai proyek perangkat lunak. Setelah estimasi harga didapatkan, tim proyek mengirimkan nilai estimasi beserta rincian singkat kepada atasan/direktur Perusahaan DTS. Kemudian atasan/direktur melakukan validasi/persetujuan dari nilai proyek yang diberikan. Setelah persetujuan dilakukan, tim proyek membuat penawaran harga untuk dikirimkan ke calon *client*. Tim kemudian melakukan konfirmasi kepada client atas penawaran yang dikirmkan. Jika

calon *client* setuju dengan penawaran kita, maka proyek akan dilaksanakan dengan dibuatkan dokumen kontrak (Project charter). Pengerjaan proyek Perusahaan DTS diawali dengan pembuatan kontrak sampai ke tahap implementasi. Pada tahap pengembangan perangkat lunak, perusahaan menggunakan metode RAD. RAD ini merupakan metode pengembangan perangkat lunak yang berdasarkan atas pembagian modul antar tim. Perusahaan DTS melakukan pendokumentasian selama proses pengembangan perangkat lunak ketika ruang lingkup dan kompleksitasnya besar [21].

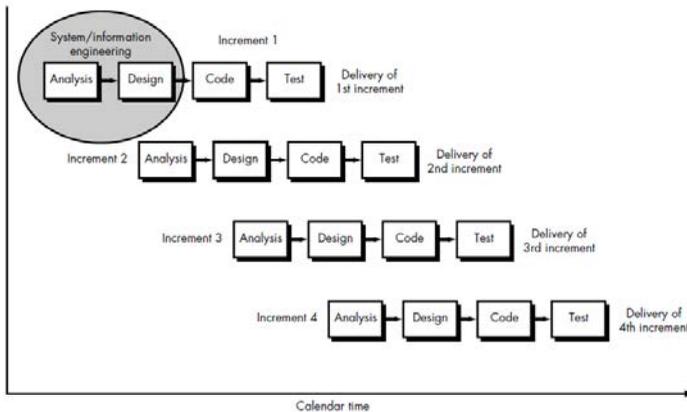
## 2.8. Metode Pengembangan Perangkat Lunak

Metodologi pengembangan perangkat lunak merupakan metode atau acuan kerja yang digunakan untuk menstrukturisasi, perencanaan, dan control dalam proses pengembangan sebuah sistem informasi [22]. Dalam metodologi juga termasuk melakukan pendefinisian awal output dari setiap proses yang ada dalam metodologi tersebut dan juga tools/artefak yang diperlukan dalam menyelesaikan sebuah proyek aplikasi [23]. Berikut ini adalah beberapa metodologi yang saat ini populer dalam pengembangan perangkat lunak [24].

Waterfall model	Feature Driven
Prototyping model	Methodology
Agile Software	Join Application
Development	Development
Rapid Application	Methodology
Development	Lean Development
(RAD)	Methodology
Dynamic System	Rational Unified
Development Model	Process
Spiral Model	Methodology
Extreme	Scrum Development
Programming	Methodology

## 2.9. Incremental Model

Model pengembangan *increment* merupakan model perangkat lunak iterative bertingkat [25] yang salah satu bentuk model proses Evolusioner (*Evolutionary Software Process Model*) [26]. Gambar 2.5 berikut ini merupakan bentuk dari model *Increment* .



Gambar 2.5 bentuk dari model *Increment* [28]

Incremental model ini juga mengambil dari filosofi dari metode prototyping. Hal ini karena pada setiap hasil *release* yang ada, bisa dijadikan sebuah *prototype* untuk development release selanjutnya. Model ini digunakan saat kebutuhan-kebutuhan dimungkinkan untuk terjadi perubahan-perubahan yang berkembang setiap waktu selama fase pengerjaan proyek [27]. Metodologi incremental mengkombinasikan elemen dari model liner squential dan filosofi dari prototyping [26]. Pengembangan perangkat lunak dengan model ini dibagi menjadi bagian yang lebih kecil yang dikerjakan secara bertingkat dan berulang. Hasil dari setiap pengulangan yang ada pada metode *increment* ini berupa *release* produk (*Delivery of Increment* )

## 2.10. Keuntungan Model Increment

Penggunaan model *Increment* ini memiliki beberapa keuntungan antara lain sebagai berikut [27].

1. Terdapat pembagian produk sehingga sehingga biaya yang diperlukan bisa optimal (tidak lebih atau tidak kurang)
2. Model bisa digunakan pada saat sumber daya tim pengembang terbatas. Hal ini dikarenakan, model ini dibagi menjadi beberapa bagian yang lebih kecil
3. Terdapat *feedback* pada setiap software release sehingga kebutuhan pengguna semakin jelas.
4. Meminimalisir resiko cacat/bug. Hal ini karena setiap realease dilakukan pengujian. Pada saat intergasi, juga dilakukan pengujian.

### 2.11. Kekurangan Model *Increment*

Pada setiap model pasti terdapat kerugian. Begitupun juga dengan model *increment* ini. Model ini memiliki beberapa kerugian antara lain sebagai berikut.

1. Rentan terhadap kelebihan waktu (overtime). Hal ini karena terjadi proses perulangan-perulangan yang meningkat.
2. Memerlukan jadwal perencanaan dengan baik agar pembagian dan perulangan dilakukan dengan baik sehingga project dapat selesai tepat waktu sesuai dengan definisi lebih awal.

### 2.12. Kesesuaian Metodologi Untuk Tugas Akhir

Berdasarkan studi leteratur yang peneliti lakukan, dalam pengembangan perangkat lunak ini peneliti menggunakan model *Increment* . Adapaun yang menjadi alasan dari peneliti dalam pgunakan model *increment* ini ada pada tabel berikut.

**Tabel 2.8 Kesesuaian Metodologi Model Increment**

Indikator	Karakteristik kebutuhan pembuatan aplikasi UCP	Cheklist kesesuaian model <i>increment</i>
Kebutuhan akan Quality	Aplikasi yang dihasilkan memerlukan pengujian	√

<b>Indikator</b>	<b>Karakteristik kebutuhan pembuatan aplikasi UCP</b>	<b>Ceklist kesesuaian model <i>increment</i></b>
Kejelasan kebutuhan pengguna	Dapat berubah sewaktu-waktu terutama dalam hal antar muka/desain	√
Komunikasi dengan Client	Client mudah ditemui dan cenderung face to face	√
Jumlah pengembang	Pengembang untuk aplikasi terbatas.	√
Penggunaan teknologi	Teknologi yang dikembangkan tidak terlalu rumit karena berbasis web PHP biasa	√
Scope Aplikasi	Scope aplikasi tidak terlalu luas.	√

### 2.13. Unified Modeling Language (UML)

UML merupakan standard bahasa untuk digunakan untuk memvisualkan, menspesifikasi, mengkonstruksi, dan mendokumentasikan artefak dari sebuah sistem atau perangkat lunak [29]. Dalam memvisualisasikan sebuah UML, digunakan diagram standar yang ada di UML. UML ini digunakan dalam pemodelan desain program yang berorientasi object (OOP) serta aplikasinya [30].

### 2.14. Diagram pada UML

Berikut ini beberapa diagram standard yang digunakan dalam UML [31].

- *use case diagram*
- *conceptual diagram*
- *Class Diagram*
- *statechart diagram*
- *object diagram*
- *activity diagram*
- *sequence diagram*
- *collaboration diagram*

- *component diagram*
- *deployment diagram*

Dalam desain dengan menggunakan UML, tidak semua diagram diimplementasikan. Adapun empat diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi object yaitu: *use case diagram*, *sequence diagram*, *collaboration diagram*, dan *Class Diagram* [30].

## 2.15. Tahapan Desain Dengan UML

Menurut Sri,2013 [31]. Terdapat tips langkah-langkah dalam pengembangan perangkat lunak dengan menggunakan UML sebagai berikut.

1. Membuat daftar *business process* 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 pendefinisikan *requirement* lain seperti non-functional requirement 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 alir pekerjaan
7. Membuat rancangan user interface model yang menyediakan antarmuka bagi pengguna untuk menjalankan scenario use case
8. Berdasarkan model yang ada, buatlah *Class Diagram*. Setiap package dipecah sehingga menjadi hierarki class yang langkah dengan method dan atributnya.
9. Melakukan pengelompokan kelas menjadi komponen-komponen.

10. Memperhalus diagram deploymen yang sudah dibuat. Misalonya dengan pendetailan requirement perangkat lunak, sistem operasi, dan sebagainya. Lakukan pemetaan kedalam node
11. Melakukan pembangunan sistem berdasarkan hasil model yang sudah dibuat.

### 2.16. Use Case

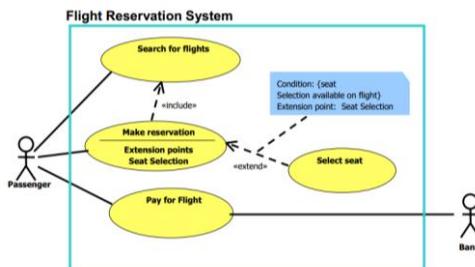
Use case merupakan deskripsi tertulis mengenai bagaimana sebuah actor melakukan suatu tugas dalam sebuah aplikasi. Use case merepresentasikan tahapan yang simple, dimulai dengan user's goal dan diakhiri ketika tujuan tercapai. [32] Use case sendiri dibagi menjadi 2. Pertama, yaitu *primary course*. Pada bagian ini menjelaskan bagaimana respon system untuk menstimulus pengguna dan asumsi system berjalan lancar. Kedua, yaitu *alternate course*, dalam *alternate course* ini, menjelaskan scenario apabila scenario utama/ main scenario belum tidak dilakukan/gagal. Alistair Cockburn, dalam bukunya yang berjudul "WRITING EFFECTIVE USE CASES", menjelaskan bahwa use case memiliki beberapa tingkatan/leveling. Adapun tingkatan yang ada dalam use case sebagai berikut [33].

1. Cloud ( Summary goal)  
Level *Cloud* ini merupakan level yang paling tinggi, melibatkan banyak tujuan pengguna (Users's goal). Contohnya Sell books online
2. Kite (Summary goal)  
Kite ini merupakan tahapan yang tinggi juga, dalam tahapan ini terjadi proses yang berlangsung beberapa jam, beberapa hari, atau bahkan beberapa minggu dan melibatkan banyak langkah
3. Sea(User Goals)  
Merupakan level dalam User Goal. Dalam level ini, actor melakukan sesuatu yang harus diselesaikan. Melibatkan beberapa level dibawahnya, yaitu underwater dan clam level.

4. Underwater(Subfunction)  
Dibutuhkan untuk menyelesaikan tujuan pengguna user goals. Misalnya “Save as a File”

## 2.17. Use Case Diagram

*Use Case diagram* merupakan diagram yang menggambarkan interaksi yang terjadi Antara element yang ada dalam sistem. Use case sendiri merupakan sebuah methodology yang digunakan untuk melakukan identifikasi, mengklarifikasi, dan mengorganisasi sebuah kebutuhan sistem [34]. Antara *Use Case* yang satu dengan stu dengan *Use Case* lainnya memiliki hubungan saling ketrgantungan (*dependency arrow*). Berikut ini adalah bentuk dari *Use Case* diagram yang ditunjukkan pada Gambar 2.6.

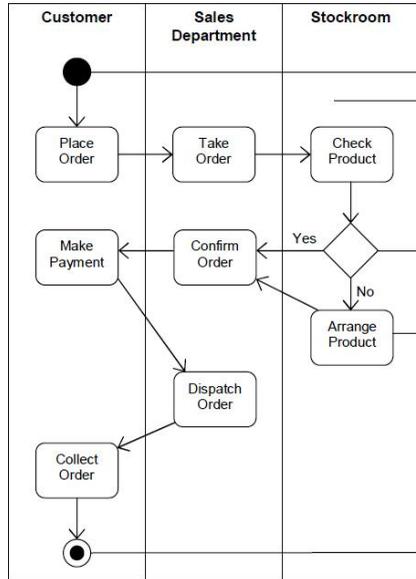


Gambar 2.6 Contoh use case diagram

## 2.18. Activity Diagram

*Activity diagram* merupakan diagram yang merepresentasikan alur/jalannya nalar dari sebuah use case *Use case* yang dibuat pada *Use case diagram*. Activity diagram ditunjukkan dengan sebuah box yang mendeskripsikan sebuah aktivitas. Antara box satu dengan box yang lainnya dihubungkan dengan sebuah garis panah yang menunjukkan tahapan yang alur yang berjalan. Dalam activity diagram, terdapat symbol titik yang merepresentasikan inisial (awal) dan akhir (stop) dari sebuah aktivitas [35]. Tujuan

dari pembuatan aktivitas diagram ini yaitu untuk memudahkan IT Bussiness Analyst untuk menggambarkan alur yang mudah dipahami oleh pengguna akhir (*end-user*) [36]. Contoh dari *activity diagram* terlihat pada Gambar 2.7. Dalam *sequence diagram* juga terdapat decision yang bertujuan untuk menggambarkan suatu kondisi (*if*). Kondisi tersebut memiliki 2 nilai. Nilai benar dan nilai salah.



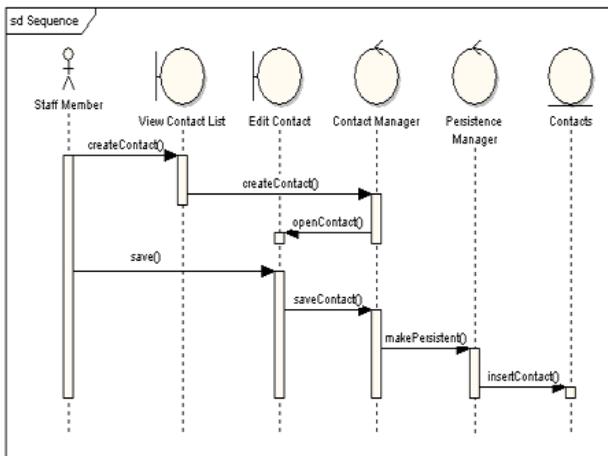
**Gambar 2.7** Contoh dari *activity diagram* [36].

## 2.19. *Sequence Diagram*

*Sequence diagram* merupakan diagram yang menggambarkan hubungan antar 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 *scenario use case*. *Sequence diagram* digunakan dalam tahapan analisa dan desain [37]. Adapun kegunaan dari *sequence diagram* ini yaitu untuk memodelkan:

- Deskripsi dari sebuah sistem yang digambarkan dalam diagram perilaku (*Use Case diagram*) kedalam bentuk hubungan antar actor dan Use Case Diagram
- Logika dari sebuah metode yang ada dalam suatu fungsi, baik dalam bentuk prosedur, operasi, atau fungsi)
- Logika dari sebuah layanan/service.

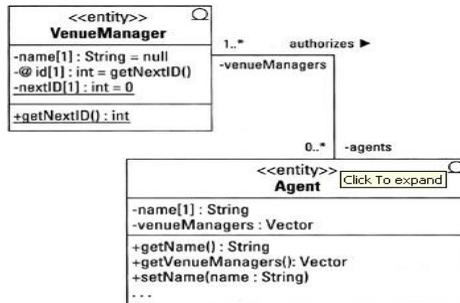
Salah satu contoh dari Sequence diagram terlihat pada Gambar2.8.



**Gambar 2.8 Contoh Sequence Diagram**

## 2.20. Class Diagram

*Class Diagram* merupakan grafik yang mengGambar kan koleksi dari deklaratif model dalam sebuah elemen [35]. *Class Diagram* menyediakan ringkasan dari sistem target dengan mendeskripsikan objek dan kelas kedalam sistem dan hubungannya diantara keduanya. *Class Diagram* memodelkan sumber daya masing - masing dalam hal struktur, hubungan, dan perilaku. tiga standar kompartemen ditemukan di hampir setiap kelas: nama, atribut,dan operasi. Adapun contoh *Class Diagram* ada pada Gambar 2.9 berikut ini.



Gambar 2.9 Contoh Class Diagram

## 2.21. Pengujian

Menurut Hetzel 1983, pengujian merupakan suatu aktivitas yang digunakan untuk melakukan evaluasi suatu atribut atau fitur dalam suatu program atau sistem untuk memastikan apakah sistem atau program telah memenuhi kebutuhan atau hasil yang diharapkan [38]. Adapun tiga aktivitas utama dalam pengujian yaitu sebagai berikut [39].

### 1. Verification

Verifikasi (*Verification*) merupakan pengujian untuk memastikan bahwa hasil perancangan sesuai dengan kebutuhan pengguna.

### 2. Validation

Validasi (*Validation*) merupakan pengujian yang dilakukan untuk memastikan bahwa sistem dibuat sesuai dengan perancangan atau design yang sudah dibuat sebelumnya.

### 3. Bug Error Finding

Merupakan pengujian yang dilakukan untuk mengetahui error dari aplikasi

## 2.22. Metode Uji Coba Black Box (*Blackbox Testing*)

Uji coba black box ini melibatkan pengecekan unit-unit/fungsi-fungsi dalam perangkat lunak dengan memberikan perlakuan

masukannya tertentu tanpa mengetahui proses internal (*Logic*) yang ada dalam perangkat lunak/sistem tersebut [40]. Untuk melakukan pengujian ini, pihak pengetes (*Tester*) tidak perlu tahu struktur dari program. Pengujian langsung melakukan pengujian dengan menginputkan sesuai dengan yang diperintahkan oleh sistem dan juga menginputkan asumsi dari pengujian dengan acuan list test case.

### **2.23. Pengujian Antarmuka Dengan Aturan Shneiderman**

Selain pengujian fungsional, terdapat juga pengujian untuk antarmuka (interface aplikasi). Pengujian interface aplikasi dapat dilakukan salah satunya dengan menggunakan aturan yang dikemukakan oleh Ben Shneiderman yang dikenal dengan “*Eight Golden Rules of Interface Design*” [41]. Adapun 8 aturan yang ada sebagai berikut.

#### *1. Strive for Consistency*

Desain antarmuka aplikasi harus memiliki konsistensi dalam tampilan antarmuka aplikasi

#### *2. Cater to Universal Usability*

Aturan ini mengenai perubahan konten yang dikhususkan ke pengguna

#### *3. Offer Information Feedback*

Pada aturan ketiga ini, memerintahkan bahwa aplikasi mempunyai fitur untuk pesan feedback dari aplikasi ke pengguna secara informatif.

#### *4. Design Dialogs to Yield Closure*

Aturan keempat ini mengharuskan pada aplikasi untuk menyediakan informasi yang menjelaskan alur/proses yang ada dalam aplikasi

#### 5. *Prevent Errors*

Aturan ini menjelaskan tentang fitur untuk mencegah agar pengguna tidak melakukan kesalahan dalam penggunaan aplikasi

#### 6. *Permit Easy reversal of Action*

Pada aturan keenam ini, aplikasi harus memiliki fitur untuk memberikan kemudahan bagi pengguna dalam kembali ke langkah sebelumnya.

#### 7. *Support Internal Locus of Control*

Aplikasi harus menjadikan pengguna sebagai kontrol terhadap aplikasi. Bukan sebaliknya.

#### 8. *Reduce Short Term Memory*

Aplikasi harus memiliki tampilan yang sederhana dan mudah dipahami oleh pengguna

### 2.24. **PHP (*Hypertext Preprocessor*)**

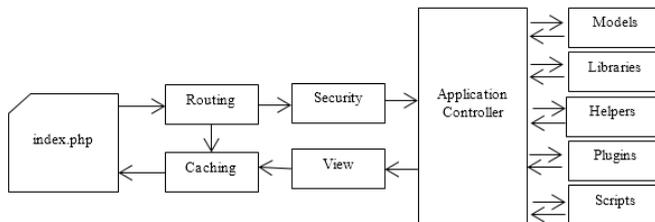
PHP merupakan Bahasa pemrograman berbasis web yang diciptakan oleh Rasmus Lerdorf pada tahun 1995 [42]. Awalnya PHP mempunyai kepanjangan kata yaitu Personal Home Page. Pemrograman ini sudah banyak digunakan untuk pembuatan aplikasi atau system yang berbasis website. Hal ini dibuktikan dari sebuah penelitian yang dilakukan pada tahun 1997 yang menyatakan bahwa bahasa pemrograman PHP sudah digunakan hampir 500.000 lebih website di dunia [43]. Keuntungan dari Bahasa pemrograman PHP ini yaitu sebagai berikut.

- a. Bahasa pemrograman PHP memberikan kemudahan kepada developer dalam hal melakukan proses rilis dan proses update perangkat lunak yang lebih cepat.
- b. Bahasa pemrograman PHP memiliki *library* yang sudah tersedia didalamnya (*built-in*) [44]

- c. Bahasa pemrograman PHP merupakan bahasa pemrograman yang Open Source sehingga memberikan fitur-fitur yang luas dan lengkap tanpa ada pembatasan.

## 2.25. Codeigniter (CI)

Codeigniter merupakan framework bahasa pemrograman berbasis dengan PHP yang direlease pertama kali pada tahun 2006 oleh EllisLab, Inc [45]. Gambar 2.10 berikut ini adalah bentuk MVC yang ada di CodeIgniter



**Gambar 2.10** Sistem Kerja Codeigniter (Ardhana, 2013)

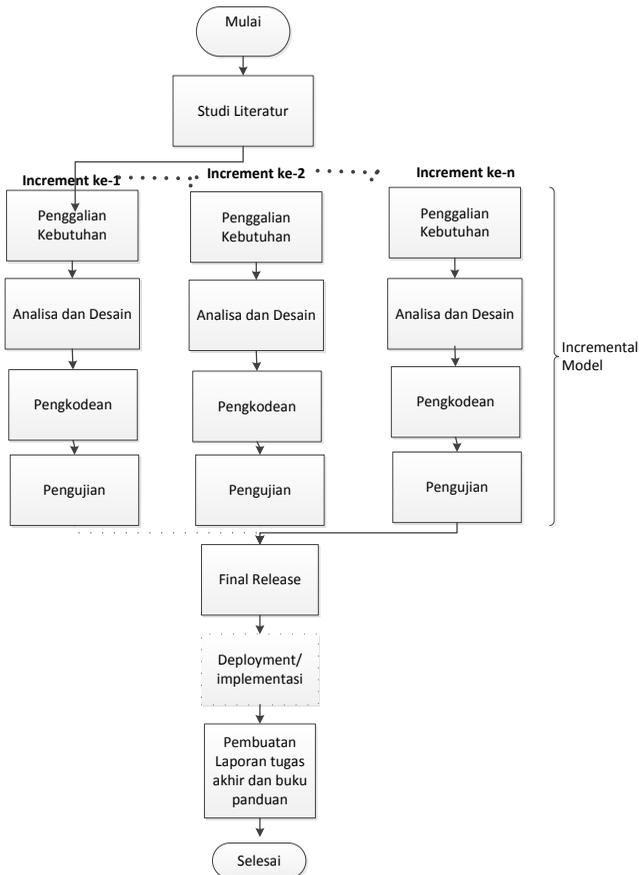
Untuk mengembangkan framework ini cukup mudah. Hal ini dikarenakan dalam framework ini, disediakan dokumentasi yang lengkap yang bias diakses pada website ellislab.com. Konsep logika dalam Codeigniter ini menggunakan konsep MVC (*Model, View, Controller*). Berikut ini adalah penjelasan detail dari ketiga bagian tersebut [46].

- *Model*, bagian ini berisikan kode-kode yang digunakan untuk mengakses database
- *View*, bagian ini berisikan kode-kode yang berkaitan dengan tampilan, yaitu kode-kode terkait dengan HTML dan PHP yang ditampilkan pada layar browser
- *Controller*, berisikan kode-kode yang digunakan untuk menghubungkan antara *View* dengan *Model* dan juga sumber daya yang dibutuhkan untuk melakukan proses permintaan layanan HTTP untuk ditampilkan kedalam halaman website.

*"Halaman ini sengaja dikosongkan"*

# BAB III METODOLOGI PENELITIAN

Metode pengerjaan tugas akhir diperlukan sebagai panduan agar pengerjaan tugas akhir menjadi terarah dan jelas. Gambar berikut ini adalah alur dari pengerjaan tugas akhir.



Gambar 3.1 Metodologi Penelitian

*“Halaman ini sengaja dikosongkan”*

Pada pengerjaan tugas akhir ini digunakan alur tahapan pengerjaan sesuai dengan tahapan model *Increment* . Adapun detail dari alur pengerjaan tugas akhir ini adalah pada tabel berikut ini.

**Tabel 3.1 Detail Alur Pengerjaan Tugas Akhir**

No.	Aktivitas	Sub-Aktivitas	Input	Teknik/tools	Output	Tujuan
1.	Studi Literatur	-	<ul style="list-style-type: none"> <li>• Latar Belakang</li> <li>• Permasalahan</li> <li>• Tujuan penelitian</li> </ul>	Penggumpulan data melalui journal, internet, dan buku literatur terkait	<ul style="list-style-type: none"> <li>• Informasi terkait UCP</li> <li>• Informasi metode pengembangan perangkat lunak</li> <li>• List pertanyaan wawancara</li> <li>• Informasi UML</li> <li>• Framework Codeigniter (CI).</li> </ul>	Memahami teori tentang SDLC, model <i>Increment</i> , Bahasa Pemograman PHP, Framework CodeIgniter, dan mengetahui informasi terkait metode UCP
2.	Penggalian kebutuhan	-	List pertanyaan yang digunakan untuk melakukan wawancara	Melakukan wawancara dengan client	<ul style="list-style-type: none"> <li>• Dokumen hasil wawancara (interview result),</li> </ul>	Mendapatkan data dan informasi untuk bahan perancangan dan pembagian

No.	Aktivitas	Sub-Aktivitas	Input	Teknik/tools	Output	Tujuan
					<ul style="list-style-type: none"> <li>Informasi proses proses bisnis</li> </ul>	aplikasi.
3.	Analisa dan Desain	Identifikasi Kebutuhan	<ul style="list-style-type: none"> <li>Informasi Estimasi Biaya dengan UCP</li> <li>Informasi proses bisnis dari Perusahaan DTS</li> </ul>	<i>Business Process Modelling</i> dengan <i>Visio</i>	Model proses bisnis	Melakukan identifikasi kebutuhan sebagai bahan dalam melakukan percangan dan desain aplikasi UCP
4.		Identifikasi Desain	Model proses bisnis	Menggunakan tools Enterprise Architect versi 10	<i>Use case diagram, sequence diagram, Class Diagram, desain database, desain antarmuka</i>	Memudahkan pengembang dalam melakukan analisis kebutuhan dan pembangunan aplikasi UCP.
5.		Menyusun SKPL dan	<ul style="list-style-type: none"> <li>Model proses bisnis</li> </ul>	Microsoft Word 2013	<ul style="list-style-type: none"> <li>Dokumen SKPL</li> <li>Dokumen DPPL</li> </ul>	Mendokumentasikan kebutuhan

No.	Aktivitas	Sub-Aktivitas	Input	Teknik/tools	Output	Tujuan
		dokumen Desain	<ul style="list-style-type: none"> <li>• <i>Use case diagram</i></li> <li>• <i>Sequence diagram</i></li> <li>• <i>Class Diagram</i></li> <li>• Desain database</li> <li>• Desain antarmuka</li> </ul>	GUI desain studio Power Designer	<ul style="list-style-type: none"> <li>• Dokumen Testing</li> </ul>	merancang Aplikasi UCP, dan merancang dokumentasi pengujian aplikasi sesuai dengan tahapan yang ada.
6.	Pembangunan Aplikasi		<ul style="list-style-type: none"> <li>• Dokumen SKPL</li> <li>• Dokumen DPPL</li> </ul>	SDLC, Model <i>Increment</i> , Notepad++, XAMPP, Navicat	Aplikasi Use Case Point (UCP) yang belum di uji.	Menghasilkan Aplikasi UCP yang belum teruji
7.	Pengujian Aplikasi		Aplikasi UCP yang belum teruji	<i>Black Box Testing</i> <i>JMeter Testing</i>	<ul style="list-style-type: none"> <li>• Release aplikasi/fitur aplikasi yang sudah teruji</li> <li>• Laporan Hasil Pengujian</li> </ul>	Memastikan fungsionalitas pada Aplikasi UCP berjalan dengan baik dan bebas dari bug

No.	Aktivitas	Sub-Aktivitas	Input	Teknik/tools	Output	Tujuan
8.	Deployment (Implementasi)	-	Aplikasi yang sudah diuji	Melakukan pemasangan aplikasi di server	Aplikasi yang sudah terpasang di server	Melakukan implementasi aplikasi UCP untuk dijalankan di perusahaan DTS
9	Final Release		Release aplikasi yang sudah teruji dengan semua fitur yang sesuai dengan kebutuhan client	Notepad++, XAMPP, Navicat	Aplikasi UCP dengan fitur yang sudah lengkap	Melakukan penggabungan fitur-fitur/Sub-Aplikasi yang sudah teruji menjadi satu bagian aplikasi yang lengkap sesuai dengan permintaan client.
10.	Menyusun Tugas Akhir		Dokumentasi pembangunan Aplikasi dan dokumen pengujian	Microsoft Word 2013	Laporan Tugas Akhir	Mendokumentasikan analisis hasil spesifikasi formal pada Aplikasi UCP

## **BAB IV PERANCANGAN**

Pada bagian ini, akan dijelaskan tentang tahapan wawancara Perancangan tahapan *increment 1*, perancangan tahap *increment 2*, dan perancangan tahap *increment 3* dalam pembuatan aplikasi estimasi harga pengembangan perangkat lunak menggunakan metode use case point studi kasus Perusahaan DTS.

### **4.1. Tahapan Perancangan**

Tahapan perancangan merupakan tahapan yang dilalui oleh peneliti dalam membuat aplikasi UCP pada studi kasus perusahaan DTS. Berikut ini merupakan tahapan yang dilakukan dalam perancangan aplikasi UCP.

#### **4.1.1. Melakukan Wawancara**

Sebelum membangun aplikasi, peneliti melakukan wawancara dengan perwakilan pihak Perusahaan DTS. Wawancara merupakan salah satu *tools* untuk melakukan penggalan kebutuhan. Sebelum melakukan wawancara ini, peneliti melakukan tahapan sebagai berikut.

1. Memilih target wawancara  
Pada tahap ini, peneliti melakukan penentuan siapa interviewee yang akan peneliti tentukan untuk mengetahui informasi proses bisnis dan kebutuhan apa saja yang diperlukan dalam pembuatan aplikasi. Adapun target interviewee dalam penggalan kebutuhan ini yaitu pemilik/direktur dari Perusahaan DTS.
2. Mendesain pertanyaan-pertanyaan untuk wawancara  
Setelah menentukan target wawancara, langkah selanjutnya yang peneliti lakukan yaitu melakukan desain pertanyaan untuk digunakan dalam melakukan wawancara. Tipe wawancara yang dilakukan yaitu wawancara semi-

terstruktur (*Semi-structured interviews*). *Semi-structured interviews* merupakan jenis interview dengan menggunakan pertanyaan awal yang sudah dibuat dan juga pertanyaan spontanitas untuk mengetahui lebih dalam ketertraikan dari *interviewee* [47]. Adapun desain pertanyaan awal untuk wawancara ada di bagian lampiran 1.

3. Persiapan wawancara  
Pada tahapan persiapan wawancara ini, peneliti menentukan jadwal dan tempat wawancara. Untuk jadwal dan tempat wawancara ada pada bagian lampiran
4. Melakukan wawancara  
Pada tahapan ini, peneliti melakukan wawancara dengan client.

#### **4.1.2. Mendapatkan Kebutuhan Pengguna**

Setelah melakukan wawancara, tahapan selanjutnya yang dilakukan oleh peneliti yaitu melakukan analisa untuk mendaptkan kebutuhan pengguna. Kebutuhan pengguna didapatkan dari hasil wawancara (interview result) yang ditranslasikan kedalam suatu bentuk kebutuhan. Kebutuhan yang didapatkan yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

#### **4.1.3. Pemodelan Proses Bisnis Aplikasi**

Setelah dilakukan tahapan penggalian kebuhan dengan melakukan wawancara, peneliti melakukan pemodelan proses bisnis aplikasi UCP. Pemodelan proses bisnis dilakukan dengan membuat diagram aktivitas. Terdapat dua diagram aktivitas yang ada.

Pertama, diagram aktivitas yang menunjukkan proses/alur untuk melakukan estimasi usaha dengan menggunakan metode Use Case Point. Diagram aktivitas ini digunakan sebagai acuan dalam pengembangan aplikasi pada tahap *increment* 1

Kedua, diagram aktivitas yang menunjukkan proses/alur estimasi biaya perangkat lunak yang dikombinasikan dengan alur/proses estimasi biaya yang ada pada perusahaan DTS. Diagram aktivitas ini digunakan sebagai acuan dalam pengembangan aplikasi pada tahap *increment 2*.

#### **4.1.4. Melakukan Desain Use Case**

Desain *use case* dibuat dari hasil analisa kebutuhan dan bentuk desain yang ada pada aplikasi UCP. Use case ini menunjukkan interaksi antara aktor atau pengguna aplikasi dengan aplikasi. Selain pembuatan desain, juga dilakukan pembuatan deskripsi dari masing-masing *use case* yang ada. Deskripsi use case (*Use Case Description*) berisikan detail behavior dari sebuah use case.

#### **4.1.5. Desain Sequence Diagram**

*Use Case Description*(UCD) yang sudah dibuat akan dijadikan acuan dalam pembuatan *Sequence Diagram*(SQ). *Sequence Diagram* merupakan diagram yang menggambarkan kan hubungan antar objek dalam waktu yang berurutan untuk memenuhi permintaan dari pengguna yang dijelaskan pada use case diagram. Pada *Squence diagram* ini juga terdapat method/fungsi untuk melakukan proses logic yang ada pada aplikasi.

#### **4.1.6. Desain Class Diagram**

*Class Diagram* merupakan grafik yang mengGambar kan koleksi dari deklaratif object dalam sebuah elemen dan hubungan antar masing-masing object tersebut [35]. Masing-masing object yang berisikan *method* pada *Sequence Diagram*(SQ) yang sudah dibuat akan dirubah kedalam bentuk class pada *Class Diagram*.

#### **4.1.7. Pembuatan Model Data Konseptual**

Model Data Konseptual (CDM) digunakan sebagai acuan dalam pembuatan desain database untuk aplikasi. Pembuatan model data konseptual ini berdasarkan object entitas pada kelas diagram.

Konseptual model data ini menggambarkan hubungan antara entitas satu dengan entitas lainnya.

#### **4.1.8. Pembuatan Model Data Fisik**

Model data fisik (*Physical Data Model*) merupakan tahapan akhir dari proses desain data model. Model data fisik ini tidak hanya menggambarkan entitas, atribut, dan hubungan antar entitas, tapi juga menunjukkan perilaku yang ada pada masing-masing entitas. Selain itu, pada pembuatan model data fisik ini, pada setiap atribut terdapat tipe dari atribut sendiri dan juga panjang/ukuran pada masing-masing tipe. Model data fisik yang sudah dibuat akan *degenerate* langsung kedalam bentuk database spesifik.

#### **4.1.9. Melakukan Desain Antarmuka**

Desain antar muka merupakan bentuk desain *User Interface* (UI) yang ada pada aplikasi UCP. Desain antar muka dibuat bersarkan daftar kebutuhan yang sudah disepakati dengan *client*.

#### **4.1.10. Perancangan Pengujian**

Perancangan Pengujian dibuat berdasarkan fitur yang ada dalam aplikasi. Pada setiap increment, dilakukan perancangan pengujian dalam bentuk perencanaan dan prosedur pengujian untuk kebutuhan fungsional dan kebutuhan non-fungsional pada aplikasi UCP. Adapun detail dari perancangan pengujian ini ada pada dokumen pengujian.

### **4.2. Perancangan Tahap *Increment* 1**

#### **4.2.1. Penggalan Kebutuhan**

Setelah peneliti melakukan wawancara, tahapan yang peneliti lakukan yaitu menganalisa hasil wawancara menjadi sebuah kebutuhan dalam pembuatan aplikasi estimasi harga. Pada tahapan ini, peneliti membuat daftar kebutuhan aplikasi berdasarkan hasil translasi wawancara kedalam inti dari hasil wawancara untuk kemudian membuat daftar kebutuhan aplikasi,

mendesripsikan perangkat lunak yang akan dibuat dan melakukan klasifikasi pengguna dari perangkat lunak yang akan dibuat.

a. Translasi Hasil Wawancara Kedalam Kebutuhan

Tabel berikut menunjukkan hasil translasi kebutuhan yang dilakukan.

**Tabel 4.1 Translasi kebutuhan**

<b>Kode</b>	<b>Point of Interview</b>	<b>Requirement</b>
PI-01-01	<p><b>Aplikasi mudah diakses dimanapun</b></p> <p>Aplikasi yang akan dibuat nanti harus mudah diakses oleh pengguna dari aplikasi ini, sehingga tidak harus datang ke kantor untuk menjalankan aplikasi ini</p> <p><b>Ref: IR-12</b></p>	<p>1. Aplikasi berbasis web</p>
PI-01-02	<p><b>Estimasi harga lebih mudah</b></p> <p>Tim pengembang bisa melakukan estimasi harga perangkat lunak lebih akurat lagi dari yang biasanya dilakukan</p> <p><b>Ref: IR-08, IR-12, IR-13</b></p>	<p>Estimasi harga perangkat lunak menggunakan metode Use Case point karena metode ini lebih akurat dibandingkan dengan metode-metode yang lain [3]. Jadi, fitur yang ada sebagai berikut.</p> <p>2. Fitur Perhitungan UUCW</p> <p>3. Fitur Perhitungan UAW</p> <p>4. Fitur Perhitungan TCF</p> <p>5. Fitur Perhitungan ECF</p> <p>Fitur perhitungan biaya</p>

Kode	Point of Interview	Requirement
		berdasarkan nilai UCP
PI-01-03	<p><b>Estimasi Aplikasi nanti hanya bisa digunakan oleh orang-orang tertentu</b></p> <p>Aplikasi nanti hanya bisa digunakan oleh orang-orang yang terlibat dalam estimasi harga perangkat lunak sesuai dengan proses bisnis yang ada di Perusahaan DTS ini. Selain itu, mereka tidak boleh mengakses. Boleh melakukan akses jika mendapatkan izin dari admin(Direktur).</p> <p><b>Ref: IR-12</b></p>	<p>Aplikasi yang ada nanti terdapat fitur:</p> <ul style="list-style-type: none"> <li>• Login/Logout</li> <li>• Fitur Manajemen pengguna</li> <li>• Fitur previlage pengguna</li> </ul>
PI-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 untui mengubah data estimasi.</p>	<p>Fitur CRUD (Create Update Delete) dari masing-masing fitur yang ada pada <b>PI-01-2</b>.</p>

b. Membuat Daftar Kebutuhan Aplikasi

Daftar kebutuhan aplikasi dibuat berdasarkan hasil analisa terhadap cerita pengguna (*User Story*). Daftar kebutuhan aplikasi ini nantinya akan dijadikan acuan dalam pembuatan dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Tabel 4-2

merupakan daftar kebutuhan yang ada pada aplikasi UCP yang sudah dikonfirmasi kepada *client*.

**Tabel 4.2 Daftar kebutuhan aplikasi UCP**

<b>Kode</b>	<b>Penjelasan</b>
KF01	Fitur Autentifikasi
KF02	Fitur deskripsi aplikasi
KF03	Aplikasi bisa melakukan perhitungan nilai UUCW
KF04	Aplikasi bisa melakukan perhitungan nilai UAW
KF05	Aplikasi bisa melakukan perhitungan nilai TCF
KF06	Aplikasi bisa melakukan perhitungan nilai ECF
KF07	Terdapat fitur untuk menampilkan rekap hasil perhitungan UCP

#### c. Deskripsi Umum Perangkat Lunak

Perangkat lunak yang dikembangkan adalah Perangkat Lunak untuk melakukan estimasi harga perangkat lunak menggunakan metode Use Case Point sesuai dengan proses estimasi perangkat lunak yang ada di Perusahaan DTS. Tujuan pembuatan perangkat lunak ini yaitu sebagai solusi dalam mengatasi permasalahan yang ada di Perusahaan DTS dalam hal melakukan estimasi biaya pembuatan perangkat lunak.

#### d. Karakteristik dan Klasifikasi Pengguna

Aplikasi estimasi harga perangkat lunak dengan menggunakan metode UCP ini penggunaan untuk Aplikasi ini digunakan oleh pengguna yang mempunyai peran sebagai Direktur Perusahaan mempunyai peran sebagai Direktur, Project Manager/Sistem Analis mempunyai peran sebagai Tim Pengembang, dan Sekretaris berperan sebagai Sekretaris. Hal ini didasarkan dari hasil wawancara yang sudah dilakukan antra pihak peneliti dengan perwakilan dari perusahaan DTS.

a. **Kebutuhan Fungsional Aplikasi**

Kebutuhan fungsional dari aplikasi estimasi harga perangkat lunak ini berdasarkan atas proses bisnis dari estimasi harga perangkat lunak dengan metode UCP yang sudah dijelaskan sebelumnya. Adapun kebutuhan fungsional dari aplikasi UCP ini tersaji dalam tabel 4.3 berikut.

**Tabel 4.3 Daftar Kebutuhan fungsional aplikasi**

<b>Kode Kebutuhan</b>	<b>Nama Kebutuhan</b>
KF01	Fitur Autentifikasi
KF02	Fitur deskripsi aplikasi
KF03	Aplikasi bisa melakukan perhitungan nilai UUCW
KF04	Fitur autentifikasi
KF05	Aplikasi bisa melakukan perhitungan nilai UAW
KF06	Aplikasi bisa melakukan perhitungan nilai TCF
KF07	Aplikasi bisa melakukan perhitungan nilai ECF
KF08	Terdapat fitur untuk menampilkan rekap hasil perhitungan UCP

b. **Kebutuhan Non Fungsionalitas Sistem**

Selain kebutuhan fungsional, dalam aplikasi estimasi harga ini juga terdapat kebutuhan-kebutuhan non fungsional. Adapun kebutuhan non fungsional dari aplikasi UCP ini adalah sebagai berikut.

1. **Kebutuhan kualitas aplikasi**

Kebutuhan akan kualitas dari aplikasi Use Case Point dilihat dari dua sisi, yaitu sisi: correctness dan portability.

a. *Correctness*

Coreectness dibutuhkan untuk emastikan bahwa setiap fitur dan kebutuhan fungsional dalam aplikasi UCP sudah

berjalan dengan baik dan benar sesuai dengan spesifikasi yang sudah didefinisikan dan disepakati dengan *client*

b. *Portability testing*

Portability merupakan kebutuhan yang perlu ada untuk membuat aplikasi bisa berjalan dalam browser yang berbeda.

Berikut ini daftar kebutuhan non-fungsional untuk kategori kualitas aplikasi pada tahap *increment 1*.

**Tabel 4.4 Daftar tools pengujian kualitas aplikasi**

Kode	Deskripsi Kebutuhan
<b><i>Correctness</i></b>	
KNF-01	Aplikasi telah berjalan sesuai dengan kebutuhan yang telah dispesifikasikan
<b><i>Usability</i></b>	
KNF-02	Rancangan antar muka aplikasi yang <i>user friendly</i>
<b><i>Portability</i></b>	
KNF-03	Aplikasi dapat dijalankan di browser yang berbeda

## 2. Kebutuhan Sistem

Aplikasi UCP ini bisa dijalankan dengan bantuan sebuah browser internet yang up to date dan Sistem Operasi yang ada pada komputer/laptop atau pada OS smartphone (KNF-04).

## 3. Kebutuhan Kinerja

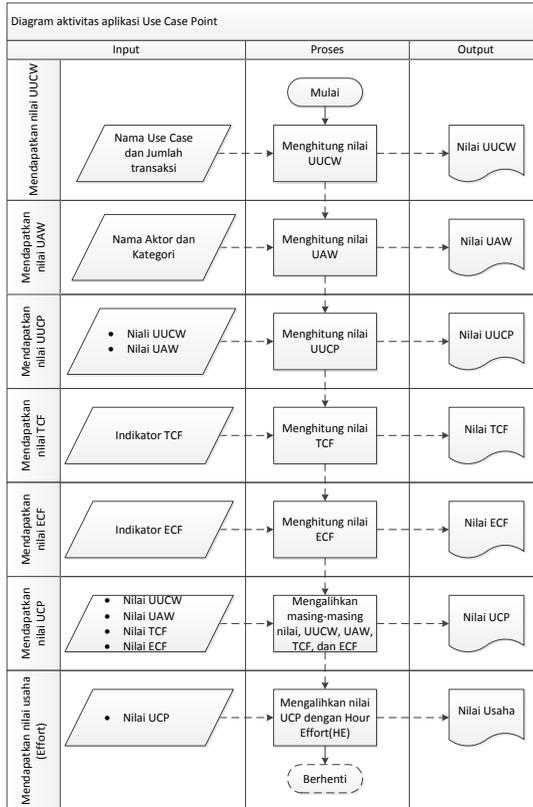
Aplikasi UCP dapat diakses di lebih dari satu komputer atau laptop dan Device Smartphone(KNF-05).

### 4.2.2. Analisa Kebutuhan Aplikasi

Analisa kebutuhan pada tahapan *increment 1* ini berfokus pada analisa kebutuhan mengenai estimasi harga perangkat lunak

dengan menggunakan metode Use Case Point(UCP). Analisa kebutuhan dilakukan berdasarkan hasil penggalian kebutuhan yang sudah dilakukan sebelumnya.

a. Alur Proses Estimasi dengan Metode Use Case Point (UCP)



**Gambar 4.1** Alur estimasi perhitungan usaha metode UCP

Gambar 4.1 adalah gambar alur proses estimasi usaha dengan Use Case Point yang sudah didiskusikan dengan peneliti dan juga perwakilan dari Perusahaan DTS sesuai dengan hasil interview

pada **IR-13**. Alur bisnis proses estimasi harga perangkat lunak dengan menggunakan metode use case point ini nantinya digunakan sebagai alur program/proses bisnis aplikasi Use Case Point pada tahap *increment* 1 ini.

Detail langkah yang ada dalam perhitungan usaha menggunakan metode ini sebagai berikut.

### 1. Menghitung nilai UUCW

Pertama kali yang akan dimasukan yaitu nama *use case* dan jumlah transaksi yang ada pada use case tersebut. Transaksi dalam use case ini diidentifikasi dalam penggunaan textual, yaitu dalam bentuk sebuah text. Use Case yang akan dilakukan pembobotan harus ditulis dalam pendekatan level yang sama, yaitu dalam level User-goals dari 5 level Use Case yang diidentifikasi oleh Alistair Cockburn [12]. Transaksi yang digunakan untuk menentukan kompleksitas dari use case ini merupakan scenario sukses dari sebuah use case [14]. Setelah didapatkan jumlah transaksi, dilakukan penentuan tipe/jenis dari *use case*. Tabel 4.5 merupakan daftar klasifikasi Use Case berdasarkan jumlah transaksi.

**Tabel 4.5 Daftar kalasifikasi use case**

<b>Tipe</b>	<b>Jumlah Transaksi</b>	<b>Bobot</b>
Simple	1 sampai 3 transaksi	5
Average	4 sampai 7 transaksi	10
Complex	8 transaksi atau lebih	15

Bobot dari *use case* didapatkan berdasarkan tipe dari *use case* tersebut. Aplikasi Use Case Point nanti mendapatkan nilai bobot berdasarkan jumlah transaksi yang dimasukan oleh pengguna. Pengguna awalnya memasukan nama use case dan jumlah transaksi. Kemudian sistem akan melakukan

pengkategorian transaksi secara otomatis kemudian melakukan perhitungan untuk mendapatkan nilai UUCW. Nilai UUCW didapatkan dengan cara menjumlahkan bobot dari semua *use case* yang ada dalam aplikasi.

## 2. Menghitung nilai UAW

Setelah nilai UUCW didapatkan, tahapan selanjutnya yaitu menghitung nilai UAW. Untuk menghitung nilai UAW ini, input yang dimasukan yaitu nama aktor dan kategori kompleksitas aktor. Kompleksitas aktor sendiri ada 3 macam, yaitu kategori Simple, Average, dan Complex. Ketiga kategori tersebut, masing-masing mempunyai bobot. Tabel 4.6 berikut ini tabel klasifikasi pembobotan aktor.

**Tabel 4.6 Klasifikasi pembobotan aktor**

<b>Tipe</b>	<b>Jumlah Transaksi</b>	<b>Bobot</b>
Simple	1 sampai 3 transaksi	5
Average	4 sampai 7 transaksi	10
Complex	8 transaksi atau lebih	15

Setiap Aktor yang ada dalam aplikasi dimasukan kedalam salah satu dari ketiga kategori yang ada. Setelah kategori aktor dilihat, akan didapatkan bobot dari aktor tersebut. Nilai UAW didapatkan dengan cara menjumlahkan semua bobot aktor yang ada ada dalam aplikasi tersebut.

## 3. Menghitung nilai UUCP

Nilai UUCP didapatkan dari perhitungan nilai UAW dan UUCW. Nilai UUCW dan nilai UAW dijumlahkan sehingga didapatkan nilai UUCP.

## 4. Menghitung nilai TCF

Setelah didapatkan nilai UUCW, langkah selanjutnya yang dilakukan yaitu mendapatkan nilai TCF. Untuk mendapatkan nilai TCF ini, inputnya berupa pilihan indikator yang ada dalam TCF. Nilai TCF didapatkan dengan menggunakan rumus sebagai berikut

$$\text{TCF} = 0,6 + (0,01 * \text{TF})$$

Nilai TF sendiri didapatkan dengan cara menjumlahkan bobot dari semua indikator yang sudah dipilih. Tabel 4.7 berikut ini adalah tabel pembobotan dari masing-masing indikator.

**Tabel 4.7 Pembobotan masing-masing indikator TCF**

Code	Indicator	Weight
T1	Distributed System Required	2.0
T2	Response Time Is Important	1.0
T3	End User Efficiency	1.0
T4	Complex Internal Processing Required	1.0
T5	Reusable Code Must Be A Focus	1.0
T6	Installation Easy	0.5
T7	Usability	0.5
T8	Cross-Platform Support	2.0
T9	Easy To Change	1.0
T10	Highly Concurrent	1.0
T11	Custom Security	1.0
T12	Dependence On Thrid-Part Code	1.0
T13	User Training	1.0

##### 5. Menghitung nilai ECF

Setelah Nilai TCF didapatkan, tahapan yang dilakukan selanjutnya yaitu mendapatkan nilai ECF. Untuk mendapatkan nilai ECF ini, inputnya berupa pilihan indikator yang ada dalam

ECF. Masing-masing indikator yang ada memiliki bobot. Sebelum mendapatkan nilai ECF, perlu dilakukan perhitungan nilai EF terlebih dahulu. Nilai EF didapatkan dengan menjumlahkan semua bobot yang ada pada masing-masing indikator yang dipilih. Adapun daftar indikator beserta bobot terdapat pada tabel 4.4 berikut ini.

**Tabel 4.8 Daftar indikator beserta bobot**

Code	Indicator	Weight
E1	Familiarity with the Project	1.5
E2	Application Experience	0.5
E3	OO Programming Experience	1.0
E4	Lead Analyst Capability	0.5
E5	Motivation	1.0
E6	Stable Requirements	2.0
E7	Part Time Staff	-1.0
E8	Difficult Programming Language	-1.0

Nilai ECF didapatkan dengan menggunakan rumus sebagai berikut

$$\text{ECF} = 1.4 + (-0.03 \times \text{EF})$$

#### 6. Mendapatkan nilai UCP

Input yang diperlukan untuk menghitung nilai UCP ini yaitu: nilai UUCW, nilai UAW, nilai TCF, dan nilai ECF. Nilai UCP didapatkan dengan mengalihkan masing-masing nilai tersebut. Berikut ini adalah rumus perhitungan nilai UCP.

$$\text{UCP} = \text{UUCP} * \text{TCF} * \text{ECF}$$

#### 7. Mendapatkan nilai usaha (*Effort*)

Nilai UCP digunakan sebagai *input* untuk menghitung nilai usaha. Nilai usaha didapatkan dengan cara mengalihkan nilai UCP dengan konstantan *Effort (Effort Rate)*. Nilai konstanta *Effort* yang akan digunakan dalam pembuatan aplikasi ini yaitu nilai konstantang yang dikemukakan oleh Gustav kernel, yaitu sebesar 20 setiap UCP. Keluaran dalam penelitian ini yaitu berupa nilai usaha. Berikut ini adalah rumus perhitungan untuk mendapatkan nilai usaha.

$$\text{USAHA} = \text{UCP} * 20$$

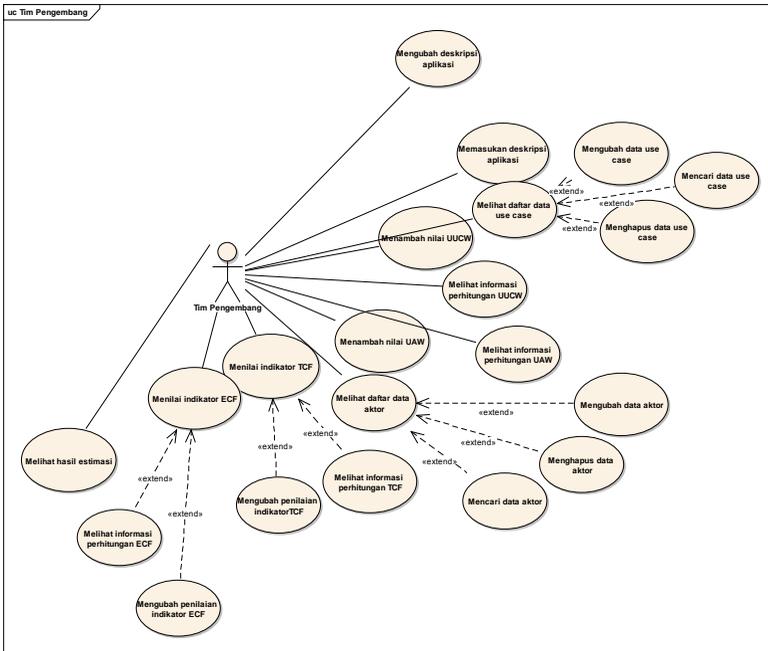
#### b. Lingkungan Operasi

Aplikasi estimasi harga perangkat lunak menggunakan metode UCP ini berbasis web dan dijalankan secara online. Aplikasi ini tidak memerlukan spesifikasi yang tinggi. Bisa diakses menggunakan browser internet yang *up to date*.

### 4.2.3. Desain Use Case Diagram

#### a. Keseluruhan Desain bagian depan (*Frontend*)

Pada tahapan *increment 1*, pembuatan use case didasarkan atas proses bisnis estimasi usaha dengan menggunakan metode *Use Case Point* (UCP) yang didapatkan dari hasil analisa kebutuhan. Gambar 4.2 merupakan desain *use case* pada aktor pengembang. Pada use case tersebut terlihat bahwa terdapat beberapa use case yang ada. Use Case yang ada pada tahap *increment 1* ini digunakan oleh aktor tim pengembang. Use case diagram yang ada tersebut kemudian dilakukan pengkategorian berdasarkan fitur yang ada pada use case.

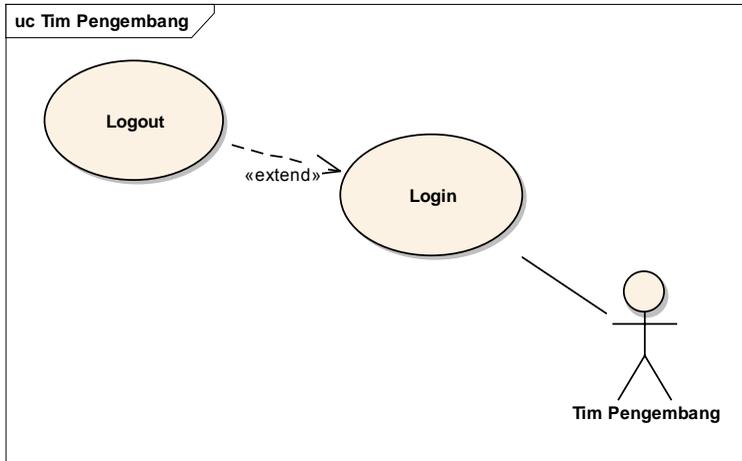


**Gambar 4.2** Desain use case pada aktor tim pengembang

Adapun pengkategorian yang ada pada use case sebagai berikut.

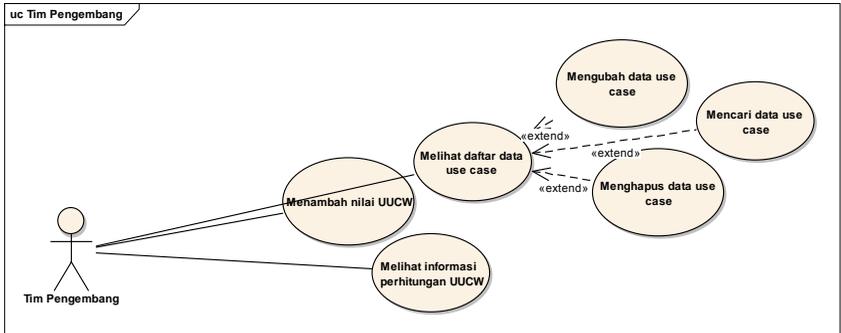
#### b. Fitur Autentifikasi

Fitur autentifikasi digunakan untuk melakukan login kedalam aplikasi. Jadi yang bisa melakukan login yaitu pengguna yang sudah terdaftar dalam sistem. Berikut ini adalah bentuk diagram use case untuk Fitur autentifikasi.



Gambar 4.3 Desain use case fitur autentifikasi

c. Fitur perhitungan nilai UUCW

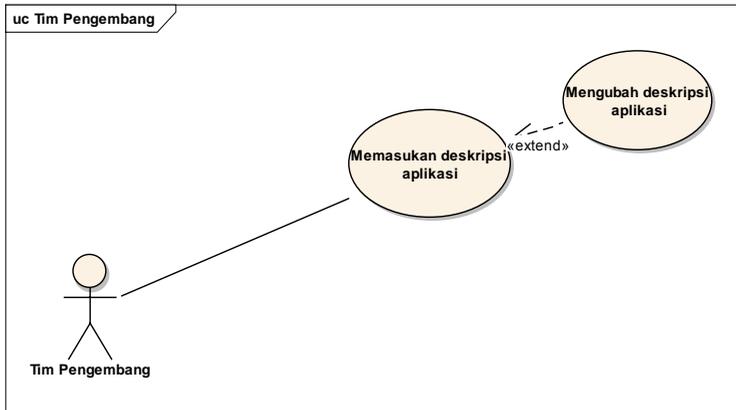


Gambar 4.4 Desain use case fitur perhitungan nilai UUCW

Gambar 4.4 merupakan fitur untuk menghitung nilai UUCW. Fitur perhitungan nilai UUCW ini terdiri dari lima use case. Pertama yaitu use case mendapatkan nilai UUCW.

d. Fitur memasukan deskripsi aplikasi

Gambar 4.5 berikut adalah bentuk dari desain Use Case untuk memasukan deskripsi aplikasi.



**Gambar 4.5** Desain *use case* memasukan deskripsi aplikasi

#### 4.2.4. *Use Case Description*

Deskripsi sebuah use case ( *Use Case Description*) berisikan tentang perilaku yang ada pada use case. Berikut ini merupakan salah satu contoh bentuk deskripsi use case yang ada pada aplikasi UCP. Use case pada tabel 4.9 merupakan bentuk use case untuk menambah nilai UUCW baru.

**Tabel 4.9 Use Case Description menambah nilai UUCW**

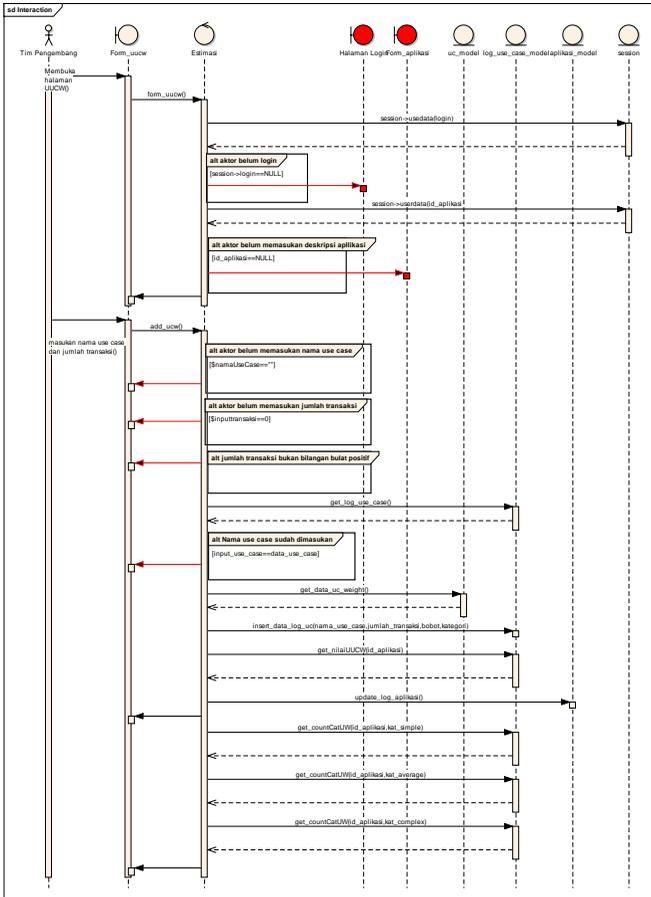
+	UCD.01.03 Menambah nilai UUCW		
<b>Ringkasan :</b>	<i>Use-case</i> ini dimaksudkan agar aktor bisa mendapatkan nilai UUCW dari aplikasi yang akan diestimasi		
<b>Direct Aktor :</b>	Tim Pengembang		
<b>Relasi antar use-case :</b>	UC.01.04		
<b>Prioritas :</b>	Penting ( <i>essential</i> )		
<b>Frekuensi Penggunaan :</b>	Bisa lebih dari satu kali		
<b>Pre Condition :</b>	Aktor sudah melakukan login, dan sudah memasukan deskripsi aplikasi		
<b>Skenario Sukses Utama :</b>	No	Aktor	Sistem
	1	Aktor membuka halaman perhitungan UUCW	Sistem menampilkan perhitungan UUCW
	2		Sistem melakukan pengecekan aktor melakukan login dan memasukan deskripsi aplikasi.
	3	Aktor memasukan nama use case dan jumlah transaksi	

	4	Aktor menekan tombol simpan	Sistem melakukan pengecekan apakah semua field sudah diisi dan jumlah transaksi yang dimasukan merupakan bilangan asli positif
	5		Sistem mengambil data perhitungan bobot use case beserta kategori use case
	6		Sistem melakukan pembobotan dan pengkategorian use case berdasarkan jumlah transaksi use case
	7		Sistem menyimpan hasil pengkategorian dan pembobotan
	8		Sistem melakukan penjumlahan bobot dari masing-masing use case membentuk nilai UUCW
	9		Sistem melakukan pengklasifikasian use case berdasarkan kategori use case
	10		Sistem menampilkan hasil nilai UUCW dan hasil klasifikasi use case

<b>Skenario Alternatif:</b>	<ul style="list-style-type: none"><li>• Alt No.1 Sistem akan menampilkan halaman login jika aktor belum melakukan login</li><li>• Alt No.1 Jika aktor belum memasukan deskripsi aplikasi, sistem akan menampilkan form deskripsi aplikasi</li><li>• Alt No.3 Sistem akan menampilkan pesan “Nama use case belum dimasukan” jika belum memasukan nama use case</li><li>• Alt No.3 Sistem akan menampilkan pesan “Jumlah transaksi belum dimasukan” jika belum memasukan jumlah transaksi</li><li>• Alt No.3 Sistem akan menampilkan pesan “Jumlah transaksi harus bilangan bulat positif”, jika memasukan jumlah transaksi bukan dalam bentuk bilangan bulat positif</li><li>• Alt No.3 Jika aktor memasukan use case yang sama dengan use case yang sudah dimasukan ke sistem, maka sistem akan menampilkan pesan “Nama use case sudah ada”.</li></ul>
-----------------------------	--

### 4.2.5. Desain Sequence Diagram

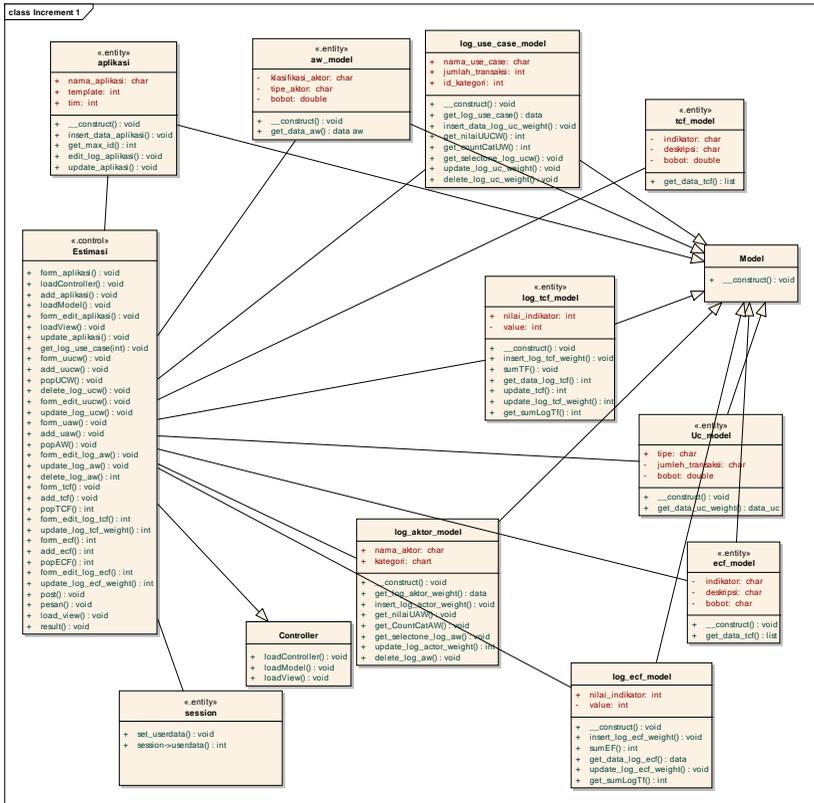
Desain sequence diagram dibuat berdasarkan dari desain *use case* dan deskripsi dari masing-masing *use case* yang ada pada aplikasi. Berikut ini adalah salah satu contoh *sequence diagram* yang ada pada aplikasi UCP



Gambar 4.6 Sequence Diagram menambah nilai UUCW

## 4.2.6. Desain Class Diagram

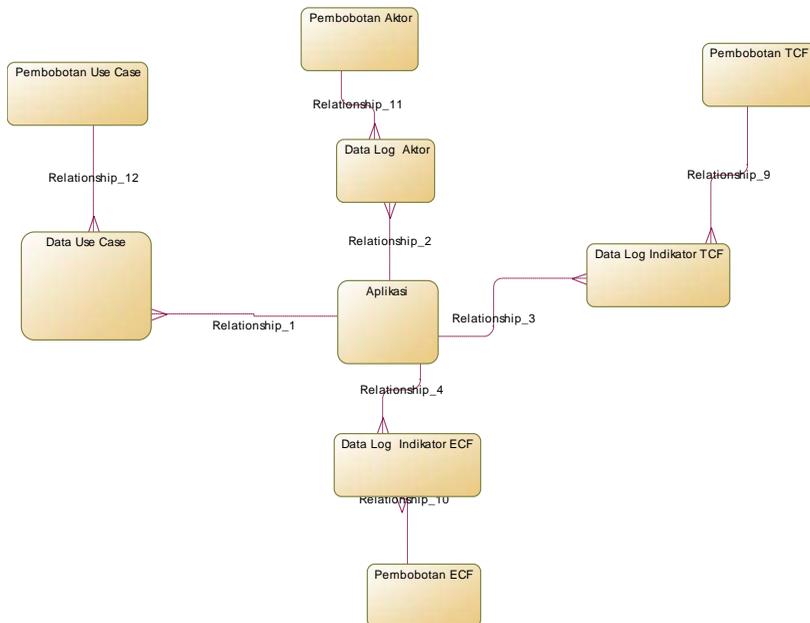
Desain *Class Diagram* berisikan fungsi-fungsi yang terdapat pada aplikasi. *Class Diagram* dibuat berdasarkan desain sequence diagram. Gambar 4.7 ini merupakan bentuk *Class Diagram* dari aplikasi UCP pada tahap *increment 1*.



Gambar 4.7 Desain Class Diagram tahap *increment 1*

#### 4.2.7. Conceptual Data Model(CDM)

Konseptual data model dari aplikasi UCP pada tahap *increment* 1 ini ada pada Gambar

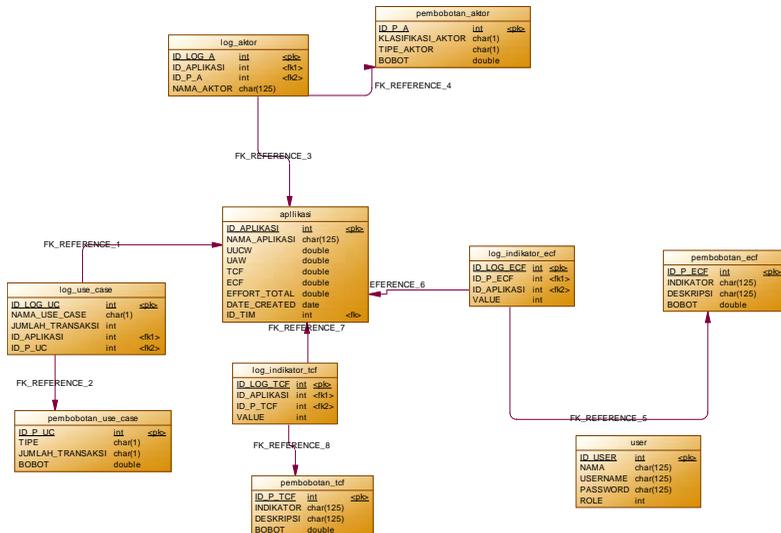


**Gambar 4.8** Desain konseptual data model

Desain konseptual data model ini nanti akan digunakan sebagai acuan dalam pembuatan database untuk aplikasi UCP ini. Kosenpuatal data model dibuat berdsasarkan deasin *Class Diagram*. Diagram konseptual hanya berisikan entitas dan antribut pada setiap entitis yang ada. Selain itu, dalam diagram ini juga terdapat relationship antar entitas yang ada.

#### 4.2.8. Physical Data Model (PDM)

Physical data model merepresentasikan desain dari database yang akan dibuat nanti. Pada Physical Data Model (PDM) ini terdapat beberapa atribut dan relasi antar tabel. Gambar 4.13 merupakan desain dari Physical Data Model (PDM) aplikasi UCP



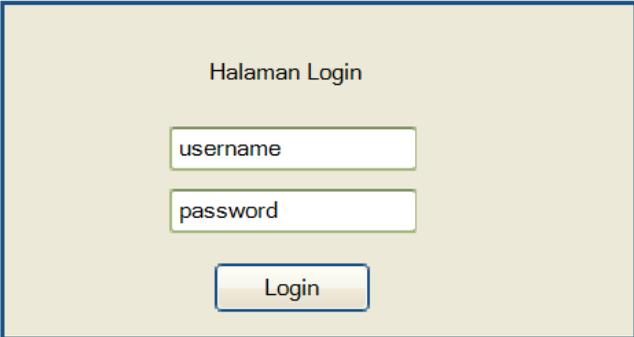
Gambar 4.9 Physical Data Model(PDM aplikasi UCP

*Physical Data Model* ini mengacu pada *Conceptual Data Model* yang sudah dibuat sebelumnya. *Physical Data Model* berisikan entitas, atribut, dan relationship antar entitas.

#### 4.2.9. Desain Antarmuka Pengguna

Desain antar muka dari Aplikasi estimasi harga ini dibagi menjadi 2 bagian, yaitu bagian depan (Frontend) dan bagian belakang (Backend). Detail antar muka pengguna dijelaskan pada dokumen SKPL. Berikut ini penjelasan beberapa bentuk antar muka pengguna pada aplikasi UCP.

##### 1. Halaman Login



The image shows a login page with a light beige background. At the top center, the text 'Halaman Login' is displayed. Below this, there are two input fields: the first is labeled 'username' and the second is labeled 'password'. At the bottom center, there is a button labeled 'Login'.

**Gambar 4.10** Desain halaman login

##### 2. Bagian Depan (Frontend)

Desain bagian *frontend* ini terdiri dari beberapa bagian. Adapun bagian yang ada pada frontend ini yaitu sebagai berikut.

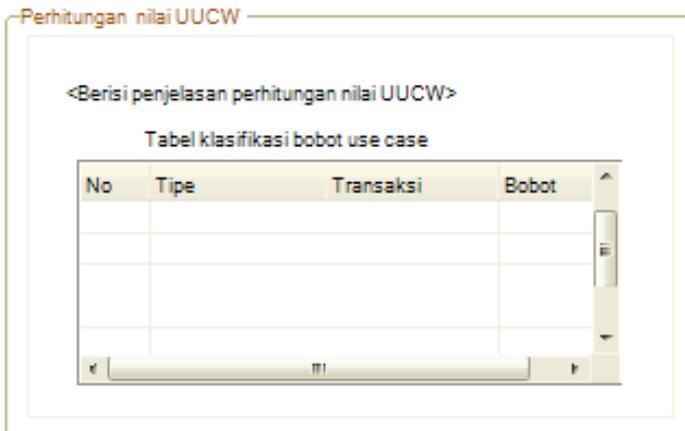
- Form deskripsi aplikasi.

Gambar 4.11 merupakan bentuk desain interface dari *form* aplikasi. Pada desain gambar tersebut terdapat *field input* nama aplikai.



- Popup informasi perhitungan UUCW

Gambar 4.13 merupakan desain dari popup informasi perhitungan use case. Popup ini bertujuan untuk memberikan informasi kepada pengguna terkait dengan perhitungan Use Case dan bagaimana melakukan pengkategorian use case yang ada berdasarkan jumlah transaksi dari use case tersebut.



**Gambar 4.13** Popup informasi perhitungan UUCW

Pada gambar tersebut terlihat bahwa terdapat bagian penjelasan dan bagian daftar dalam bentuk tabel. Tabel tersebut nantinya digunakan untuk menampilkan daftar pengkategorian use case beserta penjesalan transaksi dan nilai bobot pada masing-masing kategori use case.

#### 4.2.10. Perancangan Pengujian

- a. Perancangan pengujian kebutuhan fungsional
  - Identifikasi dan Rencana Pengujian

Berikut ini adalah beberapa contoh bentuk rencana pengujian untuk kebutuhan fungsional aplikasi pada tahap increment 1 pada tabel berikut ini.

**Tabel 4.10 Identifikasi Pengujian kebutuhan fungsional increment 1**

No	Kode KF	Kode Use Case	Nama Use Case	Pengujian	Bentuk pengujian	Identifikasi pengujian
1	KF01	UC.03.01	Login	Skenario Normal	Black box	TC001
				Skenario Alternatif	Black box	TC002
		UC.03.02	Logout	Skenario Normal	Black box	TC003
2	KF02	UC.01.01	Memasukan deskripsi aplikasi	Skenario Normal	Black box	TC004
				Skenario Alternatif	Black box	TC005
		UC.01.02	Mengubah deskripsi aplikasi	Skenario Normal	Black box	TC006
				Skenario Alternatif	Black box	TC007
		UC.01.23	Memasukan nama fitur aplikasi	Skenario Normal	Black box	TC008
				Skenario Alternatif	Black box	TC009

No	Kode KF	Kode Use Case	Nama Use Case	Pengujian	Bentuk pengujian	Identifikasi pengujian
3	KF03	UC.01.24	Menghapus nama fitur aplikasi	Skenario Normal	Black box	TC010
		UC.01.25	Melihat daftar nama fitur aplikasi	Skenario Normal	Black box	TC011
				Skenario Alternatif	Black box	TC012
		UC.01.03	Menambah nilai UUCW	Skenario Normal	Black box	TC013
				Skenario Alternatif	Black box	TC014
		UC.01.04	Melihat Informasi perhitungan UUCW	Skenario Normal	Black box	TC015
		UC.01.05	Melihat daftar data use case	Skenario Normal	Black box	TC016
				Skenario Alternatif	Black box	TC017
		UC.01.06	Mengubah data Use Case	Skenario Normal	Black box	TC018
				Skenario Alternatif	Black box	TC019
UC.01.07	Menghapus data Use Case	Skenario Normal	Black box	TC020		

- Prosedur Pengujian

Setelah itu, tahapan yang dilakukan selanjutnya yaitu menentukan prosedur pengujian. Berikut ini adalah beberapa contoh prosedur pengujian yang ada pada tahap increment 1.

**Tabel 4.11 Prosedur Pengujian kebutuhan fungsional increment 1**

No	Identifikasi pengujian	Skenario	Prosedur Pengujian	Masukan	Output Espektasi
1	TC001	Skenario Normal	<ol style="list-style-type: none"> <li>1. Membuka halaman login aplikasi</li> <li>2. Kemudian memasukan username dan password yang sudah terdaftar dalam database untuk login.</li> </ol>	Input karakter password dan username	Berhasil ke halaman login
2	TC002	Skenario Alternatif	<ol style="list-style-type: none"> <li>1. Membuka halaman login aplikasi</li> <li>2. Memasukan username dan password yang tidak ada dalam database.</li> </ol>	Input karakter username dan password	Keluar pesan “ username atau password salah “
3	TC003	Skenario Normal	Melakukan klik tombol logout	-	Masuk ke halaman login aplikasi

b. Perancangan pengujian kebutuhan non fungsional

Prancangan pengujian juga dilakukan untuk kebutuhan non-fungsional. Tahapan pengujian kebutuhan non-fungsional juga sama dengan kebutuhan fungsional, yaitu dengan melakukan identifikasi dan rencana pengujian serta membuat prosedur pengujian.

- Identifikasi dan rencana pengujian

Berikut ini merupakan beberapa contoh hasil identifikasi pengujian pada *tahap increment 1*

**Tabel 4.12 Identifikasi Pengujian kebutuhan non- fungsional increment 1**

No	Kode KNF	Kode Object Pengujian	Bentuk pengujian	Tools	Identifikasi Pengujian
1	KNF-01	TC001-TC078	Pengujian Correctness	Pengujian dilakukan dengan menggunakan pengujian <i>blackbox testing</i>	TC079
2	KNF-02	AM01-AM16	Pengujian dengan pengamatan	Browser untuk menjalankan aplikasi	TC080

- Prosedur pengujian

Berikut ini adalah beberapa contoh prosedur pengujian yang ada pada tahap increment 1 untuk kebutuhan non-fungsional

**Tabel 4.13** Prosedur pengujian kebutuhan non-fungsional increment 1

<b>No</b>	<b>Identifikasi Pengujian</b>	<b>Prosedur</b>	<b>Input</b>	<b>Output Ekspektasi</b>
1	TC079	Prosedur sama dengan prosedur yang ada pada pengujian kebutuhan fungsional	Skenario Sukses dan Skenario alternative semua use case	Hasil sesuai dengan yang sudah dispesifikasikan
5	TC080	Prosedur prosedur pengujian dilakukan dengan melakukan pengamatan secara langsung mengacu pada aturan Ben Sheinderman.	Desain Antarmuka	Hasil sesuai dengan aturan yang ada pada Ben Sheinderman

*“Halaman sengaja dikosongkan”*

### 4.3. Perancangan Tahap *Increment 2*

Tahapan *increment 2* ini dilakukan mengacu pada hasil perancangan dan implmentasi pada tahap *increment 1* dan juga hasil wawancara yang dilakukan pada tahap *increment 2*.

#### 4.3.1. Penggalian Kebutuhan

##### a. Translasi Wawancara Kedalam Kebutuhan

Pada tahap *increment 2* ini, peneliti juga melakukan wawancara. Hasil dari wawancara. Sama seperti pada tahap sebelumnya, hasil wawancara yang didapat selanjutnya ditranslasikan kedalam sebuah kebutuhan (*Requirement*) Tabel 4.14 berikut merupakan hasil translasi kebutuhan dari hasil wawancara

**Tabel 4.14** Translasi kebutuhan

Kode	Point of Interview	Requirement
PI-02-01	<p><b>Biaya estimasi berdasarkan gaji masing-masing profesi</b></p> <p>Estimasi biaya perangkat lunak didasarkan dari gaji dari profesi tim pengembang perangkat lunak Perusahaan DTS.</p> <p><b>Ref:IR-15</b></p>	Fitur Entry Gaji profesi
PI-02-02	<p><b>Gaji akan berubah-ubah</b></p> <p>Gaji yang ada pada setiap profesi bisa berubah-ubah sesuai dengan tingkat inflasi yang ada di indonesia, perubahan dilakukan oleh direktur</p> <p><b>Ref:IR-15</b></p>	Fitur perubahan gaji masing-masing profesi

Kode	Point of Interview	Requirement
PI-02-03	<p><b>pemberitahuan hasil estimasi</b></p> <p>Untuk memudahkan validasi, direktur mendapatkan informasi melalui notifikasi</p> <p><b>Ref:IR-05,IR-16</b></p>	Fitur Notifikasi email yang dilakukan oleh direktur
PI-02-04	<p><b>melihat rekap biaya hasil estimasi</b></p> <p>Pengguna aplikasi bisa melihat hasil rekap estimasi yang dilakukan oleh tim pengembang</p> <p><b>Ref: IR-10</b></p>	Fitur Rekap hasil biaya estimasi
PI-02-05	<p><b>mengubah biaya hasil estimasi</b></p> <p>Direktur mempunyai kuasa dalam melakukan perubahan/customisasi terhadap biaya dari hasil estimasi yang sudah dilakukan oleh tim pengembang</p> <p><b>Ref: IR-18</b></p>	Fitur perubahan biaya dari hasil estimasi perangkat lunak
PI-02-06	<p><b>mencetak dokumen laporan</b></p> <p>Setelah Direktur melakukan validasi , sekretaris bisa melakukan pencetakan dokumen penawaran harga</p>	Fitur cetak dokumen penawaran

Kode	Point of Interview	Requirement
	yang akan diserahkan ke client <b>Ref:IR-17</b>	
PI-02-07	<b>Mengubah pelaku aktivitas</b>  Pelaku yang ada pada masing-masing aktivitas dapat dilakukan perubahan sesuai dengan perkembangan yang ada. <b>Ref:IR-15</b>	Fitur perubahan pelaku aktivitas
PI-02-08	<b>Validasi hasil estimasi</b>  Direktur mempunyai hak dalam menolak maupun menerima hasil estimasi yang dilakukan oleh tim pengembang <b>Ref:IR-04</b>	Fitur Validasi hasil estimasi

b. Kebutuhan Fungsional Sistem

Adapun kebutuhan fungsional yang ada dalam aplikasi ini ada pada Tabel 4.15 berikut.

**Tabel 4.15** Tabel kebutuhan fungsional sistem

Kode	Deskripsi kebutuhan
KF08	Terdapat fitur pembedaan hak akses pada aplikasi
KF09	Terdapat fitur perubahan terhadap nilai hasil perhitungan UCP dan biaya
KF10	Terdapat fitur notifikasi via email
KF11	Terdapat tampilan halaman awal(homepage)

Kode	Deskripsi kebutuhan
	yang terdiri dari informasi perusahaan
KF12	Terdapat fitur pengelola pengguna yang login dalam aplikasi
KF13	Fitur pengelola data profesi
KF14	Fitur pengelola aktivitas
KF15	Fitur log aplikasi
KF16	Fitur validasi hasil estimasi
KF17	Fitur cetak dokumen penawaran

### c. Kebutuhan Non Fungsionalitas Sistem

Pada tahap *increment 2* ini, kebutuhan non fungsional sama dengan tahap *increment 1*.

## 4.3.2. Analisa Kebutuhan

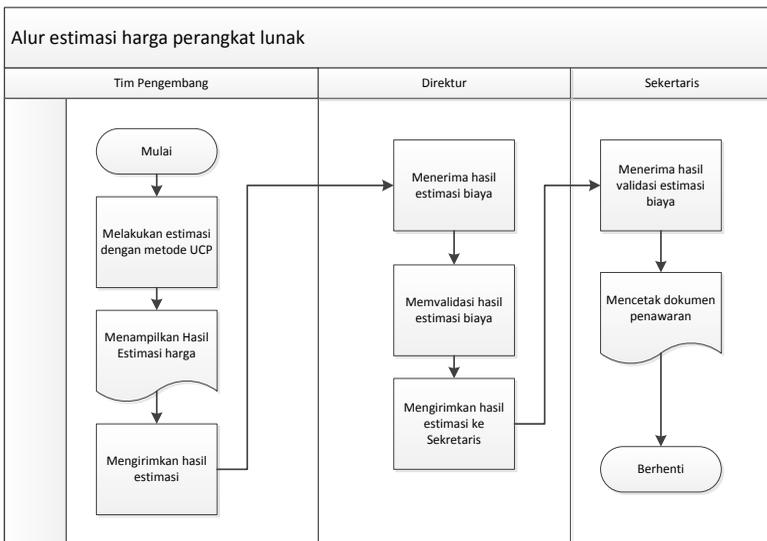
### a. Profil perusahaan

Perusahaan DTS merupakan perusahaan yang didirikan atas inisiatif mahasiswa aktif jurusan Sistem Informasi ITS. Perusahaan DTS mulai beroperasi pada pertengahan bulan september 2011. Perusahaan DTS ini bergerak di bidang jasa pembuatan website profil pribadi maupun organisasi. Selain website, juga melayani pembuatan aplikasi berbasis web yang bisa menunjang dan mempercepat proses bisnis yang ada di perusahaan/organisasi [19].

### b. Proses penentuan harga perangkat lunak

Berdasarkan hasil wawancara dan diskusi yang sudah dilakukan dengan perwakilan Perusahaan DTS, proses penentuan harga perangkat lunak yang nantinya dijadikan acuan dalam pembuatan aplikasi ini yaitu ada pada Gambar 4.15. Gambar tersebut menjelaskan bahwa proses estimasi harga awalnya dilakukan oleh tim pengembang. Tim pengembang awalnya menghitung estimasi harga perangkat lunak dengan menggunakan metode UCP. Setelah mendapatkan hasil estimasi harga, Tim pengembang kemudian menampilkan hasil estimasi tersebut

untuk dilakukan *review*. Setelah dilakukan *review*, tim pengembang kemudian mengirimkan hasil estimasi harga perangkat lunak kepada direktur. Direktur kemudian melakukan validasi terhadap hasil estimasi yang dilakukan oleh tim. Jika Direktur setuju, maka direktur akan melanjutkan ke sekretaris untuk dibuatkan dokumen penawaran, jika tidak setuju, maka direktur akan mengkonfirmasi kepada tim pengembang untuk dilakukan estimasi harga ulang untuk kemudian dilakukan validasi kembali oleh Direktur.



**Gambar 4.14** Proses penentuan harga perangkat lunak

c. Tim Pengembang Perangkat Lunak Perusahaan DTS

Tim pengembang perangkat lunak yang ada pada perusahaan DTS ini ada pada tabel berikut ini.

**Tabel 4.16 Tim pengembang perangkat lunak**

No	Jabatan tim	Jobdesk
1	Administrasi dan Keuangan	<ol style="list-style-type: none"> <li>1. Membuat dan menyiapkan laporan-laporan dan dokumen-dokumen yang diperlukan oleh Direktur.</li> <li>2. Pencatatan agenda rapat dan hasil rapat saat rapat DTS bulanan berlangsung.</li> <li>3. Membuat perencanaan harga produk dengan berkonsultasi dengan programmer.</li> </ol>
2	System Analyst	Bertugas untuk membantu Managing Director dalam menyiapkan data SKPL awal untuk kebutuhan dokumen penawaran project.
3	Software Developer	<ol style="list-style-type: none"> <li>1. Merekomendasikan anggota tim developer yang siap untuk mengikuti project DTS kepada managing director.</li> <li>2. Membantu meningkatkan kemampuan sdm developer dengan memberikan studi kasus untuk diselesaikan.</li> </ol>
4	Tim Developer	<ol style="list-style-type: none"> <li>1. Bertugas menangani dalam pembuatan aplikasi dan melayani dalam memberikan solusi IT yang tepat kepada customer.</li> <li>2. Memiliki tanggungjawab untuk terus belajar dengan software Developer dan aktif serta bertanya kepada divisi R&amp;D untuk diadakan tutorial/pelatihan.</li> </ol>
5	Designer	<ol style="list-style-type: none"> <li>1. Bertugas membuat desain aplikasi/web yang bekoordinasi dengan divisi software developer untuk membuat tampilan desain yang dibutuhkan.</li> <li>2. Bertugas membantu desain yang diperlukan untuk menunjang tim marketing DTS, seperti; desain kartu nama, brosur, poster dsb.</li> </ol>

d. Estimasi nilai

Nilai usaha yang sudah didapat menggunakan metode UCP kemudian akan dirubah kedalam bentuk nilai uang. Untuk mengubah kedalam bentuk nilai uang, nilai usaha didistribusikan kedalam bentuk aktivitas-aktivitas dalam pengembangan perangkat lunak dengan cara mengalihkan nilai usaha dengan presentase yang ada pada setiap aktivitas sehingga mendapatkan nilai *Hour Effort*. Berdasarkan penelitian yang dilakukan oleh Kassem Shaleh (2011), aktivitas yang ada pada pengembangan perangkat lunak dibedakan menjadi 3 bagian, yaitu: *Software Development*, *On Going Activity*, dan *Quality and Testing*. Tabel 4.17 berikut ini menunjukkan daftar distribusi aktivitas yang ada dalam pengembangan perangkat lunak [16].

**Tabel 4.17 Distribusi Usaha dari masing-masing aktivitas**

NO.	Aktivitas	Presentase (%)
<b>1</b>	<b>Software Development</b>	
a	Requirement	7.50%
b	Specifications & Design	17.50%
c	Coding	10.00%
d	Testing	7.00%
<b>2</b>	<b>On Going Activity</b>	
a	Project Management	7.00%
b	Configuration Management	3.00%
c	Documentation	3.00%
d	Training & Support	3.00%
e	Acceptance & Deployment	5.00%
<b>3</b>	<b>Quality and Testing</b>	
a	Quality Assurance & Control	12.34%
b	Evaluation and Testing	24.66%
	Total	<b>100%</b>

Adapun detail dari setiap aktivitas ada pada dokumen lampiran. *Hourse of Effort* ini dikalikan dengan standard gaji dari personel yang ada di masing-masing kelompok aktivitas yang ada di masing-masing tahapan proyek. Standrd Tabel 4.18 merupakan daftar personel yang ada pada masing-masing aktivitas yang ada dalam pengembangan perangkat lunak yang didasarkan atas refrensi pmbok dan dari hasil wawancara dengan client (IR-14 dan IR-15).

**Tabel 4.18 Daftar personel yang ada pada masing-masing aktivitas**

NO.	Aktivitas	Presentase (%)
<b>1</b>	<b>Software Development</b>	
a	Requirement	Sistem Analyst
b	Specifications & Design	Sistem Analyst
c	Coding	Software Developer
d	Testing	Tester
<b>2</b>	<b>On Going Activity</b>	
a	Project Management	Manajer Proyek
b	Configuration Management	Sistem Analyst
c	Documentation	Dokumentasi/Administrasi
d	Training & Support	Manajer Proyek
e	Acceptance & Deployment	Manajer Proyek
<b>3</b>	<b>Quality and Testing</b>	
a	Quality Assurance & Control	Sistem Analyst
b	Evaluation and Testing	Manajer Proyek
	Total	<b>100%</b>

Penggajian tim pengembang yang ada di perusahaan DTS berdasarkan dari besaran proyek sehingga gaji yang didapatkan oleh tim berbeda-beda di setiap proyek. Hal ini akan menjadi

masalah ketika akan melakukan estimasi harga perangkat lunak. Oleh karena itu, diperlukan standard gaji tetap dari masing-masing personel tim pengembang. Berdasarkan hasil kesepakatan dengan *client*, standard gaji yang digunakan untuk estimasi perangkat lunak yaitu standard gaji yang dikeluarkan oleh Kelly Service. Sebelum melakukan hal tersebut, terlebih dahulu dilakukan pemetaan jabatan tim yang ada di perusahaan DTS dengan jabatan yang ada di standard kelly pada tahun 2013 [17].

**Tabel 4.19 Pemetaan jabatan tim dengan jabatan standard Kelly**

Jabatan versi DTS	Jabatan versi Kelly	Pendidikan/ pengalaman	Gaji /bulan(Rp )
<i>Software Developer</i>	<i>Analyst Programmer/ Software Engineer</i>	Sarjana (min 1 th)	5.000.000
<i>System Analyst</i>	<i>Business/ system analyst</i>	Sarjana (min 2 th)	7.000.000
Manajer Proyek	<i>Project Manager</i>	Sarjana (min 3 th)	8.500.000
<i>Tester</i>	<i>Software QA/test analyst</i>	Sarjana (3-6 th)	5.000.000
<i>Dokumentasi /Administrasi</i>	<i>Seccretary</i>	Sarjana (min 1 th)	4.000.000

Menurut penelitian yang dilakukan oleh linda(2015), bahwa standard Kelly masih terlalu besar untuk standard gaji, perlu dilakukan penurunan hingga 55% untuk mendekati keakuratan yang lebih baik [18]. Hasil pengurangan gaji sebagai berikut.

**Tabel 4.20 Standard gaji Kelly setelah diturunkan 50%**

Jabatan versi Kelly	Gaji /bulan(Rp)	Gaji setelah diturunkan 50% /bulan(Rp)
<i>Analyst Programmer/ Software Engineer</i>	5.000.000	2.250.000

Jabatan versi Kelly	Gaji /bulan(Rp)	Gaji setelah diturunkan 50% /bulan(Rp)
<i>Business/ system analyst</i>	7.000.000	3.150.000
<i>Project Manager</i>	8.500.000	3.825.000
<i>Software QA/test analyst</i>	5.000.000	2.250.000
<i>Secretary</i>	4.000.000	1.800.000

Untuk mengalihkan gaji dengan nilai usaha. Gaji yang ada tersebut harus diubah menjadi jam dengan cara dibagi dengan 160 ( 1 bulan =160 jam). Perhitungan biaya dan pengubahan gaji menjadi jam ini nantinya akan dilakukan secara otomatis. Pengguna hanya memasukan input gaji dalam bentuk nilai bulan saja

Output dalam tahapan ini yaitu berupa estimasi biaya perangkat lunak dalam bentuk nilai uang.

#### 4.3.3. Hak Akses Pengguna

Pada tahap *increment 2* ini terdapat fitur untuk melakukan pembedaan hak akses. Berdasarkan hasil penggalian kebutuhan menunjukkan bahwa dalam aplikasi Use Case Point nanti terdapat 3 aktor yang menggunakan. Masing-masing aktor memiliki hak akses (*previlage*) masing-masing. Adapun detail dari hak akses aplikasi Use Case Point ini pada tabel 4.21 sebagai berikut.

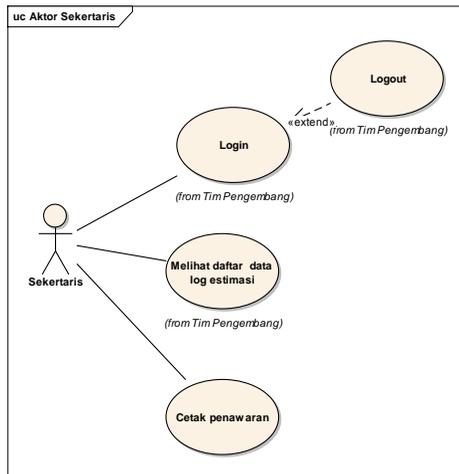
**Tabel 4.21 Peran pengguna**

Nama Peran	Peran
Direktur	<ul style="list-style-type: none"> <li>• Melakukan perhitungan estimasi biaya</li> <li>• Melakukan pengubahan hasil estimasi</li> <li>• Melakukan pengubahan estimasi biaya</li> <li>• Mengelola akun pengguna</li> <li>• Mengelola data profesi</li> <li>• Mengelola data aktivitas</li> <li>• Mengelola data anggota pengembang</li> <li>• Melakukan validasi hasil estimasi</li> </ul>
Tim pengembang	<ul style="list-style-type: none"> <li>• Melakukan perhitungan estimasi biaya</li> <li>• Melakukan pengubahan hasil estimasi</li> </ul>





c. Desain use case diagram aktor Sekretaris



Gambar 4.17 Desain use case aktor sekretaris

#### 4.3.5. Use Case Description

Tabel 4.22 berikut ini adalah salah satu contoh *Use Case Description* pada tahap *increment 2*. *Use Case Description* tersebut merupakan bentuk dari proses untuk memasukan deskripsi aplikasi Dalam memasukan deskripsi aplikasi. Pengguna/aktor membuka form deskripsi aplikasi. Kemudian aktor memasukan deskripsi aplikasi dan seterusnya sampai deskripsi aplikasi berhasil disimpan.

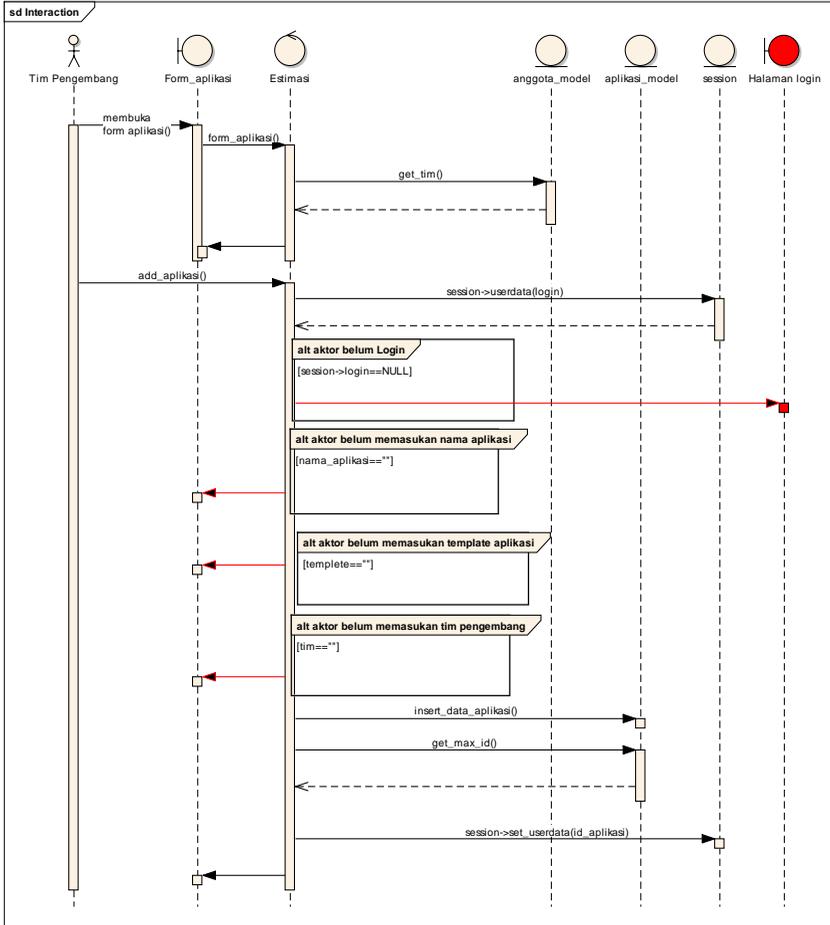
**Tabel 4.22 Use case diagram memasukan deskripsi aplikasi**

+	UCD.01.01 Memasukan deskripsi aplikasi	
<b>Ringkasan :</b>	Use-case ini dimaksudkan agar aktor bisa memasukan deskripsi aplikasi	
<b>Direct Aktor :</b>	Tim Pengembang	
<b>Relasi antar use-case :</b>	-	
<b>Prioritas :</b>	Penting ( <i>essential</i> )	
<b>Frekuensi Penggunaan :</b>	Bisa lebih dari satu kali	
<b>Pre Condition :</b>	Aktor sudah melakukan login	
<b>Skenario Sukses Utama :</b>	No	Aktor
	1	Aktor membuka halaman form deskripsi aplikasi
	2	Sistem melakukan pengecekan untuk memastikan bahwa aktor sudah melakukan login
	3	Sistem menampilkan halaman form deskripsi aplikasi
	4	Aktor memasukan deskripsi aplikasi yang
		Sistem

<b>Skenario Alternatif:</b>		terdiri dari Nama Aplikasi, Penggunaan Template, dan Tim Pengembang.	
	5	Aktor mengklik tombol simpan.	Sistem melakukan pengecekan apakah semua field yang disediakan sudah diisi
	6		Sistem menyimpan data deskripsi aplikasi
	<ul style="list-style-type: none"> <li>• Alt No.2 Sistem akan menampilkan halaman login jika aktor belum melakukan login</li> <li>• Alt No. 5 Sistem akan menampilkan pesan “Nama aplikasi belum dimasukan” jika belum memasukan nama aplikasi</li> <li>• Alt No. 5 Sistem akan menampilkan pesan “Penggunaan template belum diisi” jika belum memilih penggunaan template</li> <li>• Alt No. 5 Sistem akan menampilkan pesan “Tim pengembang belum diisi” jika belum memilih tim pengembang aplikasi</li> </ul>		

### 4.3.6. Desain Sequence Diagram

Sequence Diagram pada tahap increment 2 ini ada pada gambar 4.18 berikut.

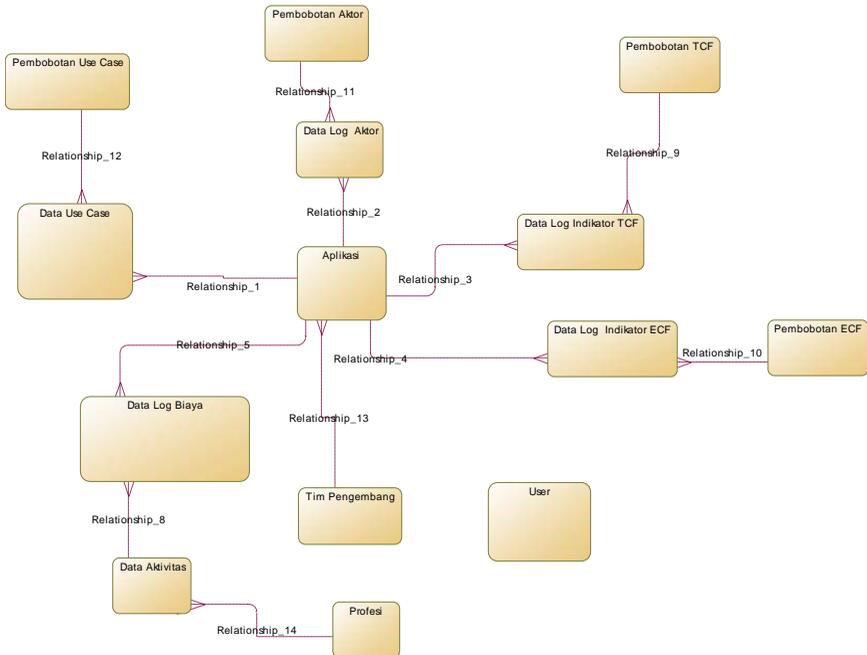


Gambar 4.18 Sequence Diagram memasukan deskripsi aplikasi



### 4.3.8. Conceptual Data Model(CDM)

Gambar 4.20 berikut adalah bentuk dari model konseptual data aplikasi UCP pada tahap *increment 2*.

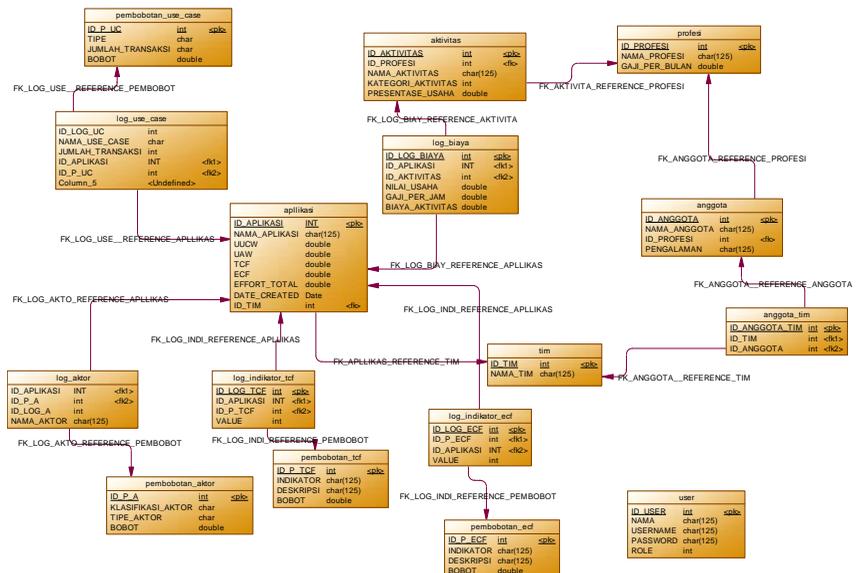


**Gambar 4.20 Conceptual Data Model (CDM)**

Conceptual Data Model (CDM) ini berisikan entitas-entitas yang nantinya dijadikan acuan dalam pembuatan tabel database untuk tahap *increment 2*. Selain berisikan entitas, juga terdapat hubungan (*relationship*) antar entitas.

### 4.3.9. Physical Data Model (PDM)

Gambar 4.21 berikut ini adalah bentuk dari Physical Data Model (PDM) aplikasi UCP pada tahap *increment 2*



Gambar 4.21 Physical Data Model (PDM)

*Physical Data Model (PDM)* merupakan bagian terakhir dari pemodelan data. Hasil dari desain PDM ini nantinya bisa langsung di-generate kedalam bentuk sebuah database. Pada setiap entitas yang ada dalam PDM, terdapat atribut-atribut yang ada. Atribut tersebut digunakan untuk menyimpan data sesuai dengan type data pada masing-masing atribut.

#### 4.3.10. Desain Antarmuka Pengguna

Desain antar muka pada tahap *increment 2* pada dasarnya sama dengan desain yang ada pada tahap *increment 1*. Berikut ini adalah beberapa contoh desain antarmuka pengguna aplikasi UCP.

##### a. Desain Form Deskripsi Aplikasi

Perbedaan desain form aplikasi pada tahap *increment 1* dengan tahap *increment 2* ini yaitu terletak pada penambahan *field* baru. *Field* tersebut antara lain yaitu: *field* penggunaan template, dan *field* anggota pengembang.

**Gambar 4.22** Desain form deskripsi aplikasi

##### b. Desain Distribusi Usaha Beserta Nilai Biaya

Gambar4.18 merupakan desain dari pelaporan/*reporting* hasil perhitungan yang sudah dalam bentuk nilai biaya uang. Dalam report ini juga ditampilkan distribusi biaya dan *Effort* yang ada dalam masing-masing aktivitas.

Aktivitas	Effort		Biaya		
	%	Hour of Effort	Standard gaji/jam	Biaya per Jam	Biaya per bulan
<b>SOFTWARE DEVELOPMENT</b>					
Coding	10.00 %	2,48	Rp. 15.625,00	Rp. 38.775,00	Rp. 6.204.000,00
Requirements	7.50 %	1,86	Rp. 25.000,00	Rp. 46.530,00	Rp. 7.444.800,00
Specifications & Design	17.50 %	4,34	Rp. 25.000,00	Rp. 108.570,00	Rp. 17.371.200,00
Testing	7.00 %	1,74	Rp. 18.750,00	Rp. 32.571,00	Rp. 5.211.360,00
Sub Total	42,00%	10,42		Rp. 226.446,00	
<b>ON GOING ACTIVITY</b>					
Configuration Management	3.00 %	0,74	Rp. 22.727,27	Rp. 16.920,00	
Training & Support	3.00 %	0,74	Rp. 22.727,27	Rp. 16.920,00	
Project management	7.00 %	1,74	Rp. 48.295,45	Rp. 83.895,00	
Acceptance & Deployment	5.00 %	1,24	Rp. 48.295,45	Rp. 59.925,00	
Documentation	3.00 %	0,74	Rp. 8.522,73	Rp. 6.345,00	
Sub Total	21,00%	5,21		Rp. 184.005,00	
<b>QUALITY AND TESTING</b>					
Quality Assurance & Control	12.34 %	3,06	Rp. 48.295,45	Rp. 147.894,90	
Evaluation and Testing	24.66 %	6,12	Rp. 17.045,45	Rp. 104.311,80	
Sub Total	37,00%	9,18		Rp. 252.206,70	
<b>TOTAL BIAYA</b>				Rp. 662.657,70	

**Gambar 4.23 Report estimasi distribusi aktivitas dan biaya**

c. Desain halaman Daftar Pengguna Aplikasi

Gambar 4.24 merupakan bentuk desain tampilan daftar pengguna. Desain daftar tampilan pengguna ini digunakan untuk melihat pengguna yang terdaftar dalam aplikasi UCP.

Daftar Pengguna

<search>

No	Nama	Username	Role	Action
1	Faiz	admin	Project Manager	Delete  Edit

**Gambar 4.24** Desain tampilan daftar pengguna

Halaman daftar pengguna bersikan data pengguna yang sudah terdaftar dalam sistem. Daftar pengguna yang ada dalam sistem terdiri dari 3 kategori peran. pertama, yaitu sebagai direktur, kedua yaitu sebagai tim pengembang, ketiga yaitu sebagai sekretaris. Setiap kategori peran mempunyai hak akses yang berbeda-beda. Direktur mempunyai hak akses paling tinggi diantara kedua peran tersebut. Pada halaman daftar pengguna, terdapat 5 kolom. Pertama, yaitu kolom No. Kolom ini berfungsi untuk menampilkan nomer urut dari data pengguna yang ada dalam sistem. Kedua, yaitu kolom username, kolom ini berisikan username yang ada pada setiap pengguna. Ketiga, yaitu kolom Role yang berfungsi untuk menampilkan role(peran) dari setiap pengguna yang sudah terdaftar dalam sistem.

### 4.3.11. Perancangan Pengujian

a. Perancangan pengujian kebutuhan fungsional

- Identifikasi dan perencanaan pengujian

Berikut ini adalah beberapa rencana pengujian yang ada pada tahap *increment 2*.

**Tabel 4.23 Identifikasi Pengujian kebutuhan fungsional *increment 2***

No	Kode KF	Kode Use Case	Nama Use Case	Pengujian	Bentuk pengujian	Identifikasi pengujian
1	KF09	UC.01.22	Melihat data daftar log estimasi	Skenario Normal	Black box	TC043
				Skenario Alternatif	Black box	TC044
		UC.02.21	Mengubah log biaya estimasi	Skenario Normal	Black box	TC045
				Skenario Alternatif	Black box	TC046
2	KF10	UC.01.20	Mengirim hasil estimasi	Skenario Normal	Black box	TC047
3	KF11	UC.03.03	Membuka halaman awal	Skenario Normal	Black box	TC048

- Prosedur pengujian

Adapun beberapa contoh prosedur untuk pengujian kebutuhan fungsional sebagai berikut.

**Tabel 4.24 Prosedur Pengujian kebutuhan fungsional increment 2**

No	Identifikasi pengujian	Skenario	Prosedur Pengujian	Masukan	Output Espektasi
1	TC043	Skenario Normal	1. Melakukan login kedalam system Membuka halaman log estimasi	-	Daftar log data hasil estimasi yang disajikan dalam bentuk tabel
2	TC044	Skenario Alternatif	1. Mengosongkan data log estimasi aplikasi 2. Melakukan login kedalam system 3. Membuka halaman log estimasi	-	Tampil pesan “Data kosong” pada halaman log estimasi

b. Perancangan pengujian kebutuhan non-fungsional

Perancangan pengujian untuk kebutuhan fungsional pada tahap increment 2 ini tidak ada.. Hal ini karena pada tahap increment 2 ini, tidak ada tambahan baru untuk kebutuhan non-fungsional

#### 4.4. Perancangan Tahap *Increment 3*

##### 4.4.1. Penggalian Kebutuhan

Pada tahap *increment 3* ini, peneliti juga melakukan penggalian kebutuhan kembali untuk mengetahui kebutuhan apa saja yang perlu ditambahkan kembali setelah hasil pengujian pada tahap *increment 2*.

##### a. Translasi Wawancara Kedalam Kebutuhan

Berikut ini adalah hasil translasi wawancara Kedalam Kebutuhan dan fitur potensial yang tersaji pada tabel 4.25 berikut.

**Tabel 4.25** Translasi kebutuhan

Kode	Point of Interview	Requirement
PI05	<p><b>Fitur otomatisasi nilai ER berdasarkan historis</b></p> <p>Nilai ER yang digunakan untuk mendapatkan nilai usaha bersifat <i>dynamic</i> berdasarkan historis estimasi</p> <p><b>Ref:IR-18,IR-19</b></p>	Fitur nilai ER yang dinamis
PI06	<p><b>Fitur melakukan feedback estimasi</b></p> <p>Setiap hasil estimasi, akan dilakukan feedback dengan cara merevisi masing-masing komponen perhitungan estimasi</p> <p><b>Ref:IR-18,IR-19</b></p>	Refine masing-masing komponen perhitungan UCP

##### b. Membuat daftar kebutuhan fungsional

Daftar kebutuhan aplikasi dibuat berdasarkan hasil analisa terhadap cerita pengguna (*User Story*). Daftar kebutuhan aplikasi

ini nantinya akan dijadikan acuan dalam pembuatan dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Berikut ini merupakan daftar kebutuhan yang ada pada aplikasi UCP yang sudah dikonfirmasi kepada *client*.

**Tabel 4.26 Daftar kebutuhan fungsional aplikasi UCP increment 3**

Kode	Penjelasan
KF19	Fitur memasukan Actual <i>Effort</i>
KF20	Fitur informasi client
KF21	Fitur Biaya Operasional
KF22	Fitur <i>Effort</i> Rate Dinamis

c. Membuat daftar kebutuhan non-fungsional

Adapun daftar kebutuhan non-fungsional yang ada pada aplikasi UCP untuk increment 3 ini yaitu sebagai berikut.

**Tabel 4.27 Daftar kebutuhan non-fungsional increment 3**

Kode	Penjelasan
KNF-02	Rancangan antar muka aplikasi yang <i>user friendly</i>
KNF-03	Adanya fasilitas pencarian untuk memudahkan pengguna mencari data dengan cepat

#### 4.4.2. Analisa Kebutuhan Aplikasi

a. Penambahan Fitur Entry Informasi Client dan Daftar Fitur

Pada perancangan di tahapan *increment 3*, terdapat penambahan fitur memasukan informasi client dan fitur dari aplikasi yang akan dilakukan estimasi menggunakan metode use case. Hal ini karena berdasarkan hasil diskusi yang dilakukan oleh peneliti sepakat bahwa dalam dokumen penawaran, daftar fitur-fitur aplikasi dan informasi dari client atau calon pelanggan dibuat secara otomatis dari sistem. Adapun acuannya yaitu pada hasil interview pada point IR-20

b. Fitur nilai ER(*Effort Rate*) dan distribusi aktivitas dinamis

*Effort Rate(ER)* atau *Productivity Factor(PF)* merupakan konstanta yang digunakan untuk menentukan nilai usaha disamping nilai UCP. Nilai usaha didapatkan dengan cara mengalihkan nilai ER dengan nilai UCP [48]. Nilai ER dapat digunakan secara default jika organisasi atau perusahaan tidak mempunyai data historis estimasi yang sudah dilakukan. Jika data historis ada, maka dapat digunakan sebagai acuan dalam menentukan nilai ER [48] . nilai ER didapat dari persamaan berikut ini.

$$ER = \frac{\text{ActualEfort}}{\text{UCP}} \quad (1)$$

Nilai ER dari hasil persamaan (1) disebut dengan *PostPF*. *ActualEffort* didapatkan dari jumlah anggota tim yang terlibat pada aktivitas pengembangan perangkat lunak dikalikan dengan jumlah waktu pengembangan perangkat lunak pada saat perangkat lunak selesai dikerjakan. Sedangkan nilai UCP merupakan nilai hasil perhitungan dengan metode UCP.

Pada aplikasi UCP ini nanti nilai ER didapatkan dari rata-rata dari *PostPF* pada setiap historis hasil estimasi yang sudah dilakukan. Proses pengambilan rata-rata didasarkan dari penggunaan framework tools yang ada. Hal ini dikarenakan, berdasarkan penelitian yang dilakukan oleh , nilai ER dari pembuatan aplikasi menggunakan CMS akan berbeda dengan nilai ER dri pembuatan aplikasi yang tidak menggunakan CMS [49]. Dalam pembangunan aplikasi berbasis web, DTS menggunakan framework CMS dan Codeigniter. Hasil nilai rata-rata tersebut nantinya digunakan sebagai faktor pengali untuk menentukan besaran usaha dan biaya pada estimasi selanjutnya.

Untuk fitur distribusi biaya dinamis dilakukan juga dengan melihat jenis estimasi aplikasi. Jadi, ada perbedaan antara distribusi aktivitas untuk pengembangan aplikasi dengan menggunakan CMS dengan tidak menggunakan CMS (Framework CI).

Sesuai dengan kesepakatan, Jika dalam membangun software tidak menggunakan CMS, maka perhitungan distribusi aktivitas akan mengikuti hasil penelitian yang dilakukan oleh Kesem Saleh [16]. Sedangkan untuk proyek yang menggunakan CMS, menggunakan distribusi aktivitas seperti pada tabel 4-28.

**Tabel 4.28 Distribusi Aktivitas pembangunan aplikasi dengan CMS**

NO.	Aktivitas	Presentase (%)
<b>1</b>	<b>Software Development</b>	
a	Requirement	22.16%
b	Specifications & Design	20.5%
c	Coding	12%
d	Testing	5%
<b>2</b>	<b>On Going Activity</b>	
a	Project Management	1.26%
b	Configuration Management	2.2%
c	Documentation	0.63%
d	Training & Support	3.77%
e	Acceptance & Deployment	1.57%
<b>3</b>	<b>Quality and Testing</b>	
a	Quality Assurance & Control	3.25%
	Total	<b>100%</b>

### c. Fitur Refine Nilai UCP

Fitur refine nilai UCP diperlukan untuk meminimalisir kesalahan dalam menentukan nilai UCP yang disebabkan oleh kesalahan dalam melakukan analisa ataupun *scope* yang berkembang. Contohnya, penambahan use case, penambahan transaksi pada use case, dan lain-lain. Hal ini dilakukan agar nantinya nilai ER hasil perhitungan dengan menggunakan persamaan (1) menjadi lebih akurat. Fitur ini digunakan setelah project aplikasi selesai dikerjakan. Fitur ini sama dengan fitur

perhitungan nilai UCP. Mulai dari menentukan UUCW sampai TCF.

d. Fitur *Actual Effort*

Fitur ini digunakan untuk memasukan nilai *Effort* sesungguhnya (*ActualEffort*). Nilai *Actual Effort* didapatkan dari jumlah anggota tim yang terlibat sesungguhnya pada setiap profesi dikalikan dengan waktu sesungguhnya yang dibutuhkan untuk membuat perangkat lunak. Tabel berikut ini merupakan bentuk formulir untuk memasukan nilai *ActualEffort*.

**Tabel 4.29** Formulir memasukan *actual Effort*

Profesi	Nama Pelaku	Total waktu aktual (hari)	Rata Waktu Aktual (jam)
<nama profesi1>	<nama pelaku1>	< waktu /hari	<jumlah waktu>
<nama profesi2>	<nama pelaku2>	< waktu /hari	<jumlah waktu>
<nama profesi3>	<nama pelaku3>	< waktu /hari	<jumlah waktu>
<nama profesin>	<nama pelakun>	< waktu /hari	<jumlah waktu>

Dari tabel diatas, terlihat bahwa *ActualEffort* merupakan *Effort* total yang dibutuhkan oleh masing-masing orang yang ada pada tim pengembang. Jumlah anggota tim pengembang berdasarkan hasil input pada saat estimasi. Oleh karena itu, untuk mengakomodir adanya perubahan anggota tim pengembang. Sebelum melakukan *entry Actual Effort* ini, yang harus dilakukan oleh tim pengembang yaitu melakukan refine hasil estimasi, yaitu dengan menambahkan anggota tim pengembang baru kedalam sistem. Setelah itu, membuka halaman form *ActualEffort*.

a. Fitur Memasukan Biaya Operasional

Biaya operasional merupakan biaya tambahan yang dikeluarkan diluar biaya pokok. Output biaya yang dihasilkan dari estimasi menggunakan metode UCP berupa biaya pokok dalam pembuatan aplikasi. Padahal dalam dokumen penawaran yang ada, terdapat biaya operasional dalam pembangunan aplikasi. Salah satu contoh biaya operasional yaitu biaya pembelian internet, dan biaya konsumsi. Oleh karena itu, dalam aplikasi ini terdapat fitur untuk memasukan biaya operasional. Besaran biaya operasional bersifat dinamis sesuai dengan hasil analisa yang dilakukan oleh tim pengembang.

c. Kebutuhan Non Fungsionalitas Sistem

Pengujian pada tahap *increment 2* dilakukan karena adanya penambahan fitur non-fungsional baru dan beberapa perubahan dari desain antarmuka pada saat release pada tahap *increment 2*. Adapun kebutuhan Non fungsionalitas dari aplikasi UCP ini yaitu sebagai berikut.

1. Kebutuhan kualitas aplikasi

Kebutuhan akan kualitas dari aplikasi UCP ini dengan dilakukan pengujian. Pengujian aplikasi UCP ini menggunakan 3 parameter sebagai berikut.

a. *Correctness testing*

Pengujian ini dilakukan untuk memastikan bahwa setiap fitur dan kebutuhan fungsional dalam aplikasi UCP sudah berjalan dengan baik dan benar sesuai dengan spesifikasi yang sudah didefinisikan dan disepakati dengan *client*

b. *Usability testing*

Pengujian ini dilakukan untuk memastikan bahwa interface aplikasi UCP tidak membingungkan dan tidak mengandung unsur ambiguitas sehingga pengguna aplikasi bisa dengan

mudah memahami alur aplikasi UCP dan mudah dalam menggunakan aplikasi UCP.

c. *Portability testing*

Pengujian ini dilakukan untuk memastikan bahwa aplikasi bisa berjalan dalam browser yang berdeda.

d. *Security testing*

Pengujian ini dilakukan untuk memastikan bahwa aplikasi aman dan tidak mudah untuk dilakukan hacking oleh orang yang tidak bertanggungjawab

Berikut ini adalah *tools* atau cara yang digunakan dalam pengujian apliasi ini ada pada tabel 4.30

**Tabel 4.30 Daftar tools pengujian kualitas aplikasi**

Kode	Kebutuhan pengujian	Tools
<b><i>Correctness</i></b>		
KNF-01	Aplikasi telah berjalan sesuai dengan kebutuhan yang telah dispesifikasikan	Pengujian dilakukan berdasarkan uji fungsionalitas sistem dengan menggunakan <i>Blackbox Testing</i> dan <i>Unit Testing</i>
<b><i>Usability</i></b>		
KNF-02	Rancangan antar muka aplikasi yang <i>user friendly</i>	Pengujian dilakukan berdasarkan delapan aturan desain antarmuka Ben Shneiderman.
<b><i>Portability</i></b>		
KNF-03	Aplikasi dapat dijalankan di browser yang berbeda	Pengujian dilakukan menggunakan <i>tools</i> Lunascape.
<b><i>Security</i></b>		
KNF-04	Aplikasi mempunyai mempunyai tingkat keamanan yang baik	

*Keterangan :*

*KNF-xx = Kebutuhan Non Fungsional pada urutan ke-xx*

2. Kebutuhan Sistem

Aplikasi UCP ini bisa dijalankan dengan bantuan sebuah browser internet yang up to date dan Sistem Operasi yang ada pada komputer/laptop atau pada OS smartphome (KNF-05).

3. Kebutuhan Kinerja

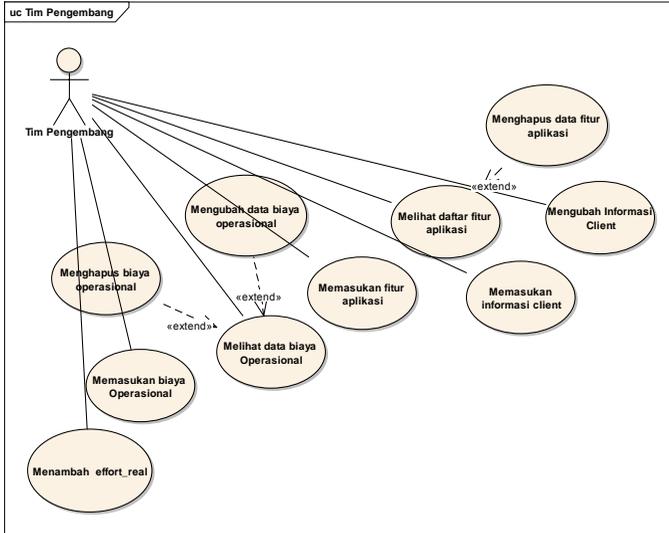
Apliksi UCP dapat diakses di lebih dari satu komputer atau laptop dan Device Smarphone(KNF-06).

4. Adanya fitur pencarian

Apliksi UCP dapat diakses di lebih dari satu komputer atau laptop dan Device Smarphone(KNF-07).

4.4.3. Desain Use Case Diagram

Berikut ini desain use case baru berdasarkan penambahan fitur yang ada pada tahap *increment 3*.



Gambar 4.25 Desain Use Case pada tahap *increment 3*

Gambar 4.25 merupakan desain use case baru yang ditambahkan pada use case utama sesuai dengan analisa kebutuhan yang sudah dilakukan sebelumnya.

#### 4.4.4. Use Case Description

Berikut ini adalah salah satu contoh bentuk *Use Case Description* yang ada pada tahap *increment ke-3*.

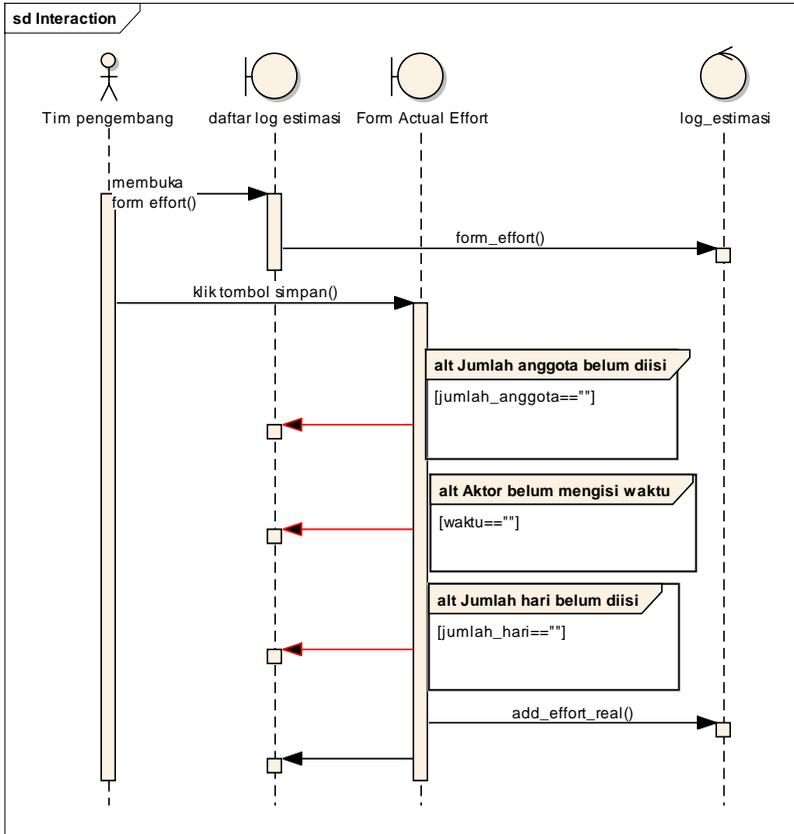
**Tabel 4.31 Memasukan *actual Effort***

+	UCD.01.34 Memasukan <i>actual Effort</i>										
<b>Ringkasan :</b>	<i>Use-case</i> ini digunakan menambahkan <i>actual Effort</i> aplikasi										
<b>Direct Aktor :</b>	Tim pengembang										
<b>Relasi antar use-case :</b>	-										
<b>Prioritas :</b>	Penting ( <i>essential</i> )										
<b>Frekuensi Penggunaan :</b>	Satu kali										
<b>Pre Condition :</b>	Melihat daftar data log estimasi, Login										
<b>Skenario Sukses Utama :</b>	<table border="1"> <thead> <tr> <th>No</th> <th>Aktor</th> <th>Sistem</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Aktor membuka form <i>actual Effort</i></td> <td>Sistem menampilkan form <i>actual Effort</i></td> </tr> <tr> <td>2</td> <td>Aktor memasukan waktu <i>actual</i> dalam satuan haripengembangan aplikasi pada setiap aktivitas</td> <td>Sistem melakukan pengecekan untuk memastikan semua field sudah terisi dengan benar</td> </tr> </tbody> </table>		No	Aktor	Sistem	1	Aktor membuka form <i>actual Effort</i>	Sistem menampilkan form <i>actual Effort</i>	2	Aktor memasukan waktu <i>actual</i> dalam satuan haripengembangan aplikasi pada setiap aktivitas	Sistem melakukan pengecekan untuk memastikan semua field sudah terisi dengan benar
No	Aktor	Sistem									
1	Aktor membuka form <i>actual Effort</i>	Sistem menampilkan form <i>actual Effort</i>									
2	Aktor memasukan waktu <i>actual</i> dalam satuan haripengembangan aplikasi pada setiap aktivitas	Sistem melakukan pengecekan untuk memastikan semua field sudah terisi dengan benar									

		pengembangan	
	3		Sistem melakukan perhitungan actual <i>Effort</i> berdasarkan hasil input aktor
	4		Sistem menyimpan data actual <i>Effort</i>
	5		Sistem menampilkan pesan “Data actual <i>Effort</i> berhasil disimpan”
	6		Sistem menampilkan halaman daftar log estimasi
<b>Skenario Alternatif:</b>	Alt No.2 Jika aktor tidak mengisi semua field yang tersedia, Sistem menampilkan pesan “Field tidak boleh kosong”		
	Alt No.2 Jika aktor tidak memasukan input dalam bentuk angka, Sistem akan menampilkan pesan “Input harus dalam bentuk angka”		

#### 4.4.5. Desain Sequence Diagram

Berikut adalah bentuk dari Desain *Sequence Diagram* untuk memasukkan *actual Effort*.

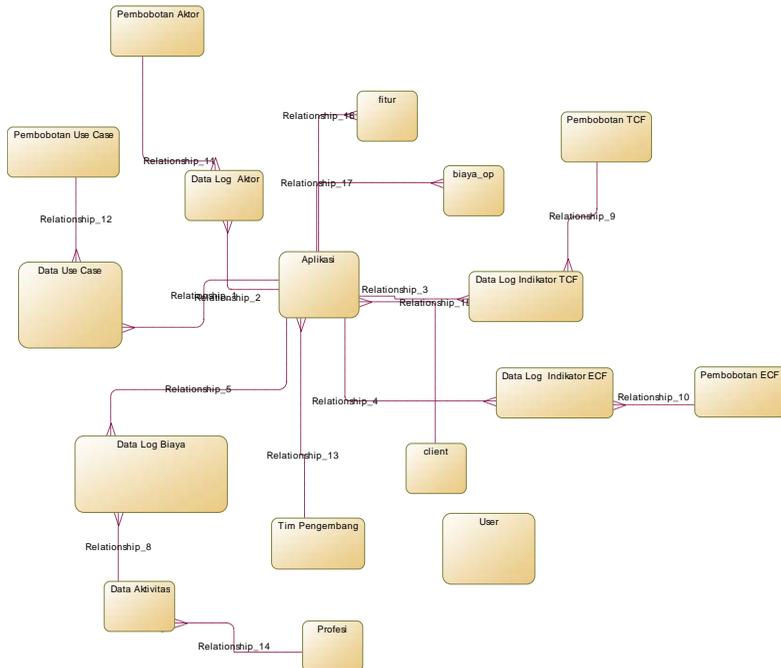


Gambar 4.26 *Sequence Diagram* Memasukan *actual Effort*

Dari gambar tersebut terlihat bahwa terdapat beberapa fungsi yang ada pada *Sequence Diagram* Memasukan *actual Effort*



#### 4.4.7. Conceptual Data Model(CDM)



**Gambar 4.28 Conceptual Data Model (CDM) Increment 3**

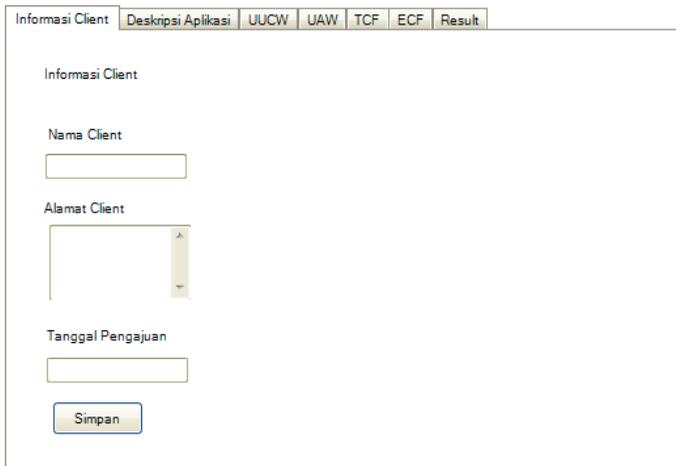
Gambar 4.28 merupakan desain dari Conceptual Data Model (CDM). Dengan adanya perubahan dan penambahan kebutuhan dalam pembagunan aplikasi Use Case Point ini, maka entitas yang ada pada CDM juga ikut berubah. Conceptual Data Model (CDM) digunakan sebagai acuan dalam pembuatan desain *Physical Data Model (PDM)*



#### 4.4.9. Desain Antarmuka Pengguna

Desain antarmuka aplikasi dilakukan berdasarkan hasil analisa yang ada. Berikut ini merupakan beberapa contoh desain antar muka dari aplikasi pada tahap *increment* 3.

a. Desain form Informasi client



The image shows a web browser window with a tab titled 'Informasi Client'. The browser's address bar contains the text 'Deskripsi Aplikasi UUCW UAW TCF ECF Result'. The main content area of the browser displays a form with the following elements:

- A title 'Informasi Client' at the top of the form.
- A label 'Nama Client' followed by a single-line text input field.
- A label 'Alamat Client' followed by a multi-line text area with a vertical scrollbar.
- A label 'Tanggal Pengajuan' followed by a single-line text input field.
- A 'Simpan' button located below the input fields.

**Gambar 4.30** Desain form informasi client

Gambar 4.30 merupakan bentuk dari antarmuka *form* informasi client. Pada bentuk antarmuka ini, terdapat beberapa field. Pertama yaitu nama client. Field ini berfungsi untuk memasukan data nama client. Kedua, field Alamat Client. Fungsinya yaitu untuk memasukan alamat dari client. Ketiga, tanggal pengajuan client. Fungsi dari field ini yaitu untuk memasukan tanggal pengajuan estimasi aplikasi

## b. Desain form deskripsi aplikasi

Informasi Client | Deskripsi Aplikasi | UUCW | UAW | TCF | ECF | Result

Form Deskripsi aplikasi

Nama Aplikasi

Nama Fitur

Penggunaan Framework  
 CMS  CodeIgniter

Anggota Pengembang  
 Pilih Anggota

<search>

No	Nama Fitur	Action

**Gambar 4.31 Form deskripsi aplikasi**

Gambar 4.31 merupakan bentuk desain dari form deskripsi aplikasi. Perbedaan dengan desain deskripsi aplikasi yang sebelumnya yaitu bahwa pada desain ini, terdapat juga form untuk memasukan list fitur-fitur yang ada dalam aplikasi nanti. list fitur ni nanti akan dicetak dalam dokumen penawaran. Form memaskannama fitur tampil setelah pengguna memasukan deskripsi aplikasi. Selain itu, terdapat tabel daftar fitur. Tabel ini bertujuan untuk menampilkan daftar fitur yang sudah dimasukan oleh pengguna. Pengguna bisa melakukan penghapusan terhadap fitur yang ada dengan mengklik tombol delete yang ada pada kolom *action*.

c. Desain form entry *Actual Effort*

Profesi	Nama Pelaku	Rata waktu/hari	Jumlah hari
Programmer	Faiz Fanani	<input type="text"/>	<input type="text"/>
Sistem Analis	Affas	<input type="text"/>	<input type="text"/>
Dokumentator	Sella Wr	<input type="text"/>	<input type="text"/>

Nilai Effort Real

**Gambar 4.32** Desain form entry *actual Effort*

Gambar 4.32 merupakan bentuk desain form entry *Actual Effort*. Form ini nanti akan ditampilkan setelah hasil estimasi disetujui dan project pembuatan aplikasi sudah selesai. Form ini terdiri dari beberapa kolom. Kolom pertama pada sebelah kiri merupakan kolom profesi. Kolom profesi menampilkan profesi yang mengembangkan aplikasi. Kolom nama pelaku berisikan nama pelaku pada masing-masing profesi. Rata waktu/hari meberisikan field untuk memasukan jumlah waktu yang diperlukan dalam bekerja mengembangkan aplikasi dalam satu hari. Kolom jumlah hari berisikan jumlah waktu yang dibutuhkan oleh masing-masing profesi dalam mengembangkan perangkat lunak.



dalam daftar list biaya operasional. Selain itu, terdapat tombol kirim untuk mengirimkan hasil estimasi kepada direktur. Dengan Fungsi ini direktur akan tahu kalau sedang ada proses estimasi harga perangkat lunak melalui email. Direktur nanti akan melakukan persetujuan setelah estimasi berhasil dilakukan.

*“Halaman ini sengaja dikosongkan”*

#### 4.4.10. Perancangan Pengujian

Perancangan pengujian pada tahapan increment 3 ini hampir sama dengan increment 1 dan 2, yaitu pengujian dilakukan terhadap kebutuhan fungsional dan kebutuhan non-fungsional.

a. Perancangan pengujian kebutuhan fungsional

- Identifikasi dan rencana pengujian

Berikut ini adalah beberapa contoh hasil identifikasi dan rencana pengujian pada tahap increment 3

**Tabel 4.32 Identifikasi Pengujian kebutuhan fungsional increment 3**

No	Kode KF	Kode Use Case	Nama Use Case	Pengujian	Bentuk pengujian	Identifikasi pengujian
1	KF19	UC.01.22	Memasukan Actual <i>Effort</i>	Skenario Normal	Black box	TC077
				Skenario Alternatif	Black box	TC078
2	KF20	UC.01.26	Memasukan informasi client	Skenario Normal	Black box	TC079
				Skenario Alternatif	Black box	TC080
		UC.01.27	Mengubah informasi Client	Skenario Normal	Black box	TC081
				Skenario Alternatif	Black box	TC082

- Prosedur pengujian  
Adapun prosedur pengujian yang ada yaitu ada pada tabel berikut ini.

Tabel 4.33 Prosedur Pengujian kebutuhan fungsional increment 3

No	Identifikasi pengujian	Skenario	Prosedur Pengujian	Masukan	Output Espektasi
3	TC080	Skenario Alternatif	<ol style="list-style-type: none"> <li>1. Login sebagai direktur atau sekretaris</li> <li>2. Membuka halaman informasi pelanggan(client) dengan cara menekan menu estimasi harga Tidak memasukan data informasi client sesuai dengan yang diminta sistem Menekan tombol simpan</li> </ol>	Username dan password untuk melakukan login	<ul style="list-style-type: none"> <li>• Sistem tidak menyimpan data informasi pelanggan kedalam database</li> <li>• Sistem menampilkan pesan “Belum diisi” pada field yang belum diisi</li> </ul>
4	TC081	Skenario normal	<ol style="list-style-type: none"> <li>1. Menyimpan data informasi pelanggan terlebih dahulu</li> </ol>	Input karakter	<ul style="list-style-type: none"> <li>• Sistem menyimpan hasil perubahan data</li> </ul>

No	Identifikasi pengujian	Skenario	Prosedur Pengujian	Masukan	Output Espektasi
			2. Melakukan pengubahahan terhadap salah satu atau lebih field input informasi pelanggan 3. Menekan tombol perbarui		informasi pelanggan kedalam database • Sistem menampilkan pesan “ Data informasi pelanggan berhasil diperbaharui”

b. Perancangan pengujian kebutuhan non-fungsional

- Identifikasi dan rencana pengujian

Berikut ini adalah beberapa contoh identifikasi dan rencana pengujian pada tahap increment 3 untuk kebutuhan non-fungsional

**Tabel 4.34 Identifikasi Pengujian kebutuhan non-fungsional increment 3**

No	Kode KNF	Kode Object Pengujian	Bentuk pengujian	Tools	Identifikasi Pengujian
2	KNF-02	AM01-AM24	Pengujian Useability	Pengujian dilakukan berdasarkan delapan aturan desain antar muka dari Ben Shneiderman	TC091
3	KNF-03	UC.02.13-UC.02.16 dan UC.03.04	Pengujian terhadap fitur pencarian	Pengujian dilakukan berdasarkan test Case	TC092
4	KNF-04-KNF-05	-	Pengujian portability	Pengujian dilakukan dengan cara menjalankan aplikasi pada browser yang berbeda	TC093

- Prosedur pengujian

Berikut ini adalah beberapa contoh prosedur pengujian yang ada pada tahap increment 3 untuk kebutuhan non-fungsional

**Tabel 4.35 Prosedur Pengujian kebutuhan non-fungsional increment 3**

No	Identifikasi Pengujian	Prosedur	Input	Output Ekspektasi
3	TC092	<ul style="list-style-type: none"> <li>• Memasukan input karakter pada fitur pencarian yang sama dengan karakter yang dicari</li> </ul>	Input karakter dalam bentuk text atau angka	<ul style="list-style-type: none"> <li>• Hasil pencarian menemukan hasil</li> </ul>
4	TC093	<ul style="list-style-type: none"> <li>• Memasukan input karakter yang tidak sama dengan karakter yang dicari</li> </ul>		<ul style="list-style-type: none"> <li>• Hasil pencarian tidak menemukan hasil</li> </ul>

*Halaman ini sengaja dikosongkan*

## **BAB V IMPLEMENTASI**

Bagian akan dijelaskan mengenai implementasi pengembangan aplikasi Use Case Point(UCP). Adapun subbab dari bab implementasi yaitu: tahapan implementasi, implementasi tahap *increment 1*, implementasi tahap *increment 2*, implementasi tahap *increment 3*.

### **5.1. Tahapan Implementasi**

Adapun tahapan yang dilakukan oleh peneliti dalam melakukan implementasi aplikasi UCP studi kasus perusahaan DTS sebagai berikut.

#### **5.1.1. Pembuatan Database**

Database dibuat berdasarkan desain model data fisik . Desain data model fisik yang sudah dibuat akan langsung degenerate kedalam bentuk database spesifik. Pada pembuatan aplikasi UCP ini, database yang dibuat dalam bentuk database MySQL.

#### **5.1.2. Pengkodean/Implementasi**

Pengkodean/implementasi dilakukan setelah database selesai dibuat. Pada pengkodean ini, peneliti menggunakan bahasa pemrograman PHP dengan framework CodeIgniter. Peneliti dalam melakukan pengkodean ini menggunakan konsep MVC yang kepanjangannya *Model View Controller*. Kode-kode program yang berhubungan langsung dengan database, seperti menjalankan query akan masuk pada folder/ bagian dari model. Sedangkan kode program yang berhubungan dengan tampilan seperti kode HTML akan masuk pada bagian *View*. Controller berisikan kode-kode program yang menghubungkan *View* dengan

*Model*. Berikut ini adalah salah satu contoh file kode yang ada pada aplikasi UCP.

### **5.1.3. Pengujian**

Pengujian dilakukan setelah pengkodean/implementasi selesai. Pengujian dilakukan berdasarkan kebutuhan fungsional dan kebutuhan non-fungsional. Pengujian kebutuhan fungsional dilakukan dengan menggunakan metode blackbox. Pengujian dengan menggunakan metode ini dilakukan dengan cara menguji scenario utama dan scenario alternative pada masing-masing use case. Sedangkan untuk pengujian kebutuhan non-fungsional dilakukan dengan menggunakan aturan Ben Sheinderman. Pengujian dengan menggunakan metode ini dilakukan dengan cara melakukan pengecekan antar muka aplikasi dengan 8 aturan desain antar muka dari Ben Sheinderman. Untuk kebutuhan non-fungsional yang berupa fitur pencarian, dilakukan dengan menggunakan metode blackbox sama dengan pengujian kebutuhan fungsional.

Berikut ini merupakan hasil dari tahapan implementasi aplikasi UCP yang terbagi dalam 3 tahapan (*increment*).

## **5.2. Implementasi Tahap *Increment* 1**

### **5.2.1. Desain Database**

Gambar 5.1 merupakan desain database dari aplikasi UCP pada *increment* 1. Desain database tersebut dibuat berdasarkan desain *Physical Data Model* (PDM) yang sudah dibuat pada tahap perencanaan. Database pada tahap *increment* 1 ini digunakan untuk memasukkan data hasil perhitungan dan juga data untuk melakukan perhitungan, mulai dari perhitungan UUCW sampai perhitungan *Effort*



b. Sampel kode aplikasi

Gambar 5.3 merupakan sample dari kode yang ada pada aplikasi\_model. Aplikasi\_model ini merupakan file yang berextensi .php

```

class Aplikasi_model extends CI_Model {
    function __construct()
    {
        parent::__construct();
        $this->CI = get_instance();
    }

    //insert anggota tim
    function insert_tim($data){
        $this->db->insert('tim',$data);
        return $this->db->affected_rows() ? true : false;
    }

    function select_max_id_tim(){
        //Select table name
        $query = $this->db->query("SELECT MAX(ID_TIM) as ID_TIM FROM tim ");
        return $query;
    }

    function insert_anggota_tim($data){
        $this->db->insert('anggota_tim',$data);
        return $this->db->affected_rows() ? true : false;
    }
}

```

**Gambar 5.3 sample kode user\_model**

Fungsi-fungsi yang ada pada Class *controller* tersebut didasarkan dari desain *Class Diagram* untuk kelas user\_model. user\_model ini merupakan class yang berfungsi sebagai penghubung antar aplikasi dengan database. user\_model berisikan kode-kode query untuk melakukan manipulasi maupun mengambil data dari database.

```

class Estimasi extends CI_Controller
{
    /**
     * Ini merupakan Sebuah controller untuk menghubungkan view uc dan model uc
     * Semua CRUD yang berhubungan dengan uc memanggil controller ini
     * Created by Mukhammad Faiz Fanani
     * V.1.0
     */
    public function __construct()
    {

    }

    public function index()
    {

    }

    public function form_aplikasi()
    {

        $sisi['anggota']      = $this->anggota_model->get_data_anggota();
        $sisi['kirim']       = $this->session->flashdata('kirim');

        $this->session->set_userdata('step',1);

        $sisi['step']       = $this->session->userdata('step');

        if($this->session->flashdata('pesan')!=""){
            $sisi['kirim']   = $this->session->flashdata('pesan');
        }

        $data['header']     = $this->load->view('backend/header');
        $data['menu_kiri']  = $this->load->view('backend/menu_kiri');
        $data['content']    = $this->load->view('content/form_aplikasi', $sisi);
        $this->load->view('/backend/main', $data);
    }
}

```

**Gambar 5.4** Sample kode controller estimasi

Gambar 5.4 merupakan bentuk sampel kode view daftar pengguna.

```

</ul>

<div id="myTabContent" class="tab-content">
<div class="tab-pane active" id="info">
<label class="control-label" for="selectError"><h3>Deskripsi Aplikasi</h3></label>

<?php
if(isset($kirim)){
    echo
<div class="alert alert-success">
<button type="button" class="close" data-dismiss="alert"></button>
<strong>Sisa digunakan!</strong> .<kirim.</div>
</div>
}
?>
<?php if($step=1){ echo 'form role="form" method="post" id="galauc" action="'.base_url().'admin/update';
echo 'form role="form" method="post" id="galauc" action="'.base_url().'admin/add_aplikasi' >;
?>

<div class="form-group">
<label class="control-label" for="inputSuccess1"> Nama Aplikasi</label>
<input type="text" name="nama_aplikasi" value=""?php if(isset($nama_aplikasi))echo $nama_aplikasi;?> <
<?php if(isset($erraplikasi))| echo '<p>font-size: 1em;</p>'.erraplikasi.</form/>?> }>
<input type="hidden" name="id_tin" value=""?php if(isset($id_tin))echo $id_tin;?> class="form-control.
</div>

```

**Gambar 5.5** Sampel kode view daftar pengguna

*“Halaman ini sengaja dikosongkan”*

### 5.2.3. Pengujian

Berikut ini merupakan bentuk laporan hasil pengujian yang dilakukan terhadap aplikasi UCP pada tahap *increment* 1. Adapun detail pengujian yang ada terdapat pada dokumen pengujian aplikasi.

Tabel 5.1 Pelaporan hasil pengujian tahap *increment* 2

No	Kode Uji	Skenario	Prosedur Pengujian	Masukan	Output Espektasi	Output	Kesimpulan
1	TC001	Skenario Normal	<ol style="list-style-type: none"><li>1. Membuka halaman login aplikasi</li><li>2. Kemudian memasukan username dan password yang sudah terdaftar dalam database untuk login.</li></ol>	Input karakter password dan username	Berhasil ke halaman login	Berhasil ke halaman login	Sesuai dengan ekspektasi
2	TC002	Skenario Alternatif	<ol style="list-style-type: none"><li>1. Membuka halaman login aplikasi</li><li>2. Memasukan username dan</li></ol>	Input karakter username dan password	Keluar pesan “ username atau password salah “	Keluar pesan “ username atau password salah “	Sesuai dengan ekspektasi

No	Kode Uji	Skenario	Prosedur Pengujian	Masukan	Output Espektasi	Output	Kesimpulan
			password yang tidak ada dalam database.				
3	TC003	Skenario Normal	Melakukan klik tombol logout	-	Masuk ke halaman login aplikasi	Masuk ke halaman login aplikasi	Sesuai dengan ekspektasi
4	TC004	Skenario Normal	<ol style="list-style-type: none"> <li>Memamasukan deskripsi aplikasi yang sesuai dengan perintah yang ada pada sistem</li> <li>Menekan tombol simpan</li> </ol>	Input karakter	Data berhasil disimpan dalam sistem dan menampilkan pesan “Deskripsi aplikasi berhasil disimpan”	Data berhasil disimpan dalam sistem dan menampilkan pesan “Deskripsi aplikasi berhasil disimpan”	Sesuai dengan ekspektasi



Deskripsi Aplikasi	UUCW	UAW	Technical Complexity Factor (TCF)	Environmental Complexity Factor (ECF)	Re
<b>Hasil Akhir Perhitungan Nilai UCP</b>			<b>Hasil Perhitungan Usa</b>		
Unadjusted Use Case Weight	: 5		Metode	: ER	
Unadjusted Actor Weight	: 1		Nilai Hour Effort	: 57	
Environmental Complexity Factor	: 0.66				
Technical Complexity Factor	: 0.74				
Use Case Point	: 2.886				
<b>Distribusi usaha dan biaya</b>					
Aktivitas	Effort		Biaya		
	%	Hour of Effort	Standard gaji/bulan	Gaji per Jam	Biaya
<b>SOFTWARE DEVELOPMENT</b>					
Requirements	7.5 %	4,33	Rp. 4.000.000,00	Rp. 25.000,00	Rp. 108.250,00
Specifications & Design	17.5 %	10,10	Rp. 4.000.000,00	Rp. 25.000,00	Rp. 252.500,00

**Gambar 5.7** interface hasil estimasi biaya

b. Sampel kode aplikasi

```

325 <td height="25" scope="row">
326 <div align="left">Nilai Hour Effort </div>
327 </td>
328 <td colspan="2"><strong></strong></td>
329 <td colspan="2"><input type="text" value="57" /></td>
330 </tr>
331 </table>
332 </div>
333 </div>
334 </section>
335 <label class="control-label" for="selectError">
336 <h3>Rekap Biaya</h3>
337 </label>
338 <div class="row">
339 <div class="col-md-7">
340 <h3>Distribusi biaya</h3>
341 <button type="submit" onclick="edit_biaya()" class="btn btn-warn">
342 <br>
343 </br>
344 <form method="post" action="<php echo base_url(); ?>estimasi/upda
345 <table width="600" border="1" class="table-bordered">
346 <tr>
347 <th width="25%" rowspan="2">Aktivitas</th>
348 <th width="25%" rowspan="2">Effort</th>
349 <th colspan="3">Biaya</th>
350 </tr>
351 <tr>
352 <th colspan="2"></th>
353 <th colspan="2"></th>
354 </tr>
355 <td width="118">Hour of Effort </td>
356 <td width="130">Standard gaji/bulan (Rp) </td>
357 <td width="160">Gaji per Jam (Rp) </td>
358 <td width="162">Biaya (Rp) </td>

```

**Gambar 5.8** Sampel kode view hasil estimasi

```

G:\Information System\Diagbox\XAMPP\public\ucapapptf\costimate\application\controllers\estimasi.php - Notepad
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

estimasi.php
1592 // NILAI SALIMASI USHAHA
1593 $distribusi_usaha = $this->aktivitas_model->get_data_akt
1594
1595 // melakukan pemrosesan apakah sudah dilakukan perubahan biaya
1596 //error handling agar tidak error pada saat input biaya pertama
1597 $check = $this->log_biaya_model->get_log_biaya($id_aplikasi)->nu
1598 $edit = false;
1599 if ($check != 0) {
1600     $edit = $this->log_biaya_model->get_log_biaya($id_aplikasi)-
1601 }
1602
1603 if ($edit == 0 || $ubah_uap == true) {
1604     //delete previous biaya if exist
1605
1606     $this->log_biaya_model->delete_current_log_biaya($id_aplikasi)
1607     foreach ($distribusi_usaha->result() as $row) {
1608
1609         $nilai_usaha_aktivitas = round($row->PRESENTASE_USAHA *
1610 $gaji_per_jam = round($row->GAJI_PER_BULAN / 12
1611 $biaya_aktivitas = round($nilai_usaha_aktivitas *
1612
1613
1614
1615         $databiaya = array(
1616             'ID_APLIKASI' => $id_aplikasi,
1617             'ID_AKTIVITAS' => $row->ID_AKTIVITAS,
1618             //'PRESENTASE_USAHA' => $row->PRESENTASE,
1619             'NILAI_USAHA' => $nilai_usaha_aktivitas,
1620             'GAJI_PER_JAM' => $gaji_per_jam,
1621             'BIAYA_AKTIVITAS' => $biaya_aktivitas
1622
1623         );
1624
1625
1626
1627     $this->log_biaya_model->insert_log_biaya($databiaya);

```

Gambar 5.9 kode perhitungan biaya pada *controller* estimasi

```

log_biaya_model.php
1 <?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Log_biaya_model extends CI_Model {
4
5     function __construct()
6     {
7         parent::__construct();
8         $this->CI = get_instance();
9     }
10
11     function delete_current_log_biaya($id_aplikasi){
12         $this->db->where('ID_APLIKASI', $id_aplikasi);
13         $this->db->delete('log_biaya');
14         return $this->db->affected_rows() > true ? false;
15     }
16
17     //insert log estimasi
18     function insert_log_biaya($data){
19         $this->db->insert('log_biaya',$data);
20         return $this->db->affected_rows() > true ? false;
21     }
22     //get_log_biaya
23
24     function get_log_biaya($id_aplikasi){
25         $query = $this->db->query('SELECT * from log_biaya as lb
26 INNER JOIN AKTIVITAS as g ON lb.ID_AKTIVITAS =g.ID_AKTIVITAS
27 INNER JOIN gajipjml as p ON p.ID_PROFESI=g.ID_PROFESI
28 WHERE ID_APLIKASI='.$id_aplikasi.'-
29 ORDER BY g.ID_AKTIVITAS ASC');
30         return $query;
31     }

```

Gambar 5.10 sampel kode pada *model* log\_biaya\_model

*Halaman ini sengaja dikosongkan*

### 5.3.3. Pengujian

Pengujian yang dilakukan dalam tahap increment 2, selain dilakukan berdasarkan tambahan fitur yang ada pada tahap ini, juga dilakukan pengujian dengan acuan atau prosedur pengujian yang ada pada tahap *increment 1*. Jadi scenario/prosedur pengujian yang ada pada tahap *increment 1* dilakukan kembali pada tahap increment 2 disamping penambahan scenario pengujian baru pada tahap *increment 2*. Berikut ini merupakan salah satu contoh bentuk pelaporan hasil pengujian pada tahap *increment 2*.

**Tabel 5.2** Pelaporan hasil pengujian tahap increment 2

No	Kode Uji	Skenario	Prosedur Pengujian	Masukan	Output Espektasi	Output	Kesimpulan
3	TC003	Skenario Normal	Melakukan klik tombol logout	-	Masuk ke halaman login aplikasi	Masuk ke halaman login aplikasi	Sesuai dengan ekspektasi
4	TC004	Skenario Normal	1. Memamasukan deskripsi aplikasi yang sesuai dengan perintah yang ada pada sistem 2. Menekan	Input karakter	Data berhasil disimpan dalam sistem dan menampilkan pesan “Deskripsi aplikasi	Data berhasil disimpan dalam sistem dan menampilkan pesan “Deskripsi aplikasi	Sesuai dengan ekspektasi

No	Kode Uji	Skenario	Prosedur Pengujian	Masukan	Output Espektasi	Output	Kesimpulan
			tombol simpan		berhasil disimpan”	berhasil disimpan”	
5	TC005	Skenario Alternatif	<ol style="list-style-type: none"> <li>1. Tidak mengisi semua field yang ada</li> <li>2. Menekan tombol simpan</li> </ol>	Tidak ada	Keluar pesan “field belum diisi” pada <i>field</i> yang belum diisi	Keluar pesan “field belum diisi” pada <i>field</i> yang belum diisi	Sesuai dengan ekspektasi
6	TC006	Skenario Normal	<ol style="list-style-type: none"> <li>1. Mengubah masing-masing data yang ada pada field deskripsi aplikasi</li> <li>2. Menekan tombol perbaharui</li> </ol>	Input karakter	Data hasil perubahan tersimpan dalam database dan menampilkan pesan “Data berhasil perbaharui”	Data hasil perubahan tersimpan dalam database dan menampilkan pesan “Data berhasil perbaharui”	Sesuai dengan ekspektasi



### 5.4.2. Sample Implementasi Aplikasi

Pada bagian ini akan diberikan beberapa contoh kode dari hasil implementasi aplikasi.

a. Antarmuka form informasi pelanggan

The screenshot shows a web application interface for a software price estimation form. The title bar reads '\$Form Estimasi Harga Perangkat Lunak'. Below the title is a navigation menu with tabs: 'Informasi Pelanggan' (selected), 'Deskripsi Aplikasi', 'UUCW', 'UAW', 'TCF', 'ECF', and 'Result'. The main content area is titled 'Informasi Pelanggan' and contains three required input fields:

- Nama Client\***: A text input field with the placeholder 'Masukan nama client' and an example 'Contoh: Bpk Denny'.
- Alamat Client\***: A text input field with the placeholder 'Masukan alamat client'.
- Tanggal Pengajuan Aplikasi\***: A text input field with the placeholder 'Masukan tanggal pengajuan' and an example 'Contoh: 20-05-2015'.

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

**Gambar 5.12** Antarmuka informasi pelanggan

Gambar 5.12 merupakan bentuk antarmuka form informasi pelanggan. *Form* informasi pelanggan ini digunakan untuk memasukkan informasi pelanggan dari aplikasi yang akan diestimasi. Data informasi pelanggan ini nantinya digunakan pada saat mencetak dokumen penawaran

## c. Sampel kode aplikasi

```

40      $('#first_fitur').dataTable({
41          "displayLength": 5,
42          "aLengthMenu": [[5, 10, 25, 50, -1], [5, 10, 25, 50, "All*"]
43      ]});
44  });
45  </script>
46  <div class="row">
47  <div class="box col-md-12">
48  <div class="box-inner">
49  <div class="box-header well" data-original-title="">
50  <h2><i class="glyphicon glyphicon-sad"></i>Form Estimasi Harga Peringkat Lunak </h2>
51  <!--
52  <div class="box-icon">
53  <a href="#"><?php echo base_url(); ?>" class="btn btn-setting btn-round btn-default"><i
54  class="glyphicon glyphicon-cog"></i></a>
55  <a href="#"><?php echo base_url(); ?>" class="btn btn-minimize btn-round btn-default"><i
56  class="glyphicon glyphicon-chevron-up"></i></a>
57  <a href="#"><?php echo base_url(); ?>" class="btn btn-close btn-round btn-default"><i
58  class="glyphicon glyphicon-remove"></i></a>
59  </div>
60  <!--
61  </div>
62  <div class="box-content">
63  <ul class="nav nav-tabs">
64  <?php if($step >= 1) ?>
65  <li class="active" ><a href="#"><?php echo base_url(); ?>estimasi/form_edit_client/<?php echo $this->sessi
66  <?php : else ( ?>
67  <li class="active" ><a href="#">Informasi Pelanggan<a></li>
68  <?php : ?>
69  <?php if($step >= 2) ?>
70  <li ><a href="#"><?php echo base_url(); ?>estimasi/form_edit_aplikasi/<?php echo $this->session->userdata
71  <?php : ?>
72  </ul>
73  </div>

```

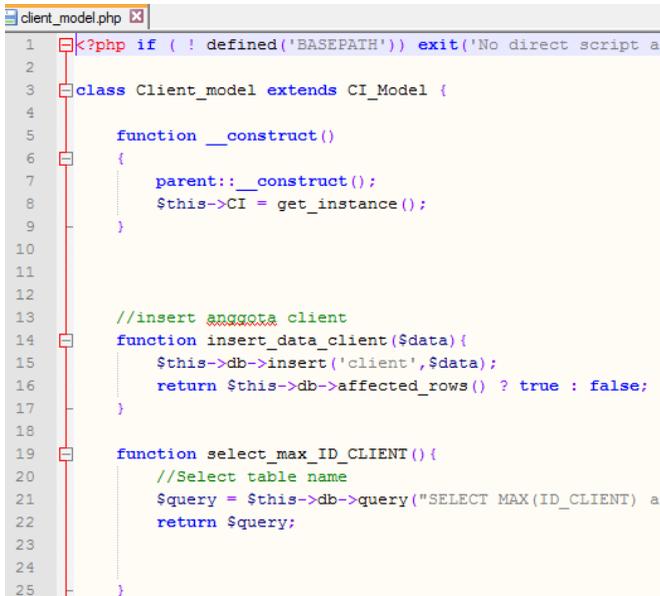
Gambar 5.13 Sampel kode *view* informasi pelanggan

```

4  exit('No direct script access allowed');
5  class Log_estimasi extends CI_Controller
6  {
7  /**
8   * Ini merupakan sebuah controller untuk menghubungkan view ke dan mode
9   * semua CRUD yang berhubungan dengan ke menangani controller ini
10  * Created by Mukhamad Faiz Fanani
11  * V.1.0
12  */
13  public function __construct()
14  {
15  parent::__construct();
16  $this->load->helper(array(
17  'form',
18  'url'
19  ));
20
21  $this->load->model('aplikasi_model');
22
23  $this->load->model('log_estimasi_model');
24  $this->load->model('log_biaya_model');
25  $this->load->model('fitur_model');
26  $this->load->model('aktivitas_model');
27
28
29
30  $this->load->library('session');
31  if ($this->session->userdata('username') == NULL) {
32
33  // redirect('login');
34  }
35
36  }

```

Gambar 5.14 Sampel kode *controller* Log\_estimasi



```

1  <?php if ( ! defined('BASEPATH')) exit('No direct script a
2
3  class Client_model extends CI_Model {
4
5      function __construct()
6      {
7          parent::__construct();
8          $this->CI = get_instance();
9      }
10
11
12
13      //insert anggota client
14      function insert_data_client($data){
15          $this->db->insert('client',$data);
16          return $this->db->affected_rows() ? true : false;
17      }
18
19      function select_max_ID_CLIENT(){
20          //Select table name
21          $query = $this->db->query("SELECT MAX(ID_CLIENT) a
22          return $query;
23
24
25      }

```

Gambar 5.15 Sampel kode model Client\_model

### 5.4.3. Pengujian

Berikut ini adalah salah satu hasil pelaporan pengujian yang dilakukan pada tahapn increment 3 yang terbagi menjadi dua, yaitu pengujian kebutuhan fungsional dan non-fungsional. Detail pengujian ada pada dokumen pengujian aplikasi.

a. Pelaporan pengujian kebutuhan fungsional

**Tabel 5.3 Pelaporan pengujian incement 3**

Kode Pengujian	Skenario	Prosedur Pengujian	Input	Output Ekspektasi	Hasil	Kesimpulan
TC066	Skenario Alternatif	<ol style="list-style-type: none"> <li>1. Login sebagai direktur atau sekretaris</li> <li>2. Membuka halaman informasi pelanggan(client) dengan cara menekan menu estimasi harga</li> <li>3. Tidak memasukan data informasi client sesuai dengan yang diminta sistem Menekan tombol simpan</li> </ol>	Username dan password untuk melakukan login	<ul style="list-style-type: none"> <li>• Sistem tidak menyimpan data informasi pelanggan kedalam database</li> <li>• Sistem menampilkan pesan “Belum diisi” pada field yang belum diisi</li> </ul>	<ul style="list-style-type: none"> <li>• Sistem tidak menyimpan data informasi pelanggan kedalam database</li> <li>• Sistem menampilkan pesan “Belum diisi” pada field yang belum diisi</li> </ul>	Sesuai dengan ekspektasi
TC067	Skenario normal	<ol style="list-style-type: none"> <li>1. Menyimpan data informasi</li> </ol>	Input karakter	<ul style="list-style-type: none"> <li>• Sistem menyimpan hasil</li> </ul>	<ul style="list-style-type: none"> <li>• Sistem menyimpan</li> </ul>	Sesuai dengan ekspektasi

Kode Pengujian	Skenario	Prosedur Pengujian	Input	Output Ekspektasi	Hasil	Kesimpulan
		<p>pelanggan terlebih dahulu</p> <ol style="list-style-type: none"> <li>2. Melakukan pengubahan terhadap salah satu atau lebih field input informasi pelanggan</li> <li>3. Menekan tombol perbarui</li> </ol>		<p>pengubahan data informasi pelanggan kedalam database</p> <ul style="list-style-type: none"> <li>• Sistem menampilkan pesan “ Data informasi pelanggan berhasil diperbaharui”</li> </ul>	<p>hasil pengubahan data informasi pelanggan kedalam database</p> <ul style="list-style-type: none"> <li>• Sistem menampilkan pesan “ Data informasi pelanggan berhasil diperbaharui”</li> </ul>	

b. Pelaporan pengujian kebutuhan non fungsional

Berikut ini adalah beberapa contoh dari pelaporan hasil pengujian kebutuhan non fungsional

**Tabel 5.4 Hasil pengujian pada tahap increment 3**

Kode KNF	Kode Object Pengujian	Bentuk pengujian	Tools	Identifikasi Pengujian	Hasil Pengujian
KNF-02	AM01-AM23	Pengujian Useability	Pengujian dilakukan berdasarkan delapan aturan desain antar muka dari Ben Shneiderman	TC071	
KNF-03	-	Pengujian terhadap fitur pencarian	Pengujian dilakukan berdasarkan test Case	TC072	Fitur pencarian bisa berjalan degan baik
KNF-06	Aplikasi UCP	Security Testing	Pengujian dilakukan dengan menggunakan tools <i>accunetix</i>	TC075	Hasil pengujian menunjukkan bahwa keamanan website menurut hasil scanning yang dilakukan oleh accunetix menunjukkan pada level 2.(medium). Jadi level tersebut dikatakan pada level yang cukup aman (middle)

*"Halaman ini sengaja dikosongkan"*

Berikut ini adalah pelaporan hasil pengujian antarmuka dengan menggunakan beberapa indikator.

1. *Strive for Consistency*

**Tabel 5.5 Indikator pengujian Strive for Consistency**

<b>Indikator pengujian</b>	<b>Hasil</b>
Aplikasi memiliki perintah konsisten dan memiliki kesamaan dalam setiap kondisi	√
Menggunakan tombol/button yang sama untuk menu,info, pesan bantuan, maupun fungsi-fungsi yang lainnya	√

2. *Cater to Universal Usability*

**Tabel 5.6 Indiaktor pengujian Cater to Universal Usability**

<b>Indikator pengujian</b>	<b>Hasil</b>
Memungkinkan perubahan dari isi konten	v
Adanya fungsi untuk pengguna ahli dan pengguna awal	-
Adanya pembeda antara pengguna seperti umur, diasbelitias, dan pembedaan teknologi,dll	v

*“Halaman ini sengaja dikosongkan”*

## BAB VI

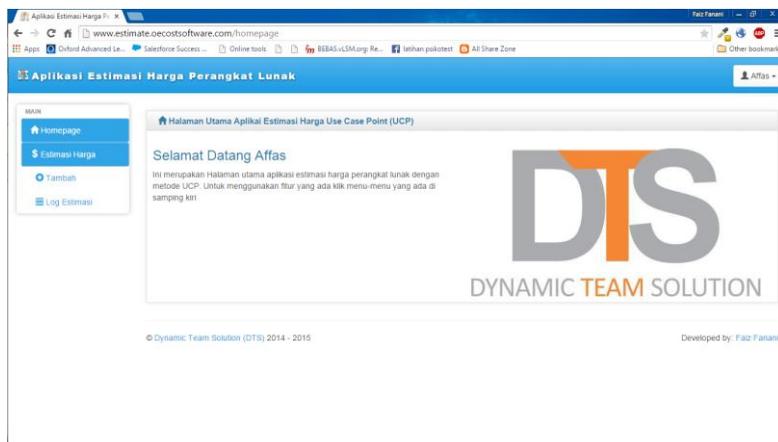
### HASIL DAN PEMBAHASAN

Bagian ini menjelaskan hasil dan pembahasan dari pengembangan aplikasi UCP studi kasus perusahaan DTS.

#### 6.1. Aplikasi Use Case Point (UCP)

Hasil dari rancang bangun aplikasi Use Case Point(UCP) ini yaitu berupa aplikasi UCP yang dibuat melalui 3 tahapan *increment*. Berikut ini merupakan beberapa contoh gambar bentuk interface aplikasi Use Case Point(UCP) selama 3 kali tahapan increment.

- Halaman awal (homepage)  
Halaman awal homepage ini merupakan halaman ketika pengguna sudah melakukan login dan berhasil.



Gambar 6.1 Halaman awal (*Homepage*) aplikasi UCP

- Form Informasi pelanggan  
Form ini bertujuan untuk memasukan informasi pelanggan (*client*) dari aplikasi yang akan diestimasi.

Gambar 6.2 form informasi pelanggan

- Halaman log estimasi

Halaman log estimasi berfungsi untuk menampilkan semua aplikasi yang sudah dilakukan estimasi baik aplikasi yang dibangun menggunakan CMS maupun aplikasi yang dibangun menggunakan *Framework*.

No	Tanggal estimasi	Nama Client	Nama Aplikasi	UUCW	UAW	TCF	ECF	Effort Estimate	Effort Real	Biaya Estimasi (Total)	Tim Pengembang	Status	Actions
1	04-06-2015	Sudaharto	Aplikasi Coba Final	5	2	0.81			0.00	Rp. 0.00	Tim Aplikasi	Belum lengkap	Edit
2	30-05-2015	Mr. A	Aplikasi penSeleksi karakter manusia	30	6	1.3	1.265	49.73	11.00	Rp. 1.515.349,75	Tim Aplikasi penSeleksi karakter manusia	Penutup	Edit Project Goal
3	25-05-2015	Aplikasi Sabun	Aplikasi Coba Final	10	1	0.76	0.66	4.76	0.00	Rp. 135.312,50	Tim Aplikasi Coba Final	Penutup	Edit

Gambar 6.3 Halaman daftar log estimasi

## 6.2. Verifikasi

Verifikasi digunakan untuk memastikan bahwa hasil perancangan sesuai dengan kebutuhan dari client. Proses verifikasi dilakukan pada setiap selesai siklus *increment* yang ada dalam pengembangan perangkat lunak UCP. Verifikasi dilakukan dengan cara membuat dokumen Berita Acara Pemeriksaan (BAP) dan *acceptance checklist* yang berisikan fitur-fitur yang sudah direview dan disetujui oleh client. Adapun *acceptance checklist* pada bagian lampiran.

## 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 perangkat lunak yang sudah dibuat dengan skenario yang sudah disusun. Untuk menentukan presentase kesesuaian antara kebutuhan yang sudah dispesifikasikan dengan kebutuhan sistem dapat menggunakan rumus sebagai berikut.

$$\% \text{ kesesuaian} = \frac{\sum \text{kebutuhan dimplementasikan}}{\sum \text{kebutuhan dispesifikasikan}} \times 100\%$$

Dari hasil validasi menunjukkan bahwa dari 23 kebutuhan fungsional (KF) yang sudah dispesifikasikan, sudah diimplementasikan semua. Sehingga presentase kesesuaian dapat ditentukan dengan

$$\% \text{ kesesuaian} = \frac{23}{23} \times 100\% = 100\%$$

Adapun hasil dari pengujian terdapat pada dokumen pengujian. Kesesuaian kebutuhan fungsional sistem dapat dicapai 100% karena pada setiap hasil *increment* yang sudah dilalui dilakukan perbaikan-perbaikan penyesuaian, baik dari segi kebutuhan maupun dari segi perancangan. Pada setiap tahapan juga memungkinkan untuk dilakukan *reverse engineering* untuk menyesuaikan hasil perancangan dan implementasi. Hasil *reverse*

engineering yang sudah dibuat akan dijadikan acuan pada tahap increment selanjutnya. Namun, untuk kebutuhan non fungsional (KNF). Namun untuk desain antarmuka, dari 34 desain antarmuka yang sudah dibuat, 95% terimplementasi tepat sesuai dengan desain. Hal ini karena terdapat beberapa improvisasi pada bentuk desain antarmuka sesuai dengan hasil pengujian desain antarmuka dengan 8 aturan Ben Sheniderman. Misalnya, ada penambahan *hint* pada setiap field untuk memasukan data yang bertujuan untuk memberikan contoh/panduan bagaimana cara mengisikan data pada field tersebut.

#### 6.4. Data Percobaan

Data percobaan ini menggunakan data proyek pengembangan perangkat lunak yang sudah dikerjakan oleh DTS. Proyek ini dengan SubKontraktor dengan perusahaan lain. Adapun deskripsi umum dari proyek ini yaitu sebagai berikut.

- Deskripsi umum proyek

**Tabel 6.1 Deskripsi umum proyek**

Nama Proyek	Pembuatan Aplikasi SIM Pelayanan Publik Disperdagin modul Tanda Daftar Industri (TDI)
Jumlah Tim Pengembang	5 orang
Client	Disperdagin Pemerintah Kota Surabaya
Kontraktor	PT. Trust Solution
Sub Kontraktor	CV. Dynamic Team Solution
Lama pengerjaan Proyek	6 bulan
Nilai Proyek	30.000.000

- Data Actual *Effort* proyek percobaan  
Berikut ini adalah data actual *Effort* dari pengerjaan aplikasi TDI selama 6 bulan kerja
  - 1 bulan = 22 hari
  - 1 minggu= 5 hari kerja
  - 1 hari = +/- maksimal 8-9 jam kerja

**Tabel 6.2 Data Actual Effort aplikasi TDI**

Profesi	bulan	Orang	waktu aktual (hari)	Waktu Aktual (jam)	Effort
System Analyst	4	1	88	8	704
Junior Programmer	2	1	44	9	792
Software Tester	0.5	1	11	8	88
PM	6	1	132	5	660
<b>Total Effort</b>					<b>2244</b>

- Data *use case* aplikasi percobaan

**Tabel 6.3 Data transaksi pada setiap use case**

Kode Use Case	Nama Use Case	Jumlah Transaksi
UC-01.01	Menambah data pra-permohonan TDI	3
UC-01.02	Mengunggah dokumen persyaratan TDI	3
UC-01.03	Mengubah data permohonan TDI	2
UC-01.04	Melihat detil data permohonan TDI	2
UC-01.05	Mencetak bukti permohonan TDI	2
UC-01.06	Memberikan notifikasi via SMS	2
UC-01.07	Men- <i>generate</i> nomor masuk pendaftaran TDI	4
UC-01.08	Memverifikasi antara data pra-permohonan TDI dengan dokumen asli	3
UC-01.09	Menambah data pemeriksaan lapangan	2
UC-01.10	Mengubah data pemeriksaan lapangan	2
UC-01.11	Men- <i>generate</i> nomor Surat Pernyataan Tugas	2
UC-01.12	Mencetak Surat Pernyataan Tugas	2
UC-01.13	Mengunggah <i>file</i> hasil pemeriksaan lapangan (BAP)	4

<b>Kode Use Case</b>	<b>Nama Use Case</b>	<b>Jumlah Transaksi</b>
UC-01.14	Memverifikasi antara BAP dengan data permohonan TDI	2
UC-01.15	Membuat draft Surat Keputusan (SK) Perijinan TDI	4
UC-01.16	Mencetak draft SK Perijinan TDI	3
UC-01.17	Mengubah draft SK Perijinan TDI	4
UC-01.18	Men-generate nomor SK Perijinan TDI	4
UC-01.19	Mencetak SK Perijinan TDI	3
UC-01.20	Memberikan status permohonan TDI oleh Kepala Dinas	6
UC-01.21	Memberikan status permohonan TDI oleh Sekretaris Dinas	6
UC-01.22	Memberikan status permohonan TDI oleh Kabid Industri	6
UC-01.23	Memberikan status permohonan TDI oleh Kasi ILMEA/IKAHH	6
UC-01.24	Men-generate nomor keluar berdasarkan hasil realisasi perijinan TDI	2
UC-01.25	Merekap data TDI per periode	4
UC-01.26	Mencetak laporan data industri yang masuk per periode	3
UC-01.27	Mencetak laporan realisasi penyelesaian perijinan TDI per periode	3
UC-01.28	Merekap perijinan TDI per periode	4
UC-01.29	Mencetak surat keterangan penolakan permohonan TDI per periode	4
UC-01.30	Mencetak surat keterangan persetujuan permohonan TDI per periode	4

Kode Use Case	Nama Use Case	Jumlah Transaksi
UC-01.31	Mengirim status permohonan TDI via SMS	1
UC-01.32	Memperpanjang perijinan TDI	3
UC-01.33	Mengubah Perijinan TDI	3
UC-01.34	Menutup perijinan TDI	3
UC-01.35	Mutasi keluar TDI	3

- Data aktor aplikasi percobaan

**Tabel 6.4 Data aktor aplikasi sim pelayanan publik**

Aktor	jml Aktor
Admin	1
Kepala Dinas	1
Sekretaris	1
Kepala Bidang	1
kepala Seksi	1
Operator Bidang	1
Petugas Lapangan	1
UPTSA	1
Pemohon	1
<b>Total</b>	<b>9</b>

- Data Technical Complexity Factor (TCF)

Data penilaian TCF ini diambil dari hasil diskusi dengan tim yang membangun aplikasi TDI

**Tabel 6.5 Data Penilaian TCF**

No	Faktor Teknis	Nilai
1	Distributed System Required	2
2	Response Time Is Important	4
3	End User Efficiency	4
4	Complex Internal Processing Required	3
5	Reusable Code Must Be A Focus	2
6	Installation Easy	4
7	Usability	4
8	Cross-Platform Support	3
9	Easy To Change	3
10	Highly Concurrent	3
11	Custom Security	3
12	Dependence On Thrid-Part Code	3
13	User Training	4

- Data Enviromental Complexity Factor (ECF)

Data ECF ini didapatkan dari penilaian tim pengembaang aplikasi TDI

**Tabel 6.6 Data hasil penilaian ECF**

No	Indiaktor ECF	Nilai
1	Familiarity with the Project	3
2	Application Experience	0
3	OO Programming Experience	4
4	Lead Analyst Capability	4
5	Motivation	4
6	Stable Requirements	4
7	Part Time Staff	3
8	Difficult Programming Language	3

## 6.5. Perhitungan Biaya Dengan Aplikasi

Setelah data diperoleh, tahapan selanjutnya yang dilakukan yaitu memasukan data kedalam aplikasi UCP. Berikut ini adalah hasil dari memasukan data kedalam aplikasi.

## 6.6. Hasil Percobaan Menggunakan Aplkasi UCP

- Perhitungan UUCW

No	Nama Use Case	Jumlah Transaksi	Actions
1	Menambah data pra-pemohonan TDI	3	<a href="#">Ubah</a> <a href="#">Hapus</a>
2	Mengunggah dokumen persyaratan TDI	3	<a href="#">Ubah</a> <a href="#">Hapus</a>
3	Mengubah data permohonan TDI	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
4	Melihat detil data permohonan TDI	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
5	Memberikan notifikasi via SMS	2	<a href="#">Ubah</a> <a href="#">Hapus</a>

Showing 1 to 5 of 35 entries [Previous](#) [Next](#)

Rekapitulasi Perhitungan Use Case:

Simple : 21  
Average : 14  
Complex : 0

Nilai UUCW:

[Selanjutnya](#)

**Gambar 6.4** Perhitungan UUCW dengan aplikasi

Gambar 6.4 menunjukkan daftar use case yang sudah dilakukan perhitungan nilai bobotnya. Masing-masing use case yang sudah dimasukan akan secara otomatis dikategorikan nilai kompleksitasnya yang didasarkan atas jumlah transaksi yang ada pada setiap use case. Pada bagan tersebut, terdapat juga jumlah ue case yang terklasifikasi pada masing-masing kategori dan juga nilai UUCW.

- Hasil perhitungan UAW

Daftar aktor aplikasi

Show 5 entries

Search:

No	Nama Aktor Aplikasi	Kategori	Bobot	Actions
1	Admin	Complex	3	<a href="#">Edit</a> <a href="#">Delete</a>
2	Kepala Dinas	Complex	3	<a href="#">Edit</a> <a href="#">Delete</a>
3	Sekretaris	Complex	3	<a href="#">Edit</a> <a href="#">Delete</a>
4	Kepala Bidang	Complex	3	<a href="#">Edit</a> <a href="#">Delete</a>
5	kapala Seksi	Complex	3	<a href="#">Edit</a> <a href="#">Delete</a>

Showing 1 to 5 of 9 entries. [Previous](#) [Next](#)

Rekapitulasi perhitungan aktor

Simple : 0

Average : 0

Complex : 9

Nilai UAW :

**Gambar 6.5 Hasil perhitungan UAW**

Gambar 6.5 merupakan hasil dari perhitungan kompleksitas aktor yang dilakukan oleh sistem. Pada bagian tersebut, aktor-aktor yang sudah terdaftar dalam sistem terkategoriisasi berdasarkan nilai kompleksitas dari aktor tersebut.

- Rekapitulasi keseluruhan perhitungan

Perangkat Lunak

Form Estimasi Harga Perangkat Lunak

Informasi Client | Deskripsi Aplikasi | UUCW | UAW | TCF | ECF | Result | Effort Real

**Hasil Akhir Perhitungan Nilai UCP**

Unadjusted Use Case Weight : 245

Unadjusted Actor Weight : 27

Environmental Complexity Factor : 0.935

Technical Complexity Factor : 1.03

Use Case Point : 261.9496

**Hasil Perhitungan Usaha**

Nilai Effort Rate : 8.23

Nilai Hour Effort : 2155.845208

Rekap Biaya

**Gambar 6.6 Hasil perhitungan *Effort***

Gambar 6.6 merupakan hasil rekapitulasi perhitungan *Effort* yang sudah dilakukan oleh sistem Pada halaman tersebut juga ditampilkan rekapitulasi hasil perhitungan UCP mulai dari perhitungan kompleksitas Use Case sampai dengan perhitungan kompleksitas faktor lingkungan

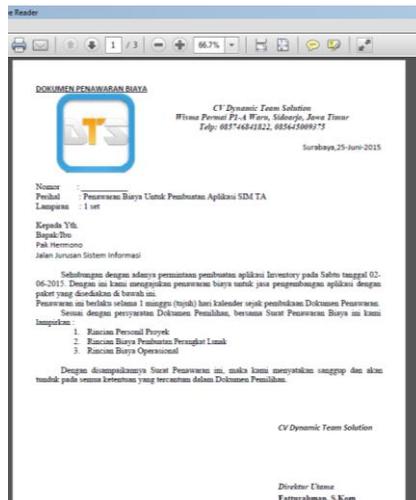
ON GOING ACTIVITY					
Project management	7 %	153,94	3.825.000,00	23.906,25	3.680.128,13
Configuration Management	3 %	65,98	3.150.000,00	19.687,50	1.298.981,25
Documentation	3 %	65,98	1.800.000,00	11.250,00	742.275,00
Training & Support	3 %	65,98	3.150.000,00	19.687,50	1.298.981,25
Acceptance & Deployment	5 %	109,96	3.825.000,00	23.906,25	2.628.731,25
Sub Total	21,00%	461,84			9.649.096,88
QUALITY AND TESTING					
Quality Assurance & Control	12.34 %	271,38	3.825.000,00	23.906,25	6.487.678,13
Evaluation and Testing	24.66 %	542,32	2.000.000,00	12.500,00	6.779.000,00
Sub Total	37,00%	813,70			13.266.678,13
TOTAL BIAYA POKOK PERANGKAT LUNAK					38.756.640,64

**Gambar 6.7 Perhitungan rekap biaya**

Gambar 6.7 menunjukkan nilai usaha dan biaya pada masing-masing aktivitas yang ada pada pengembangan perangkat lunak dan juga rekapitulasi nilai usaha dan biaya total. Nilai biaya yang ada didasarkan pada masing-masing standard gaji pada yang ada pada masing-masing aktivitas.



- Hasil cetak penawaran



**Gambar 6.10 Dokumen penawaran**

Gambar 6.10 merupakan bentuk dari dokumen penawaran yang dibuat setelah hasil estimasi disetujui oleh direktur. Dokumen ini nantinya dicetak dan si stempel oleh sekretaris untuk kemudian dilakukan pengiriman kepada client dari aplikasi

## 6.7. Pembahasan

Aplikasi UCP ini dibuat sesuai dengan proses bisnis dari perhitungan UCP sendiri dan juga proses bisnis yang ada di perusahaan DTS. Penggunaan estimasi perhitungan dengan menggunakan metode UCP ini bertujuan untuk memberikan acuan atau landasan bagi tim pengembang DTS dalam menentukan estimasi harga perangkat lunak yang akan dikerjakan. Selain itu, dengan menggunakan metode ini, estimasi perangkat lunak akan lebih akurat dan mendekati sebenarnya. Hal

ini dapat ditunjukkan dari hasil data percobaan yang menunjukkan nilai usaha (*Effort*) yang dihasilkan dalam pengembangan dengan menggunakan metode UCP yaitu sebesar 2115.845 *Man-Hour*. sedangkan untuk actual *Effort* yang dihasilkan yaitu sebesar 2244 *Man-Hour*. perbedaan yang terjadi yaitu hanya 88.15 *Man-Hour* atau 96% mendekati kenyataan. Perbedaan juga terjadi antara nilai proyek hasil estimasi dengan nilai proyek aktual.

Hal ini dikarenakan adanya standard gaji yang mungkin terlalu tinggi. Jadi solusinya, untuk harga bisa dilakukan kustomisasi sesuai dengan standard atau kesepakatan yang ada pada perusahaan DTS. Berikut adalah perbandingan hasil perhitungan aplikasi dengan data sesungguhnya (*real*) ada pada tabel.

**Tabel 6.7 Perbandingan hasil estimasi dengan aktual**

Indikator	Aplikasi TDI		
	Hasil Estimasi	Hasil aktual	Deviasi
Usaha ( <i>Effort</i> )	2115.845	2244	5%
Nilai biaya(Rp)	38.756.640,64	30.000.000	2.1%

Aplikasi UCP juga terdapat fitur untuk memberikan notifikasi dalam bentuk email yang dikirimkan langsung kepada direktur setiap kali tim pengembang selesai melakukan estimasi harga. Hal ini bertujuan untuk meminimalisir *miss communication* dalam hal validasi hasil estimasi. Selain itu, dengan adanya fitur ini direktur bisa mengetahui aplikasi apa saja yang sudah diestimasi oleh tim pengembang dan status dari aplikasi yang akan diestimasi.

## **BAB VII PENUTUP**

Pada bab ini dibahas mengenai kesimpulan dari semua proses yang telah dilakukan dan saran yang dapat diberikan untuk pengembangan yang lebih baik.

### **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. Aplikasi UCP dibuat berdasarkan alur yang ada pada estimasi harga dengan UCP dan juga proses penentuan harga perangkat lunak yang ada di perusahaan DTS
2. Metode pengembangan yang digunakan untuk mengembangkan perangkat lunak UCP yaitu dengan metode incremental dengan tiga kali *increment*.
  - a. Tahap *increment* pertama menghasilkan aplikasi UCP yang bisa melakukan estimasi usaha dengan menggunakan metode UCP
  - b. Pada tahap *increment* kedua, aplikasi hasil dari *increment* pertama ditambahkan fitur baru sesuai dengan proses bisnis penentuan harga yang ada di perusahaan DTS diantaranya yaitu: fitur menghasilkan nilai biaya, dokumen penawaran, notifikasi e-mail, dan pengelolaan aplikasi UCP seperti pengelolaan pengguna, profesi, dan lain-lain.
  - c. Pada tahap *increment* ketiga, aplikasi dari hasil *increment* kedua ditambahkan beberapa fitur, baik itu fitur fungsional dan fitur non-fungsional. Fitur fungsional diantaranya ditambahkan fitur biaya operasional, fitur informasi *client* aplikasi, fitur nilai ER yang dinamis, dan lain-lain.

3. Pembangunan aplikasi menggunakan metode *increment* dapat menghindari ketidaksesuaian antara perancangan dengan implementasi. Hal ini dikarenakan pada setiap hasil implementasi di setiap *increment*, dilakukan pengujian *Correctness Testing* untuk menguji kesesuaian antara perancangan dan implementasi. Jika tidak sesuai, maka akan dilakukan perbaikan salah satunya dalam bentuk *Reverse Engineering* untuk bahan *increment* selanjutnya.
4. Tahapan Implementasi aplikasi UCP untuk melakukan estimasi harga perangkat lunak menghasilkan aplikasi UCP berbasis web yang sudah diuji dengan menggunakan metode pengujian *blackbox testing/correctness testing* untuk pengujian *use case*, *useability testing* untuk pengujian antarmuka, dan *portability testing* untuk pengujian felibilitas aplikasi dalam device atau browser, dan *security testing* untuk menguji keamanan web.
5. Terdapat beberapa perbaikan dari hasil pengujian, baik itu hasil pengujian untuk kebutuhan fungsional maupun untuk kebutuhan non fungsional.
6. Perbaikan pengujian dilakukan disetiap tahapan *increment* yang ada.
7. Kesesuaian kebutuhan fungsional (KF) sistem dengan implementasi aplikasi dapat dicapai 100% karena pada setiap hasil *increment* yang sudah dilalui, dilakukan perbaikan-perbaikan penyesuaian, baik dari segi kebutuhan maupun dari segi perancangan. Pada setiap tahapan juga memungkinkan untuk dilakukan *reverse engineering* untuk menyesuaikan hasil perancangan dan implementasi.
8. Keunggulan aplikasi UCP ini dibandingkan dengan aplikasi estimasi lainnya yaitu adanya fitur *Effort Rate (ER)* yang dinamis sesuai dengan history proyek. Hal ini membuat hasil estimasi menjadi lebih akurat.

## 7.2. Saran

Adapun saran yang diberikan untuk pengembangan kedepan yaitu sebagai berikut.

1. Aplikasi UCP bisa dikaitkan dengan time management, karena dalam aplikasi UCP ini nanti bisa degenerate waktu estimasi yang dibutuhkan dalam pengembangan perangkat lunak. Jumlah waktu estimasi nantinya bisa digunakan dalam menentukan timeline pengerjaan proyek.
2. Aplikasi UCP juga bisa digunakan untuk melihat available tim. Setiap tim yang masih terlibat dalam pengembangan aplikasi akan tercatat dalam aplikasi UCP sehingga bisa menghindari pemberian beban yang berlebihan pada tim pengembang.
3. Hasil pengujian keamanan menggunakan tools *accunetix* masih tergolong dalam kategori medium (level-2). Kategori ini sudah menunjukkan level cukup baik. Namun, untuk kedepan, level keamanan perlu dilakukan peningkatan lagi menjadi level low risk (level-1).
4. Notifikasi hasil esitmasi yang dikirmkan tidak hanya dalam bentuk email, tpi juga dalam bentuk sms gateway agar pengiriman hasil notifikasi bisa lebih cepat sampai ke penerima

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] Karmilasari, "Pengembangan dan Peranan Sistem Teknologi Informasi," 2014.
- [2] F. Hanif, "Strategi Pengembangan Perangkat Lunak SI," Yogyakarta, 2007.
- [3] C. Murali, "Software Estimation Best Practices, Tools and Techniques for Software Project Estimators," J.Ross Publishing, 2009, pp. 84-87.
- [4] Edward dan Carroll, "Estimating Software Based on Use Case Points.,"" 2005 *Object-Oriented, Programming, Systems, Languages, and Applications (OOPSLA) Conference, San Diego, CA*, 2005.
- [5] S. Sehlhorst, "Software Cost Estimation With Use Case Points – Free Excel Spreadsheet," 20 February 2007. [Online]. Available: <http://tynerblain.com/blog/?s=use+case+point+>.
- [6] "Use case point calculator," 10 December 2014. [Online]. Available: <http://tools.michalkoci.com/UseCasePointCalculator.htm>.
- [7] G. Karner, "Resource Estimation for Objectory Projects," Objective Systems SF AB,, 1993.
- [8] S. Kumar, "Understanding Testing Estimation Using Use Case Metrics," 2009.
- [9] M. H. M. L. Y. Yang, "Phase DIstribution of Software Development *Effort*," 2008.
- [10] K. Saleh, "*Effort* and Cost Allocation in Medium to Large Software Development Projects," *International Journal of Computers*, vol. 5, p. 1, 2011.
- [11] E. R. Carroll, "Estimating Software Based on Use Case Points," *Object-Oriented, Programming, Systems*,

*Languages, and Object Oriented Programming Systems Languages and Applications (OOPSLA) Conference*, pp. 257-265, 2005.

- [12] C. Mike, "Estimating With Use Case Points," 2005. [Online]. Available: <http://www.methodsandtools.com/archive/archive.php?id=25>.
- [13] M. Ochodek, B. Alchimowicz, J. Jurkiewicz dan J. Nawrocki, "Improving the reliability of transaction identification in use cases," *Information and Software Technology*, pp. 885-897, 2011.
- [14] G. Desy, "Perbaikan Formula Unadjusted Use Case Weight pada Use Case Point Untuk Estimasi *Effort* Proyek Pengembangan Perangkat Lunak," ITS-JSI, Surabaya, 2015.
- [15] A. P. Subriadi, Sholiq dan P. A. Ningeum, "Critical Review Of The *Effort* Rate Value In Use Case Point Method For Estimating Software Development *Effort*," *Journal of Theoretical and Applied Information Technology*, vol. 59, no. 3, p. 743, 2014.
- [16] S. KASSEM, "*Effort* and Cost Allocation in Medium to Large Software Development Projects," *INTERNATIONAL JOURNAL OF COMPUTERS*, vol. 5, no. 1, pp. 75-78, 2011.
- [17] Kelly Services, "2013 Salary Guide," Kelly Services Indonesia, Jakarta, 2013.
- [18] Linda, Estimasi Biaya Proyek Pengembangan Perangkat Lunak pemerintahan Berskala Small-medium Dengan Metode Use Case Point (UCP), Surabaya: Not Yet Published, 2015.
- [19] Perusahaan DTS, "Company Profile," Dynamic Team Solution, Surabaya, 2013.
- [20] Perusahaan DTS, "Struktur Organisasi Perusahaan DTS,"

Dynamic Team Solution, Surabaya, 2014.

- [21] Fathurahman, Interviewee, *Proses bisnis perusahaan Dynamic Team Solution*. [Wawancara]. 4 Febuary 2015.
- [22] “Software Development Methodologies,” November 2014. [Online]. Available: <http://www.itinfo.am/eng/software-development-methodologies/>.
- [23] cms.Gov, “Selecting A Developmen Approach,” 27 March 2008. [Online]. Available: <http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>.
- [24] TatvaSoft - Software outsourcing and Offshore Software development company, “Top 12 Software Development Methodologies and Its Advantages/Disadvantages,” 21 November 2014. [Online]. Available: <http://www.tatvasoft.com/blog/top-12-software-development-methodologies-and-its-advantages-disadvantages>.
- [25] F. P. Deek, J. A. M. McHugh dan O. M. Eljabiri, *Strategic Software Engineering: An Interdisciplinary Approach*, Boca Raton, FL: Auerbach Publications, 2005.
- [26] S. P. Roger, “The Incremental Model,” dalam *Software Engineering, A Practitioner's Approach*, New York, McGraw-Hill Series in Computer Science, p. 36.
- [27] S. Sangeeta, *Software Engineering*, New Delhi: New Age International, 2010.
- [28] S. Roger, *Software Engineering*, New York: McGraw-Hill, 2001.
- [29] Q. Terry, “Introduction to UML 2.0,” IBM Corporation, New York, 2005.
- [30] I. A. M. Kom, *Oriented Programming Php 5*, Jakarta: Elex Media Komputindo, p. 118.
- [31] D. Sri, “Pengantar Unified Modeling Language (UML),”

Ilmu Komputer, Jakarta, 2003.

- [32] C. Robert, *UML for Java Programmers*, New Jersey: Prentice-Hall, Inc., 2002.
- [33] M. Patrick, "Use Case Leveling Definitions," 4 April 2013. [Online]. Available: <https://wiki.nci.nih.gov/display/seminfra/Use+Case+Leveling+Definitions>.
- [34] TechTarget, "use case diagram," 2014. [Online]. Available: <http://whatis.techtarget.com/definition/use-case-diagram>.
- [35] R. James, J. Ivar dan B. Grady, "Activity Diagram," dalam *The Unified Modeling Language Reference Manual*, California, ADDISON-WESLEY, 1998, pp. 81-84.
- [36] D. Preeti, "Ezine Articles," 19 August 2011. [Online]. Available: <http://ezinearticles.com/?Activity-Diagrams---Advantages,-Disadvantages-and-Applications-of-Use&id=6506446>.
- [37] W. Scott, "UML 2 Sequence Diagrams: An Agile Introduction," 2014. [Online]. Available: <http://www.agilemodeling.com/artifacts/sequenceDiagram.htm>.
- [38] STIKOM, "Definisi Testing," dalam *Testing dan Implementasi Sistem*, Surabaya, STIKOM, 2003, p. 3.
- [39] K. Yoga, Sholiq dan A. Feby, "Rancang Bangun Perangkat Lunak untuk Workflow Pengelolaan Surat Menyurat Dinas Bagian Surat Masuk Di Kabupaten Buton Utara," *Jurnal Teknik ITS*, vol. 1, p. 272, 2012.
- [40] A. R. F. Giuseppe A. Di Lucca, "Testing Web-based applications: The state of the art and future trends," *Information and Software Technology* 48, p. 1172–1186, 2006.
- [41] B. a. P. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*:

- Fifth Edition, New York: Addison-Wesley Publ. Co, 2010.
- [42] M. Sharon, "PHP creator Rasmus Lerdorf," 4 February 2012. [Online]. Available: <http://www.computerworld.com/article/2586472/app-development/q-a--php-creator-rasmus-lerdorf.html>.
- [43] Y. K. Ardhana, *Pemrograman PHP Codeigniter Blackbox*, Purwokerto: Jasakom, 2013.
- [44] E. Quigley dan M. Gargenta, *PHP and MySQL by Example*, Massachusetts: Prentice Hall Professional, 2006.
- [45] "Change Log," 28 February 2006. [Online]. Available: <https://ellislab.com/codeigniter/user-guide/changelog.html>.
- [46] R. Dudi dan I. Dedy, "Implementasi Metode Model View Controller Menggunakan Framework Code Igniter dalam Pengembangan Aplikasi Manajemen Depo Petikemas pada Unit Usaha Belawan Logistics Center," *Seminar Nasional Teknologi Informasi dan Komunikasi*, pp. 2-2, 2012.
- [47] S. Arif dan K. Qadeem, "Requirement Engineering Process, Tools/Technologies, & Methodologies," *International Journal of Reviews in Computing*, p. 46, 2009-2010.
- [48] M. Ochodek, J. Nawrocki dan K. Kwarciaak, "Simplifying Effort estimation based on Use Case Points," *Elsevier*, pp. 200-2013, 2011.
- [49] Sholiq, T. Susanto, A. P. Widodo dan K. , "Effort Rate On Use Case Point Method For Effort Estimation Of Website Development," *Journal of Theoretical and Applied Information Technology*, vol. 63, p. 217, 2014.

*“Halaman ini sengaja dikosongkan”*

## **LAMPIRAN A-List Pertanyaan awal**

1. Bagaimana awal berdirinya perusahaan DTS ini?
2. Bagaimana struktur organisasi yang ada pada perusahaan DTS?
3. Bagaimana proses bisnis yang ada pada perusahaan DTS ?
4. Bagaimana proses dalam menentukan harga perangkat lunak yang ada pada perusahaan DTS?
5. Apakah ada kendala/permasalahan yang ada dalam proses estimasi harga perangkat lunak selama ini?. Jika iya, apa saja kendala yang ada.
6. Apakah sudah ada solusi jika terjadi permasalahan estimasi harga?
7. Sudah efektifkah solusi yang diberikan selama ini?
8. Apakah harapan dari bapak terkait dengan proses estimasi harga perangkat lunak ini?

A- 2 -

*Halaman ini sengaja dikosongkan*

## LAMPIRAN B-Rekaman Wawancara

<b>Pertanyaan dan Jawaban</b>
<b>Wawancara tahap 1</b>
Bagaimana awal berdirinya perusahaan DTS?
DTS (Dynamic Team Solution) atas dasar inisiatif beberapa mahasiswa aktif jurusan sistem informasi. DTS mulai beroperasi pada pertengahan septeber 2011. Nama DTS ini dipilih karena kami ingin membangun sebuah tim kerja yang dinamis. Tim dinamis adalah tim yang berkinerja tinggi, tim yang memanfaatkan energinya untuk menghasilkan sesuatu. Tim yang para anggotanya menyadari kekuatannya dan menggunakannya untuk mencapai visi bersama.
Bagaimana struktur organisasi yang ada pada perusahaan DTS?
Saat ini DTS masih beranggotakan 12 orang yang merupakan mahasiswa aktif Jurusan Sistem Informasi ITS. Untuk detail dari struktur organisasi, anda bisa lihat pada dokumen yang akan saya berikan.
Bagaimana proses bisnis yang ada pada perusahaan DTS ?
Perusahaan DTS ini merupakan perusahaan yang berbasis proyek, jadi mendapatkan client ini merupakan point kunci keberhasilan perusahaan kami. Client kami selama ini berasal dari instansi pemerintahan, perusahaan BUMN, UKM dan mahasiswa. Produk dan layanan kami baik itu pembangunan aplikasi yang berbasis web maupun website company profile dan toko online.
Bagaimana proses dalam menentukan harga perangkat lunak yang ada pada perusahaan DTS?

<b>Pertanyaan dan Jawaban</b>
<p>Selama ini, proses estimasi harga perangkat lunak yang ada pada perusahaan DTS yaitu dilakukan dengan cara manual yaitu dengan Pendapat ahli, dalam hal ini yaitu Project Manager atau Sistem Analis dari tim pengembang. Ketika ada calon klien, sistem analis dan PM datang menemui untuk melakukan penggalan kebutuhan awal. Setelah melakukan penggalan, PM, Sistem Analis dan anggota tim pengembang melakukan diskusi untuk menentukan harga dari aplikasi yang diinginkan oleh calon klien tersebut. Setelah disepakati oleh tim, estimasi harga tersebut dikomunikasikan kepada Direktur untuk dilakukan validasi persetujuan. Setelah disetujui, diberikan kepada sekretaris untuk dilakukan cetak dokumen penawaran harga untuk diberikan ke calon klien.</p>
<p>Apakah ada kendala/permasalahan yang ada dalam proses estimasi harga perangkat lunak selama ini?. Jika iya, apa saja kendala yang ada.</p>
<p>Selama proses estimasi harga perangkat lunak ini, terdapat kendala-kendala yang dihadapi. Pertama, dalam proses estimasi penentuan harga perangkat lunak, harga yang diestimasi tidak sesuai dengan harga kenyataan (Overbudget). Selain itu, dalam proses pengajuan penawaran ke client, client sering tidak setuju dengan alasan harga terlalu tinggi. Kedua, dalam melakukan konfirmasi hasil estimasi ke direktur. Kadang-kadang antara tim pengembang dan direktur miss dalam melaporkan estimasi harga. Direktur dalam melakukan validasi persetujuan, tidak ada dasar atau acuan. Sehingga terkadang keputusan dari direktur dalam menentukan harga tidak disetujui oleh tim pengembang</p>

<b>Pertanyaan dan Jawaban</b>
Apakah sudah ada solusi jika terjadi pemasalahan estimasi harga?
Selama ini masih belum ada solusi yang ada dalam estimasi harga perangkat lunak.
Sudah efektifkah solusi yang diberikan selama ini?
Masih belum efektif
Apakah harapan dari bapak terkait dengan proses estimasi harga perangkat lunak ini?
Harapan saya dengan perangkat lunak estimasi harga nanti yang dibuat dapat memudahkan koordinasi dan komunikasi antara tim pengembang, System analyst dan direktur dalam melaporkan data permintaan project dan persetujuan validasi harga sebuah project.
Aplikasi seperti apa yang bapak harapkan untuk mengatasi permasalahan estimasi harga perangkat lunak ini?
Fitur apakah yang sekiranya bapak inginkan dalam aplikasi ini nanti?
Yang jelas mengikuti solusi yang akan anda berikan dalam mengatasi permasalahan ini. Saya harap anda memiliki solusi yang efektif dalam melakukan estimasi harga ini. Yang penting, solusi yang ada berikan tidak merubah proses utama dalam melakukan estimasi harga di perusahaan kami. Hasil estimasi bisa dilakukan perubahan jika terdapat kesalahan
Maksudanya tidak merubah pak?

<b>Pertanyaan dan Jawaban</b>
<p>Ya proses utama, kan dalam estimasi harga harus dilakukan oleh orang yang kompeten, dalam hal ini yaitu PM dan Sistem Analist yang berasal dari internal perusahaan kami. Selanjutnya harga hasil estimasi harus disetujui dulu oleh direktur sebelum dibuatkan penawaran ke calon client.</p>
<p>Siapa saja yang bisa menggunakan aplikasi ini nanti pak?</p>
<p>Sesuai dengan proses bisnis yang ada, yang menggunakan aplikasi ini adalah orang-orang tertentu, yaitu Direktur, Project Manager, Sistem Analist, dan Sekretaris. Mungkin ada pengguna lain nantinya, tapi harus dimasukan dulu data nya.</p>
<p>Untuk tahap peratama ini, saya akan memberikan solusi estimasi harga dengan menggunakan Use Case Point pak. Metode yang saya ajukan ini didasarkan atas penelitian yang dilakukan oleh dosen pembimbing saya pak. Bagaimana menurut bapak?</p>
<p>Kalau saya ngikut dari faiz saja. Pokoknya yang penting aplikasi bisa digunakan di DTS dan bisa mengatasi permasalahan yang ada</p>
<p><b>Wawancara tahap kedua</b></p>
<p>Pak, dalam melakukan estimasi harga ini, saya membutuhkan gaji tetap tiap bulan dari masing-masing tim yang terlibat dalam pengembangan perangkat lunak. Bolehkah saya meminta data itu?</p>
<p>Maaf, untuk gaji tetap perusahaan DTS masih belum ada. Gaji yang diberikan oleh programmer didasarkan atas proyek yang ada. Kami mempunyai SOP dalam melakukan pembagian fee setiap ada project yang dikerjakan</p>

<b>Pertanyaan dan Jawaban</b>
Oh begitu, kami menyarankan bagaimana kalau untuk sementara dalam melakukan estimasi ini saya menggunakan standard gaji. Kami sudah berdiskusi dengan dosen pembimbing saya bahwa dalam membuat standard gaji lebih baik menggunakan standard gaji yang dikeluarkan oleh Kelly service.
Oh jadi tetep ada standard gaji ya...?
Iya pak,
Okey deh kalau begitu. Oh iya kamu buat dinamis saja untuk nilai gaji. Soalnya, nilai gaji kan fluktuatif, jadi saya nanti bisa melakukan perubahan sewaktu-waktu jika diperlukan.
Okey pak. Saya akan memberikan menu perubahan juga pada pelaku aktivitas. Soalnya hal tersebut akan berpengaruh terhadap hasil estimasi yang dilakukan.
Pak, untuk sistem notifikasi ke bapak mengenai hasil estimasi yang sudah dilakukan oleh tim, saya menawarkan lewat email. Bagaimana pak?. Soalnya saya rasa untuk menggunakan sms gateway ada biaya tambahan lagi pak.
Untuk sementara,saya rasa penggunaan notifikasi via email cukup. Saya juga sering online soalnya.
Apakah nanti dalam melakukan penawaran dijadikan satu dengan sistem ini pak?
Iya, jadikans satu saja. Oh iya untuk output dokumen penawaran, dalam bentuk doc saja biar nanti kalau ada perubahan bisa dilakukan dengan mudah.
Apakah biaya hasil estimasi bisa dilakukan perubahan?
Iya, Direktur saja yang melakukan itu. Biasanya harga yang diberikan oleh sistem tidak cocok/terlalu tinggi. Keputusan perubahan dari nilai biaya terletak pada direktur
<b>Wawancara tahap ketiga</b>

<b>Pertanyaan dan Jawaban</b>
<p>Pak, dalam metode UCP ini kan terdapat komponen yang bernama ER, ER ini merupakan konstanta yang digunakan untuk menentukan besar usaha yang dibutuhkan untuk membuat perangkat lunak. Dari refrensi yang saya dapatkan, nilai ER dapat ditentukan dari history proyek. Jadi nilai ER akan berubah-ubah tergantung dari proyek sebelumnya. Bagaimana menurut bapak?</p>
<p>Oh iya, Gpp yang penting nanti bisa dipertanggungjawabkan hasil yang ada.</p>
<p>Terus apa lagi yang bapak sarankan ke saya?</p>
<p>Itu saja sih. Pokoknya Kamu buat se simple mungkin agar tim saya nanti dalam menggunakan aplikasi ini tidak kesulitan</p>
<p>Pak dalam dokumen penawaran apa saja yang harus dilakukan otomatisasi</p>
<p>Pokoknya memudahkan user saja, terutama sekretaris nanti dalam melakukan pencetakan dokumen. Kecuali stempel dan ttd harus dibuat manual.</p>

## LAMPIRAN C-Interview Result

Kode Interview	Pertanyaan dan Jawaban
IR-01	<p>DTS (Dynamic Team Solution) didirikan atas dasar inisiatif beberapa mahasiswa aktif jurusan sistem informasi. DTS mulai beroperasi pada pertengahan septeber 2011. Nama DTS ini dipilih karena kami ingin membangun sebuah tim kerja yang dinamis. Tim dinamis adalah tim yang berkinerja tinggi, tim yang memanfaatkan energinya untuk menghasilkan sesuatu. Tim yang para anggotanya menyadari kekuanyannya dan menggunakannya untuk mencapai visi bersama.</p>
IR-02	<p>Saat ini DTS masih beranggotakan 12 orang yang merupakan mahasiswa aktif Jurusan Sistem Informasi ITS. Adapun struktur organisasi dari DTS sendiri ada pada <b>lampiran</b>.</p>
IR-03	<p>Perusahaan DTS ini merupakan perusahaan yang berbasis proyek, jadi mendapatkan client ini merupakan point kunci keberhasilan perusahaan kami. Client kami selama ini berasal dari instansi pemerintahan, perusahaan BUMN, UKM dan mahasiswa. Produk dan layanan kami baik itu pembangunan aplikasi yang berbasis web maupun website company profile dan toko online.</p>

<b>Kode Interview</b>	<b>Pertanyaan dan Jawaban</b>
IR-04	Selama ini, proses estimasi harga perangkat lunak yang ada pada perusahaan DTS yaitu dilakukan dengan cara manual yaitu dengan Pendapat ahli,dalam hal ini yaitu Project Manager atau Sistem Analis dari tim pengembang. Ketika ada calon klien, sistem analis dan PM datang menemui untuk melakukan penggalan kebutuhan awal. Setelah melakukan penggalan, PM, Sistem Analis dan anggota tim pengembang melakukan diskusi untuk menentukan harga dari aplikasi yang diinginkan oleh calon klien tersebut. Seteleh disepakati oleh tim, estimasi harga tersebut dikomunikasikan kepada Direktur untuk dilakukan validasi persetujuan. Setelah disetujui, diberikan kepada sekretaris untuk dilakukan cetak dokumen penawaran harga untuk diberikan ke calon klien.
IR-05	Selama proses estimasi harga perangkat lunak di DTS, terdapat kendala-kendala yang dihadapi. Pertama, dalam proses estimasi penentuan harga perangkat lunak, harga yang diestimasi tidak sesuai dengan harga kenyataan(Overbudget). Selain itu, dalam proses pengajuan penawaran ke client, client sering tidak setuju dengan alasan harga terlalu tinggi. Kedua, dalam melakukan konfirmasi hasil estimasi ke direktur. Kadang-kadang anantara tim pengembang dan direktur miss dalam melaporkan estimasi harga. Direktur dalam melakukan validasi persetujuan, tidak ada dasar atau acuan. Sehingga terkadang keputusan dari direktur dalam menentukan harga tidak disetujui oleh tim pengembang
IR-06	Selama ini masih belum ada solusi yang ada dalam estimasi harga perangkat lunak. Selama ini, masing-masing anggota tim mempunya kriteria tersendiri dalam menentukan estimasi harga perangkat lunak
IR-07	Solusi dalam melakukan estimasi harga masih belum ada. Oleh karena itu, DTS berharap terdapat solusi baru

Kode Interview	Pertanyaan dan Jawaban
	yang bisa menangani permasalahan estimasi harga dengan baik
IR-08	Harapan DTS dengan adanya perangkat lunak estimasi harga dapat memudahkan koordinasi dan komunikasi antara tim pengembang, System analyst dan direktur dalam melaporkan data permintaan project dan persetujuan validasi harga sebuah project.
IR-09	Aplikasi yang akan dibuat nanti harus mudah untuk dioperasikan dan membuat estimasi harga nantinya menjadi lebih akurat
IR-10	Apapun solusi yang ditawarkan nantinya bisa membuat estimasi harga menjadi lebih baik dan solusi yang diberikan tidak merubah proses utama dalam melakukan estimasi harga di perusahaan kami. Hasil estimasi bisa dilakukan perubahan jika terdapat kesalahan
IR-11	Aplikasi estimasi harga nantinya akan digunakan oleh orang-orang yang kompeten dalam bidangnya dalam hal ini yaitu PM dan Sistem Analist yang berasal dari internal perusahaan kami. Selanjutnya harga hasil estimasi harus disetujui dulu oleh direktur sebelum dibuatkan penawaran ke calon client.
IR-12	Pengguna dari aplikasi ini adalah orang-orang tertentu, yaitu Direktur, Project Manager, Sistem Analist, dan Sekretaris. Mungkin ada pengguna lain nantinya, tapi harus dimasukan dulu data nya.
IR-13	Pada tahap pembangunan awal, dibuatkan aplikasi yang mengacu pada metode estimasi harga dengan menggunakan <i>Use Case Point</i> (UCP)

<b>Kode Interview</b>	<b>Pertanyaan dan Jawaban</b>
IR-14	perusahaan DTS masih belum ada gaji tetap. Gaji yang diberikan oleh programmer didasarkan atas proyek yang ada. Kami mempunyai SOP dalam melakukan pembagian fee setiap ada project yang dikerjakan
IR-15	Nilai gaji yang digunakan dalam melakukan estimasi menggunakan standard gaji yang dikeluarkan oleh Kelly service. Terdapat fitur untuk melaukan perubahan terhadap gaji yang ada pada setiap profesi yang ada pada pelaku aktivitas. Soalnya hal tersebut akan berpengaruh terhadap hasil estimasi yang dilakukan.
IR-16	Adanya notifikasi via email untuk memudahkan direktur dalam mengetahui estimasi aplikasi yang dilakukan oleh tim pengembang dalam melakukan estimasi harga perangkat lunak
IR-17	Selian menghasilkan hasil estimasi, aplikasi nantinya juga menghasilkan output berupa dokumen penawaran sesuai dengan format dokumen penawaran yang ada di DTS
IR-18	Hasil estimasi bisa dilakukan perubahan oleh diretur maupun oleh tim pengembang
IR-18	Terdapat fitur nilai ER yang dinamis agar nilai hasil estimasi lebih akurat.
IR-19	Aplikasi dibuat tidak terlalu ribet agar tim bisa melakukan estimasi dengan mudah

## LAMPIRAN D-Acceptance checklist

- Fitur kebutuhan fungsional

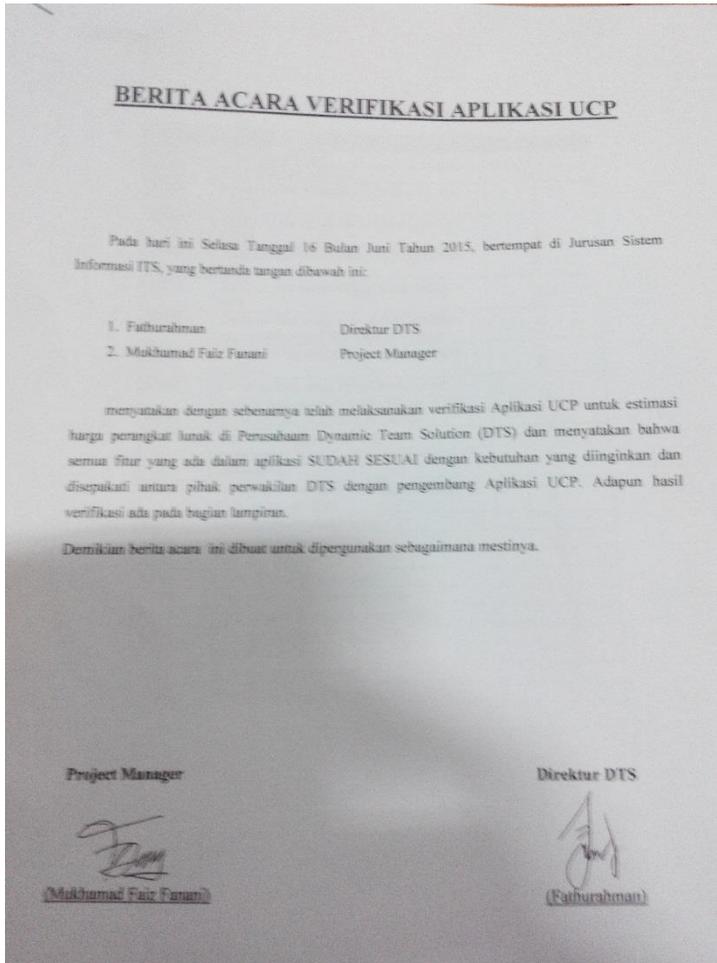
<b>Kode</b>	<b>Penjelasan</b>	<b>TTD Client</b>
KF01	Fitur Autentifikasi	
KF02	Fitur deskripsi aplikasi	
KF03	Aplikasi bisa melakukan perhitungan nilai UUCW	
KF04	Aplikasi bisa melakukan perhitungan nilai UAW	
KF05	Aplikasi bisa melakukan perhitungan nilai TCF	
KF06	Aplikasi bisa melakukan perhitungan nilai ECF	
KF07	Terdapat fitur untuk menampilkan rekap hasil perhitungan UCP dan nilai biaya	
KF08	Terdapat fitur pembedaan hak akses pada aplikasi	
KF09	Terdapat fitur perubahan terhadap nilai hasil perhitungan Estimasi UCP dan biaya	
KF10	Terdapat fitur notifikasi via email	
KF11	Terdapat tampilan halaman awal(homepage) yang terdiri dari informasi perusahaan	
KF12	Terdapat fitur pengelola pengguna yang login dalam aplikasi	
KF13	Fitur pengelola data profesi	
KF14	Fitur pengelola aktivitas	
KF15	Fitur log estimasi aplikasi	
KF16	Fitur validasi hasil estimasi	

Kode	Penjelasan	TTD Client
KF17	Fitur cetak dokumen penawaran	
KF18	Fitur pengelola anggota pengembang	
KF19	Fitur memasukan Actual <i>Effort</i>	
KF20	Fitur informasi client	
KF21	Fitur Biaya Operasional	
KF22	Fitur <i>Effort Rate</i> Dinamis	
KF23	Fitur distribusi aktivitas dinamis	

- Fitur kebutuhan non-fungsional

Kode	Penjelsan
KNF-01	Aplikasi telah berjalan sesuai dengan kebutuhan yang telah dispesifikasikan
KNF-02	Rancangan antar muka aplikasi yang <i>user friendly</i>
KNF-03	Aplikasi dapat dijalankan di browser yang berbeda
KNF-04	Aplikasi mempunyai mempunyai tingkat keamanan yang baik
KNF-05	Aplikasi UCP dapat diakses di lebih dari satu komputer atau laptop dan Device Smarphone
KNF-06	Adanya fasilitas pencarian untuk memudahkan pengguna mencari data

## LAMPIRAN E-Hasil Verifikasi Aplikasi



RAN

## ACCEPTANCE CHECKLIST

Kode	Penjelasan	TTD Client
KF01	Fitur Autentifikasi	✓
KF02	Fitur deskripsi aplikasi	✓
KF03	Aplikasi bisa melakukan perhitungan nilai UUCW	✓
KF04	Aplikasi bisa melakukan perhitungan nilai UAW	✓
KF05	Aplikasi bisa melakukan perhitungan nilai TCF	✓
KF06	Aplikasi bisa melakukan perhitungan nilai ECF	✓
KF07	Terdapat fitur untuk menampilkan rekap hasil perhitungan UCP dan nilai biaya	✓
KF08	Terdapat fitur pembedaan hak akses pada aplikasi	✓
KF09	Terdapat fitur perubahan terhadap nilai hasil perhitungan Estimasi UCP dan biaya	✓
KF10	Terdapat fitur notifikasi via email	✓
KF11	Terdapat tampilan halaman awal(homepage) yang terdiri dari informasi perusahaan	✓
KF12	Terdapat fitur pengelola pengguna yang login dalam aplikasi	✓
KF13	Fitur pengelola data profesi	✓
KF14	Fitur pengelola aktivitas	✓
KF15	Fitur log estimasi aplikasi	✓
KF16	Fitur validasi hasil estimasi	✓

## LAMPIRAN F-Data Use Case Percobaan

Berikut ini merupakan data use case percobaan pada aplikasi SIM Pelayanan Publik Disperdagin Pemkot Surabaya

UC.01   Menambah Data Pra-Permohonan TDI	
<b>Ringkasan</b> :	<i>Use case</i> ini berupa pengisian data-data sesuai syarat permohonan perijinan.
<b>Direct Actor</b> :	Masyarakat (pemohon)
<b>Prioritas</b> :	<i>Essential</i>
<b>Pre Conditions</b> :	Aktor berhasil Login dan berada pada <u>halaman Formulir Pra-Permohonan</u> (untuk setiap perijinan).
<b>Skenario Sukses Utama</b> :	<ol style="list-style-type: none"><li>1. Aktor membuka halaman formulir kemudian Sistem menampilkan halaman Formulir pra-Permohonan</li><li>2. Aktor mengisi dan menekan tombol Simpan setelah data-data sesuai dengan persyaratan yang sudah ditentukan.</li><li>3. Sistem menampilkan pesan “Apakah anda yakin data telah sesuai dan ingin menyimpannya?”</li><li>4. Aktor menekan tombol Ya dan sistem menjalankan UC.02.</li></ol>
<b>Skenario Alternatif</b> :	<ul style="list-style-type: none"><li>• Jika aktor tidak mengisi beberapa kolom yang seharusnya tidak boleh kosong, maka akan muncul tanda peringatan “Data kolom ini harus terisi”.</li><li>• Jika aktor menekan tombol Reset, maka sistem akan menampilkan formulir pra-permohonan dalam keadaan kosong.</li></ul>

UC.02   Mengunggah Dokumen Persyaratan TDI	
<b>Ringkasan :</b>	<i>Use case</i> ini berupa pengisian data-data pendukung sesuai syarat permohonan perijinan yang harus diunggah.
<b>Direct Actor:</b>	Masyarakat (pemohon)
<b>Prioritas :</b>	<i>Essential</i>
<b>Pre Conditions:</b>	Aktor berada pada <u>halaman Formulir Pra-Permohonan</u> (untuk setiap perijinan) dan selesai mengisi data pra-permohonan.
<b>Skenario Sukses Utama :</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih jenis berkas yang akan diunggah, kemudian menekan tombol Browse.Sistem kemudian menyimpan data berkas ke database</li> <li>2. Sistem akan menampilkan setiap berkas yang telah berhasil diunggah.</li> <li>3. Aktor menekan tombol Simpan kemudian sistem menampilkan proses proses berikutnya (UC.03)</li> </ol>
<b>Skenario Alternatif :</b>	<ul style="list-style-type: none"> <li>• Jika aktor mengunggah berkas dengan ukuran atau jenis yang tidak sesuai dengan ketentuan, maka sistem akan menampilkan “Ukuran atau jenis <i>file</i> tidak sesuai”.</li> </ul>

UC.03   Mengubah Data Permohonan TDI	
<b>Ringkasan :</b>	<i>Use case</i> ini berfungsi untuk mengubah data permohonan perijinan yang telah tersimpan sebelumnya.
<b>Direct Actor:</b>	Masyarakat (pemohon)

<b>Prioritas :</b>	<i>Rare</i>
<b>Pre Conditions:</b>	Aktor telah berhasil Login dan berada pada halaman utama.
<b>Skenario Sukses Utama :</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu Ubah Data Permohonan.</li> <li>2. Sistem akan menampilkan formulir permohonan perijinan yang telah terisi sesuai dengan data sebelumnya.</li> <li>3. Aktor dapat melakukan perubahan data kemudian menyimpannya dengan menekan tombol Simpan Perubahan. Sistem kemudian melakukan perubahan data</li> </ol>
<b>Skenario Alternatif :</b>	<ul style="list-style-type: none"> <li>• Jika aktor menekan tombol Batal, maka sistem tetap menyimpan data yang telah diisikan sebelumnya.</li> </ul>

<b>UC.04   Melihat Detil Data Permohonan TDI</b>	
<b>Ringkasan :</b>	<i>Use case</i> ini berfungsi untuk melihat rincian data yang telah diisikan dengan status terkirim.
<b>Direct Actor:</b>	Semua aktor
<b>Prioritas :</b>	<i>Essential</i>
<b>Pre Conditions:</b>	Aktor telah menekan tombol Kirim untuk mengirimkan data permohonan.
<b>Skenario Sukses Utama :</b>	<ol style="list-style-type: none"> <li>1. Setelah aktor menekan tombol Kirim, sistem akan memunculkan tab baru untuk memperlihatkan data yang telah diisikan beserta nomor masuk permohonan perijinan.</li> <li>2. Aktor mengklik tombol melanjutkan.</li> </ol>

	Kemudian sistem menampilkan form proses pencetakan bukti permohonan (UC.05).
<b>Skenario Alternatif :</b>	-

<b>UC.05   Mencetak Bukti Permohonan TDI</b>	
<b>Ringkasan :</b>	<i>Use case</i> ini berfungsi untuk mencetak bukti permohonan untuk setiap perijinan.
<b>Direct Actor:</b>	Masyarakat (pemohon)
<b>Prioritas :</b>	<i>Essential</i>
<b>Pre Conditions:</b>	Sistem telah menampilkan halaman Detil Data Permohonan.
<b>Skenario Sukses Utama :</b>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol Cetak Permohonan. Sistem akan menampilkan pengaturan pencetakan dokumen.</li> <li>2. Aktor mengklik tombol cetak. Sistem mencetak dokumen permohonan TDI</li> </ol>
<b>Skenario Alternatif :</b>	-

<b>UC.06   Men-generate Nomor Masuk Pendaftaran Pra-Permohonan</b>	
<b>Ringkasan :</b>	<i>Use case</i> ini secara otomatis dijalankan oleh sistem aplikasi dengan penambahan digit otomatis ( <i>auto-increment</i> ).
<b>Direct Actor:</b>	Masyarakat
<b>Prioritas :</b>	<i>Essential</i>

<b><i>Pre Conditions:</i></b>	Aktor telah menyimpan seluruh berkas pra-permohonan.
<b>Skenario Sukses Utama :</b>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol Kirim.</li> <li>2. Sistem akan menampilkan pesan “Apa Anda yakin data yang diunggah sudah benar? Menekan tombol Kirim berarti Anda tidak bisa mengubah data lagi”.</li> <li>3. Aktor menekan tombol Ya.</li> <li>4. Sistem menampilkan detail data pra-permohonan (UC.05) beserta nomor surat masuk pendaftarannya.</li> </ol>
<b>Skenario Alternatif :</b>	-

F- 6 -

*“Halaman ini sengaja dikosongkan”*

## LAMPIRAN G-Lampiran Penjelasan indikator TCF

INDIKATOR	DESKRIPSI
Distributed System Required	<p>Seberapa kompleks (terpusat ataupun terdistribusi) kebutuhan arsitektur pada proyek perangkat lunak ini?</p> <p>Semakin kompleks kebutuhan arsitektur, maka nilai semakin tinggi.</p>
Response Time Is Important	<p>Apakah menurut tim anda kecepatan respon (server) bagi pengguna merupakan faktor penting?</p> <p>Semakin pentingnya peningkatan waktu respon, maka nilai semakin tinggi.</p>
End User Efficiency	<p>Apakah menurut tim anda proyek perangkat lunak yang dikembangkan ini untuk mengoptimalkan efisiensi pengguna, atau hanya mengutamakan kemampuan tim saja?</p> <p>Semakin optimal efisiensi pengguna, maka nilai semakin tinggi.</p>
Complex Internal Processing Required	<p>Seberapa banyak algoritma yang sulit (kompleks) untuk dilakukan dan diuji pada proyek perangkat lunak ini?</p> <p>Semakin kompleks algoritma (resource leveling, OLAP cubes, dll) maka nilai semakin tinggi. Namun, database sederhana, maka nilai semakin rendah.</p>
Reusable Code Must Be A Focus	<p>Seberapa besar penggunaan ulang kode pada proyek perangkat lunak ini?</p> <p>“Penggunaan ulang kode mengurangi</p>

INDIKATOR	DESKRIPSI
	<p>jumlah usaha yang diperlukan untuk mendistribusikan sebuah proyek dan mengurangi jumlah waktu yang diperlukan untuk debugging sebuah proyek.”</p> <p>Semakin tinggi tingkat penggunaan ulang kode, maka nilai semakin rendah.</p>
Installation Easy	<p>Apakah menurut tim anda kemudahan instalasi proyek perangkat lunak ini bagi pengguna akhir merupakan faktor penting?</p> <p>Semakin tinggi tingkat kompetensi pengguna dalam instalasi proyek perangkat lunak ini, maka nilai semakin rendah.</p>
Usability	<p>Apakah kemudahan dalam penggunaan aplikasi merupakan kriteria utama dari proyek pembuatan perangkat lunak tim anda?</p> <p>Semakin besar pentingnya kegunaan, semakin tinggi nilai yang diberikan.</p>
Cross-Platform Support	<p>Apakah dibutuhkan dukungan multi-platform untuk aplikasi dari tim anda?</p> <p>Semakin banyak platform yang harus didukung (ini bisa versi browser, perangkat mobile, dll atau Windows / OSX / Unix), semakin tinggi nilai yang diberikan.</p>
Easy To Change	<p>Apakah aplikasi anda mudah untuk diubah atau disesuaikan oleh pengguna /</p>

INDIKATOR	DESKRIPSI
	customer di masa depan? Semakin mudah perubahan atau penyesuaian aplikasi anda, maka nilai semakin tinggi.
Highly Concurrent	Apakah dalam aplikasi anda dapat mengatasi penguncian database atau masalah konkurensi lainnya?  Semakin tinggi perhatian yang diberikan untuk menyelesaikan permasalahan dalam data atau aplikasi, maka nilai semakin tinggi.
Custom Security	Apakah dalam aplikasi anda solusi keamanan yang ada mudah digunakan, atau kode kustom harus dikembangkan?  Apabila kode kustom kemanan lebih harus dilakukan, maka nilai semakin tinggi
Dependence On Third-Party Code	Apakah aplikasi anda masih memerlukan kontrol dari pihak ketiga, seperti penggunaan ulang kode?  Apabila kebutuhan kontrol dari pihak ketiga tidak terlalu penting, maka nilai semakin tinggi.
User Training	Berapa lama waktu yang dibutuhkan untuk pelatihan pengguna diperlukan?  Semakin lama waktu yang dibutuhkan pengguna untuk penguasaan aplikasi, maka semakin tinggi nilai yang diberikan.

G- 4 -

*“Halaman ini sengaja dikosongan”*

## LAMPIRAN H-Lampiran Penjelasan Indikator ECF

<b>INDIKATOR</b>	<b>DESKRIPSI</b>
Familiarity with the Project	Apakah tim anda merasa familiar (menguasai) dengan proyek pengembangan perangkat lunak  Semakin familiar (menguasai), maka nilai semakin tinggi.
Application Experience	Sejauh mana pengalaman tim anda dalam membuat perubahan pada proyek pengembangan perangkat lunak yang sama? Semakin banyak pengalaman dalam membuat perubahan pada proyek yang sama maka nilainya semakin tinggi
OO Programming Experience	Sejauh mana pengalaman tim anda dalam membuat proyek perangkat lunak berbasis Object Oriented (OO) programming?. Semakin banyak pengalaman dalam Object Oriented (OO) programming, maka nilai semakin tinggi.
Lead Analyst Capability	Seberapa besar kapabilitas (kemampuan) dan pengetahuan menganalisa dalam tim anda? “Analisa kebutuhan yang buruk adalah pembunuh nomor satu dalam proyek - Standish Group melaporkan bahwa 40% sampai 60% dari cacat produk berasal dari kebutuhan yang buruk. Semakin besar kapabilitas dan pengetahuan, maka nilai semakin tinggi.
Motivation	Seberapa besar motivasi pada tim anda dalam membuat proyek pengembangan perangkat lunak ini?

	<p>Semakin besar motivasi, maka nilai semakin tinggi.</p>
<p>Stable Requirements</p>	<p>Seberapa besar kebutuhan pada proyek pengembangan perangkat lunak ini mengalami perubahan?</p> <p>Semakin besar kebutuhan akan perubahan, maka nilai semakin tinggi.</p>
<p>Part Time Staff</p>	<p>Apakah dalam tim anda terdapat anggota tim yang bekerja paruh waktu?</p> <p>Semakin banyak waktu yang digunakan anggota tim untuk bekerja paruh waktu, maka nilai semakin tinggi.</p>
<p>Difficult Programming Language</p>	<p>Seberapa sulit bagi tim anda, bahasa pemrograman yang digunakan dalam pembuatan proyek perangkat lunak ini?</p> <p>Semakin sulit bahasa pemrograman, maka nilai semakin tinggi.</p>

## LAMPIRAN I-Lampiran Penjelasan Aktivitas

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

Fase	Variabel Tahapan	Deskripsi aktivitas	Jenis Aktivitas
	Specifications	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> <hr/> <ul style="list-style-type: none"> <li>• Penyusunan dokumen SKPL</li> <li>• Rapat hasil analisis dengan stakeholder</li> <li>• Perbaikan dokumen SKPL</li> <li>• Dan lain-lain yang terkait</li> </ul>

Fase	Variabel Tahapan	Deskripsi aktivitas	Jenis Aktivitas
	Design	Rincian aktivitas ini antara lain merumuskan rancangan arsitektur teknologi, arsitektur tingkat tinggi, basis data, antarmuka pengguna, dan perancangan detail lainnya	<ul style="list-style-type: none"> <li>• Sosialisasi hasil analisis kebutuhan ke internal tim pengembang</li> <li>• Pembagian SDM dan jobdesk</li> <li>• Benchmark template aplikasi</li> <li>• Pembuatan mockup/user interface</li> <li>• Dan lain-lain yang terkait</li> </ul>

Fase	Variabel Tahapan	Deskripsi aktivitas	Jenis Aktivitas
	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 template dan plugin</li> <li>• Eksekusikode program</li> <li>• Revisi Program</li> <li>• Rapat dengan stakeholder internal atu external</li> <li>• <i>Release</i> apliaksi/sistem versi beta</li> <li>• Pembuatan user guide versi beta</li> <li>• Dan lain-lainya yang terkait</li> </ul>

Fase	Variabel Tahapan	Deskripsi aktivitas	Jenis Aktivitas
	Testing	<p>Aktivitas ini dilakukan merupakan aktiivtas pengujian (<i>Testing</i>) yang bertujuan mengetahui apakah aplikasi yang dibangun sudah terintegrasi dengan gambaran modul, sistem, hingga komponen penyusunnya dang mengetahui kesalahan atau ketidaksesuaian perangkat lunak dengan kebuhan yang sudah dispesifikasikan. Keluaran dari aktivitas ini berupa perangkat lunak yang sudah teruji dan terintegrasi(jika ada banyak model)</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 blackbox</li> <li>• Penyesuaiaan SKPL</li> <li>• Perbaikan dan melengkapi user guide</li> <li>• Dan lain-lain yang terkait</li> </ul>

Fase	Variabel Tahapan	Deskripsi aktivitas	Jenis Aktivitas
On Going Activity	Project Management	Aktivitas ini dimulai dari <i>kick-off</i> hingga penutupan proyek. Proses yang dilakukan dimulai dari pengelolaan sumber daya 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> <li>• Rapat penutupan proyek</li> <li>• Penentuan target proyek</li> <li>• Dan lain-lain yang terkait</li> </ul>

Fase	Variabel Tahapan	Deskripsi aktivitas	Jenis Aktivitas
	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> <li>• Dan lain-lain yang terkait</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. Aktivitas ini dilakukan selama proyek berlangsung	<ul style="list-style-type: none"> <li>• Pembuatan user guide aplikasi</li> <li>• Pembuatan video tutorial</li> <li>• Penataan laporan akhir proyek</li> <li>• Dan lain-lain yang terkait</li> </ul>
	Training &	Aktivitas ini dilakukan apabila	

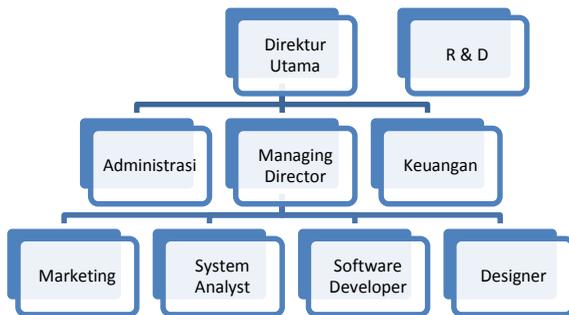
Fase	Variabel Tahapan	Deskripsi aktivitas	Jenis Aktivitas
	Support	pemrograman membutuhkan bantuan tambahan untuk mengatasi permasalahan tertentu maupun pelatihan terhadap pihak-pihak yang menggunakan aplikasi ( <i>user</i> ).	<ul style="list-style-type: none"> <li>• Pemberian materi pelatihan</li> <li>• Pembelian konsumsi pelatihan</li> <li>• Dan lain-lain yang terkait</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 dapat dilakukan oleh tim proyek	<ul style="list-style-type: none"> <li>• Pemeriksaan aplikasi oleh pengguna</li> <li>• Penandatanganan software quality</li> <li>• Maintenance aplikasi secara berkala sesuai kesepakatan antara PM dengan Client</li> <li>• Dan lain-laing yang terkait</li> </ul>

Fase	Variabel Tahapan	Deskripsi aktivitas	Jenis Aktivitas
	Evaluasi dan Pengujian	<p>Aktivitas ini termasuk dalam kegiatan verifikasi dan validasi. Di 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)</p>	<ul style="list-style-type: none"> <li>• Maintenance data</li> <li>• Maintenance hardware internal</li> <li>• Dain lain-lain yang terkait</li> </ul>

*Halaman ini sengaja dikosongkan*

## LAMPIRAN J-Lampiran Struktur Organisasi DTS

Perusahaan DTS merupakan perusahaan yang didirikan atas inisiatif mahasiswa aktif jurusan Sistem Informasi ITS. Perusahaan DTS mulai beroperasi pada pertengahan bulan september 2011. Perusahaan DTS ini bergerak di bidang jasa pembuatan website profil pribadi maupun organisasi. Selain website, juga melayani pembuatan aplikasi berbasis web yang bisa menunjang dan mempercepat proses bisnis yang ada di perusahaan/organisasi [19]. Adapun struktur organisasi yang ada pada Perusahaan DTS ini adalah sebagai berikut [20].



Berikut ini adalah *Job Description* dari masing-masing peran yang ada di Perusahaan DTS ini yaitu:

- **Direktur**

Bertugas sebagai pemimpin tunggal dari perusahaan yang memiliki hak penuh dalam pengambilan keputusan dan bertanggungjawab atas seluruh proses bisnis yang ada di perusahaan.

- **Managing Director (Direktur Program)**

5. Bertugas sebagai tangan kanan dari Direktur dalam membantu mengkoordinasikan setiap Divisi yaitu Divisi Administrasi, Keuangan, Divisi Marketing, Divisi Software Developer dan Divisi Designer
6. Bertugas mengkoordinasikan project manager dan tim yang sedang menjalankan atau akan menjalankan sebuah project.
7. Mencari peluang kerjasama dengan membangun relasi antar individu atau perusahaan guna membangun jaringan bisnis yang lebih luas
8. Mencari dan merekomendasikan calon sdm yang tepat untuk di rekrut menjadi anggota DTS kepada direktur

- **Administrasi dan Keuangan**

5. Bertugas membuat dan menyiapkan laporan-laporan dan dokumen-dokumen yang diperlukan oleh Direktur.
6. Bertugas melakukan pencatatan agenda rapat dan hasil rapat saat rapat DTS bulanan berlangsung
7. Bertugas melakukan pencatatan keluar masuk keuangan perusahaan. Membuat perencanaan harga produk dengan berkonsultasi dengan programmer.
8. Bertugas untuk membayar pajak setiap bulan di bank dan menyiapkan dokumen pelaporan pajak setiap bulannya.

- **Marketing**

7. Bertugas menentukan strategi pemasaran, dan produk-produk yang akan dijual atau ditawarkan kepada customer guna meningkatkan omzet dari perusahaan.
8. Mencari dan Membuat resume profil tentang perusahaan yang akan ditarget untuk proyek DTS.
9. Melakukan kontak dan penawaran kepada klien sesuai dengan SOP marketing DTS.
10. Melakukan tindaklanjut terkait klien melalui User Requirement.

11. Melakukan promo secara online dengan skala waktu tertentu melalui media social network DTS minimal 2 kali update status dalam sehari.
  12. Membuat laporan perkembangan marketing yang didaftarkan setiap minggu kepada koordinator (Managing Director) DTS.
- **System Analyst**
    3. Bertugas untuk mendampingi Managing Director, dan marketing untuk menganalisa kebutuhan yang diinginkan oleh customer.
    4. Bertugas untuk membantu Managing Director dalam menyiapkan data SKPL awal untuk kebutuhan dokumen penawaran project.
  - **Software Developer**
    3. Merekomendasikan anggota tim developer yang siap untuk mengikuti project DTS kepada managing director.
    4. Membantu meningkatkan kemampuan sdm developer dengan memberikan studi kasus untuk diselesaikan.
  - **Tim Developer**
    3. Bertugas menangani dalam pembuatan aplikasi dan melayani dalam memberikan solusi IT yang tepat kepada customer.
    4. Memiliki tanggungjawab untuk terus belajar dengan software Developer dan aktif serta bertanya kepada divisi R&D untuk diadakan tutorial/pelatihan.
  - **Designer**
    3. Bertugas membuat desain aplikasi/web yang bekoordinasi dengan divisi software developer untuk membuat tampilan desain yang dibutuhkan.
    4. Bertugas membantu desain yang diperlukan untuk menunjang tim marketing DTS, seperti; desain kartu nama, brosur, poster dsb.

- **Riset & Development (R&D)**

5. Bersama sama direktur dan wakil direktur untuk mentransfer knowledge serta pengalaman yang dimiliki kepada seluruh anggota DTS guna kesinambungan jalannya perusahaan.
6. Mengagendakan program training/pelatihan minimal sebulan sekali untuk tim DTS yang memerlukannya untuk menunjang kinerja DTS Team.
7. Melakukan riset dan monitoring riset DTS yang sedang dan akan berjalan.
8. Mengusulkan kepada direktur/wakil untuk produk/jasa baru yang bisa diambil sesuai kebutuhan pasar terkini.

## RIWAYAT PENULIS



Penulis bernama lengkap Mukhamad Faiz Fanani. Biasanya dipanggil Faiz. Penulis dilahirkan pada tanggal 23 Desember 1992 dan merupakan anak pertama dari 3 bersaudara. Pendidikan yang pernah ditempuh penulis yaitu: pendidikan formal di SD Negeri Banjarasri, SMP Negeri 1 Candi Sidoarjo, dan SMA Negeri 1 Sidoarjo. Penulis diterima menjadi mahasiswa di Jurusan Sistem Informasi – Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2011 dengan NRP 5211100043. Selama perkuliahan, penulis aktif di bidang akademik dan non-akademik. Dalam bidang akademik, penulis pernah menjadi asisten dosen pada mata kuliah Algoritma Pemrograman 2, asisten laboratorium di Laboratorium Pemrograman Sistem Informasi (LPSI), dan asisten laboratorium di Laboratorium Perencanaan dan Pengembangan Sistem Informasi (PPSI). Pada bidang non-akademik, penulis pernah mengikuti organisasi mahasiswa, Staff RISTEK HMSI 2012/2013, Manger RISTEK FTif 2013/2014, serta beberapa acara kepanitiaan baik tingkat jurusan, tingkat institut. Penulis juga pernah menjadi junior project manager pada perusahaan startup Dynamic Team Solution (DTS) pada tahun 2013-2014.

Penulis mengambil konsentrasi bidang minat Perencanaan dan Pengembangan Sistem Informasi dengan topik tugas akhir di Rancang Bangun Sistem Informasi dan Investasi TI. Untuk keperluan penelitian penulis dapat dihubungi melalui email: [fananifaiz@yahoo.com](mailto:fananifaiz@yahoo.com)