



**SISTEM ESTIMASI *EFFORT* PROYEK PENGEMBANGAN
PERANGKAT LUNAK MENGGUNAKAN *FUZZY USE CASE POINT*
(STUDI KASUS PERPUSTAKAAN DAERAH KABUPATEN JEMBER)**

SKRIPSI

Oleh

M. ANDHIKA PUJI HERMAWAN

NIM 132410101087

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER**

2018



**SISTEM ESTIMASI *EFFORT* PROYEK PENGEMBANGAN
PERANGKAT LUNAK MENGGUNAKAN *FUZZY USE CASE POINT*
(STUDI KASUS PERPUSTAKAAN DAERAH KABUPATEN JEMBER)**

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan pendidikan di Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember dan mendapat gelar Sarjana Sistem Informasi

Oleh

M. ANDHIKA PUJI HERMAWAN

NIM 132410101087

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER**

2018

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi.
2. Ayahanda tersayang H.Sugiarto dan Ibunda tercinta Hj. Hairani.
3. Istriku Tersayang Diah Tri Lestari.
4. Segenap anggota keluarga H. Sugiarto atas dukungan beserta doanya.
5. Sahabatku Yuca Akbar Maulana dan Savira Oktari atas dukungan, waktu dan doanya.
6. Temen – Temen Admin Senyum Media Group atas dukungan, waktu dan doanya.
7. Guru – guruku dari taman kanak-kanak sampai dengan perguruan tinggi.
8. Almamater Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

MOTO

“Jadilah yang terbaik atau tidak sama sekali”



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : M. Andhika Puji Hermawan

NIM : 132410101087

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Estimasi *Effort* Proyek Pengembangan Perangkat Lunak Menggunakan *Fuzzy Use Case Point* (Studi Kasus Perpustakaan Daerah Kabupaten Jember)” adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 16 November 2018

Yang menyatakan,

M. Andhika Puji Hermawan

NIM 132410101087

SKRIPSI

**ESTIMASI *EFFORT* PROYEK PENGEMBANGAN PERANGKAT
LUNAK MENGGUNAKAN *FUZZY USE CASE POINT*
(STUDI KASUS PERPUSTAKAAN DAERAH KABUPATEN JEMBER)**

Oleh :

M. Andhika Puji Hermawan

132410101087

Pembimbing :

Dosen Pembimbing Utama : **Fahrobby Adnan, S.Kom., M.msi**

Dosen Pembimbing Pendamping : **Tio Dharmawan, S.Kom., M.Kom**

PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Estimasi *Effort* Proyek Pengembangan Perangkat Lunak Menggunakan *Fuzzy Use Case Point* (Studi Kasus Perpustakaan Daerah Kabupaten Jember)”, telah diuji dan disahkan pada:

hari, tanggal : Jumat, 16 November 2018

tempat : Program Studi Sistem Informasi Fakultas Ilmu Komputer
Universitas Jember.

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Fahrobby Adnan, S.Kom.,MMSI
NIP. 198706192014041001

Tio Darmawan, S.Kom.,M.Kom
NIP. 760016851

PENGESAHAN PENGUJI

Skripsi berjudul “Sistem Estimasi *Effort* Pengembangan Perangkat Lunak Menggunakan *Fuzzy Use Case Point* (Studi Kasus Perpustakaan Daerah Kabupaten Jember)”, telah diuji dan disahkan pada:

hari,tanggal : Jumat, 16 November 2018

tempat : Program Studi Sistem Informasi Fakultas Ilmu Komputer
Universitas Jember

Tim Penguji :

Penguji I,

Penguji II,

Priza Pandunata, S.Kom., M.Sc
NIP. 198301312015041001

Beny Prasetyo, S.Kom., M.Kom
NIP 760016852

Mengesahkan
Dekan Fakultas Ilmu Komputer

Prof. Drs. Slamir, M.Comp.Sc.,Ph.D
NIP. 19670420 1992011001

RINGKASAN

Sistem Estimasi *Effort* Proyek Pengembangan Perangkat Lunak Menggunakan *Fuzzy Use Case Point* (Studi Kasus Perpustakaan Daerah Kabupaten Jember); M. Andhika Puji Hermawan, 132410101087 2018, 97 halaman ; Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

Teknologi Informasi yang awalnya hanya faktor pendukung bagi kegiatan bisnis, pada saat ini menjadi hal yang sangat krusial untuk mendukung proses kerja bisnis. Permintaan akan proyek perangkat lunak di bidang Bisnis semakin banyak dan semakin dibutuhkan untuk proses bisnis perusahaan. Untuk mengurangi kegagalan proyek perangkat lunak di bidang Bisnis, *planning* harus diperbaiki dan estimasi yang dilakukan harus benar. Salah satu cara untuk memperbaiki *planning* proyek yaitu dengan melakukan estimasi yakni melakukan prediksi mengenai keluaran dari sebuah proyek dengan meninjau jadwal, biaya, bahkan hingga ke resiko yang akan ditanggung serta *Effort* dalam proyek tersebut. Salah satu metode dalam melakukan estimasi adalah dengan menggunakan *use case point* yang dikembangkan dalam sebuah sistem estimasi effort proyek pengembangan perangkat lunak dengan studi kasus Perpustakaan Daerah Kabupaten Jember. Sistem ini bertujuan untuk membantu menghitung nilai atau biaya estimasi dari sebuah pengembangan perangkat lunak. Sistem ini diharapkan dapat membantu merumuskan beberapa permasalahan yang berkaitan dengan pengembangan sistem yang selama ini belum dilakukan estimasi dengan baik.

PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Sistem Estimasi *Effort* Proyek Pengembangan Perangkat Lunak Menggunakan *Fuzzy Use Case Point* (Studi Kasus Perpustakaan Daerah Kabupaten Jember)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Fahrobby Adnan, S.Kom.,M.MSI selaku Dosen Pembimbing Utama dan Tio Dharmawan, S.Kom., M.Kom. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Muhamad Arief Hidayat, S.Kom., M.Kom.sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Ayahanda tercinta Bapak Ahmad Sugiarto, Ibuku Hairani & Saudara - Saudaraku yang selalu mendukung dan mendoakan;
6. Istriku Tercinta Diah Tri Lestari yang selalu menyemangati;
7. Teman – Teman Admin Senyum Media Group atas dukungan, waktu dan doanya.
8. Sahabatku Yuca Akbar Maulana dan Savira Oktari atas dukungan, waktu dan doanya.
9. Teman-teman angkatan 2013;
10. Teman-Teman Program Studi Sistem Informasi di semua angkatan;
11. Semua pihak yang tidak dapat disebutkan satu persatu.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 16 November 2018

Penulis



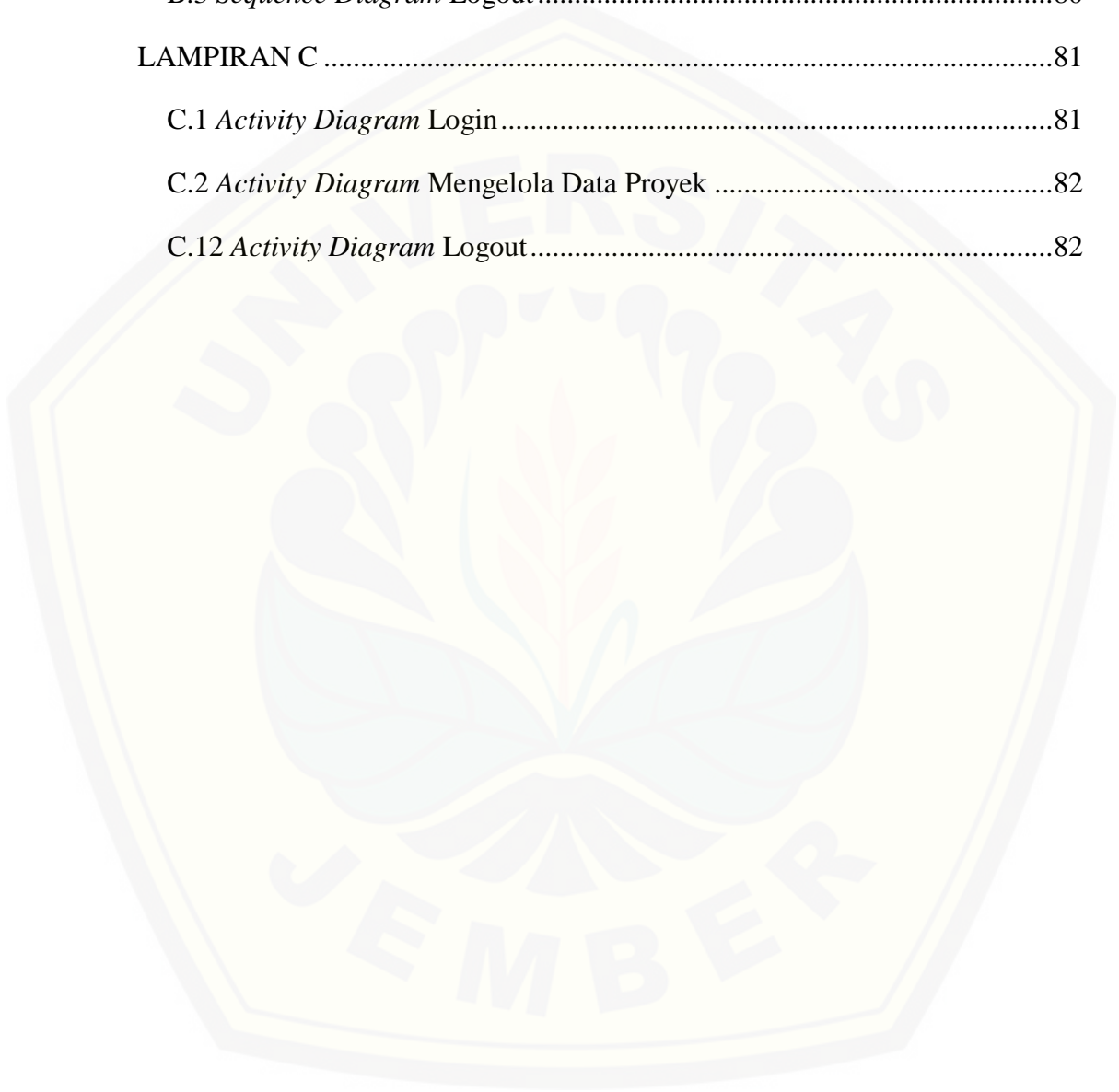
DAFTAR ISI

SKRIPSI.....	i
PERSEMBAHAN	ii
MOTO.....	iii
PERNYATAAN.....	iv
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN	viii
PRAKATA.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah.....	4
1.5 Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA	6
2.1 Proyek Pengembangan Perangkat Lunak.....	6
2.2 <i>Use Case Diagram</i>	6
2.3 <i>Use Case Point</i>	7
2.4 Logika Fuzzy.....	10
2.5 Metode Fuzzy <i>Use Case Point</i>	10
BAB 3. METODOLOGI PENELITIAN.....	13

3.1 Waktu Penelitian	13
3.2 Tahapan <i>Fuzzy Use Case Point</i>	13
3.3.1 Analisis Kebutuhan	18
3.3.2 Desain.....	18
3.3.3 Implementasi Sistem	19
3.3.4 Pengujian Sistem.....	19
3.3.5 Pemeliharaan.....	20
BAB 4. PENGEMBANGAN SISTEM.....	21
4.1 Analisis Kebutuhan Sistem	21
4.1.1 <i>Statement of Purpose (SOP)</i>	21
4.1.2 Kebutuhan Fungsional	21
4.1.3 Kebutuhan Non-Fungsional	22
4.1.4 Fungsi Sistem	22
Desain Sistem.....	23
4.2.1 <i>Business Process</i>	24
4.2.2 <i>Use Case Diagram</i>	25
4.2.3 Skenario Sistem.....	27
4.2.4 <i>Sequence Diagram</i>	32
4.2.5 <i>Activity Diagram</i>	36
4.2.6 <i>Class Diagram</i>	41
6.1.1 <i>Entity Relationship Diagram</i>	42
4.3 Penulisan Kode Program.....	42
4.4 Pengujian Sistem.....	43
4.4.1 Pengujian <i>White Box</i>	43
4.4.2 Pengujian <i>Black Box</i>	51

BAB 5. HASIL DAN PEMBAHASAN.....	54
5.1 Hasil Pembangunan Sistem.....	54
5.2 Hasil Pengumpulan Data.....	54
5.3 Hasil Perhitungan Nilai Actual Effort.....	54
5.3.1 Tampilan Halaman Data Proyek	55
5.3.2 Tampilan Halaman Data Unadjusted Actor Weight	56
5.3.3 Tampilan Halaman Data <i>Fuzzy un-ajusted usecase weight</i>	57
5.3.4 Tampilan Halaman Data Technical Complexity.....	58
5.1.5 Tampilan Halaman Environmental Complexity	59
5.4 Hasil Perhitungan Estimasi Use Case Point.....	60
5.4.1 Perhitungan Unadjusted Usecase Point (UUCP)	61
5.4.2 Perhitungan Technical Complexity Factor (TCF).....	63
5.4.3 Perhitungan Environmental Complexity Factor (ECF)	64
5.4.4 Perhitungan Estimasi Fuzzy Use Case Point	65
5.4.7 Perhitungan Nilai Effort.....	65
5.5 Pembahasan.....	66
BAB 6. PENUTUP.....	67
6.1 Kesimpulan	67
6.2 Saran.....	68
DAFTAR PUSTAKA	70
LAMPIRAN.....	76
LAMPIRAN A	76
A.1 Skenario Login	76
A.2 Skenario Mengelola Data Proyek.....	77
A.3 Skenario Keluar.....	78

LAMPIRAN B	80
B.1 <i>Sequence Diagram</i> Login	80
B.2 <i>Sequence Diagram</i> Mengelola Data Proyek.....	80
B.3 <i>Sequence Diagram</i> Logout	80
LAMPIRAN C	81
C.1 <i>Activity Diagram</i> Login.....	81
C.2 <i>Activity Diagram</i> Mengelola Data Proyek	82
C.12 <i>Activity Diagram</i> Logout.....	82



DAFTAR GAMBAR

Gambar 4.10 Kode Program Proses Environmental Complexity	49
Gambar 4.21 Diagram Alir Proses Environmental Complexity	49
Gambar 5.1 Halaman Login	55
Gambar 5.2 Halaman Data Proyek Pada Menu Dashboard	56
Gambar 5.3 Form Untuk Penilaian Estimasi Unadjusted Actor Weight	56
Gambar 5.4 Halaman Lihat Hasil Nilai Effort Unadjusted Actor Weight	57
Gambar 5.5 Form Untuk Penilaian Estimasi Fuzzy Unadjusted Usecase Weight..	57
Gambar 5.6 Halaman Lihat Hasil Nilai Effort Unadjusted Actor Weight	58
Gambar 5.7 Form Untuk Penilaian Estimasi Technical Complexity	59
Gambar 5.8 Halaman Lihat Hasil Nilai Effort Technical Complexity	59
Gambar 5.9 Form Untuk Penilaian Estimasi Environmental Complexity	60
Gambar 5.10 Halaman Lihat Hasil Nilai Effort Environmental Complexity	60
Gambar 5.11 Usecase website Perpustakaan Daerah Kabupaten Jember	60

DAFTAR TABEL

Tabel 2.1 Tipe dan pembobotan Use Case (Karner, 1993)	7
Tabel 4.1 Definisi Aktor	26
Tabel 4.2 Definisi Use Case.....	27
Tabel 4.3 Skenario Mengelola Anjusted Actor Weight	28
Tabel 4.4 Skenario Mengelola Fuzzy unajusted Use Case weight	29
Tabel 4.5 Mengolah Technical Complexity Factor.....	30
Tabel 4.6 Mengolah Environmental Complexity Factor.....	31
Tabel 4.7 Pengujian Testcase Proses Fuzzy Unadjusted Use Case Point	44
Tabel 4.8 Pengujian Testcase Proses Fuzzy Use Case Point	45
Tabel 4.9 Pengujian Testcase method count_phm.....	47
Tabel 4.10 Pengujian Testcase method count_effort	50
Tabel 4.11 Pengujian Blackbox	51
Tabel 5.1 Data Unadjusted Actor Weight Total (UAW)	61
Tabel 5.2 perhitungan nilai FUUCW	62
Tabel 5.3 Nilai FUUCP proyek pengembangan website perpustakaan	63
Tabel 5.4 Data Nilai Technical Factor	63
Tabel 5.5 Data Nilai Environmental Factor	64
Tabel 5.6 Nilai FUCP.....	65

BAB 1. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan.

1.1 Latar Belakang

Teknologi Informasi yang awalnya hanya faktor pendukung bagi kegiatan bisnis, pada saat ini menjadi hal yang sangat krusial untuk mendukung proses kerja bisnis. Permintaan akan proyek perangkat lunak di bidang bisnis semakin banyak dan semakin dibutuhkan untuk proses bisnis perusahaan. Namun dengan semakin banyaknya proyek pengembangan perangkat lunak banyak perusahaan yang kewalahan dan mengalami kegagalan dalam proses pengembangan tersebut, hal ini terjadi akibat buruknya perencanaan proyek (wijaya, 2011).

Menurut Standish Group tingkat kegagalan pada proyek pengembangan perangkat lunak sangatlah tinggi. Pada tahun 2002 sampai 2010 hanya terdapat 37% proyek teknologi informasi yang sukses (Standish Group, 2011). Salah satu prosentase terbesar yang menyebabkan kegagalan proyek pengembangan perangkat lunak yaitu kurang baiknya perencanaan proyek, yakni mempunyai prosentase sebesar 39% (Bull Survey, 1998). Pendapat senada juga didapatkan dari survey yang dilakukan oleh KPMG (kuwabara payne mckenna blumberg) Canada, Salah satu penyebab utama kegagalan pelaksanaan proyek pengembangan perangkat lunak yaitu buruknya perencanaan proyek (KPMG Canada, 1997). Untuk mengurangi kegagalan proyek perangkat lunak tersebut, diperlukan perbaikan dalam perencanaan proyek. Salah satu cara untuk memperbaiki perencanaan proyek yaitu dengan melakukan estimasi usaha (*effort*) pada proyek pengembangan perangkat lunak menggunakan metode yang tepat.

Definisi estimasi perangkat lunak menurut (Galorath, 2016) adalah suatu kegiatan melakukan prediksi atau ramalan mengenai keluaran dari sebuah proyek dengan meninjau jadwal, biaya, bahkan hingga risiko yang akan ditanggung serta

effort dalam proyek tersebut. *Effort* merupakan kerja *real* yang dilakukan dalam menyelesaikan suatu proyek. Satuannya adalah *man-days* atau *man-hours*, walaupun estimasi tidak mungkin dapat menghasilkan sebuah hasil yang sangat akurat, tetapi ketidakakuratan tersebut dapat diminimalkan dengan menggunakan beberapa metode yang sesuai dengan proyek yang akan dilakukan estimasi (Galorath, 2016)

Objek pada penelitian ini adalah SMP (Sistem Manajemen Penyuratan). Sistem ini merupakan salah satu menu yang terdapat pada website Perpustakaan Daerah Kabupaten Jember (<http://perpustakaan.jemberkab.go.id/>) yang beroperasi sejak tahun 2017, SMP ini dibangun untuk mempermudah proses kerja dalam pengelolaan arsip surat masuk dan surat keluar, untuk lebih mengefisienkan pengelolaan data secara terkomputerisasi untuk menunjang pelaksanaan teknis dalam pengelolaan data secara lebih akurat, efektif dan efisien.

Pada penelitian ini, penulis menggunakan metode *Use Case Point* (UCP) yang mempunyai kemampuan untuk memberikan estimasi *effort* yang diperlukan untuk membuat suatu proyek berdasarkan jumlah dan kompleksitas *use case* yang dimiliki oleh proyek perangkat lunak tersebut (Karner, 1993). Metode *Use Case Point* (UCP) berperan penting dalam konteks *early size measurement dan estimasi effort*, karena mempekerjakan *use case* sebagai masukan (input). *Use case* merupakan bentuk populer yang mewakili persyaratan fungsional, menurut survei yang dilakukan oleh Neill dan Laplante pada tahun 2003, 50% dari proyek memiliki persyaratan fungsional yang disajikan sebagai scenarios atau *use case*, dalam tahap awal pengembangan perangkat lunak.

Perkembangan *information technology* (IT) yang pesat turut membawa perubahan signifikan dalam berbagai bidang, termasuk perpustakaan. Pemanfaatan IT dalam mengelola atau menjalankan operasional perpustakaan dapat meningkatkan kualitas layanannya, yakni dari segi kecepatan dan kualitas informasi yang diberikan, akan tetapi seiring berkembangnya IT, terdapat kendala pada pembuatan Sistem itu sendiri, diantaranya kurang efisiensinya waktu dan SDM yang di butuhkan pada pembuatan Sistem tersebut sehingga berdampak pada

besarnya biaya yang dikeluarkan pada saat pembuatannya (diana, samosir, & Widiyasa, 2008)

Pada penelitian ini metode *Fuzzy Use Case Point* digunakan pada Sistem estimasi proyek pengembangan pada perpustakaan daerah Kabupaten Jember, dikarenakan sistem pada perpustakaan daerah Kabupaten Jember masih kurang efisien dilihat dari aspek pembuatan sistem yang dinilai terlalu lama karena sumber daya manusia yang digunakan sedikit, sehingga biaya yang dibutuhkan semakin mahal.

Berdasarkan uraian diatas, penulis melihat adanya masalah dalam melakukan estimasi proyek pengembangan perangkat lunak, sehingga dibutuhkan sebuah sistem yang dapat menghitung nilai dari sebuah pengembangan perangkat lunak. Sistem ini diharapkan dapat membantu merumuskan beberapa permasalahan yang berkaitan dengan pengembangan sistem yang selama ini belum dilakukan estimasi dengan baik.

1.2 Rumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang mendefinisikan beberapa permasalahan yang harus diselesaikan dalam penulisan ini adalah sebagai berikut:

1. Bagaimana cara menerapkan *Fuzzy Use Case Point* pada estimasi *effort* proyek pengembangan perangkat lunak?
2. Bagaimana mengimplementasikan hasil dari pengolahan data berdasarkan *Fuzzy Use Case Point* pada pembuatan sistem estimasi *effort* proyek pengembangan perangkat lunak?

1.3 Tujuan

Tujuan dalam penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan. Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Merancang dan membangun system estimasi proyek pengembangan perangkat lunak berdasarkan *Fuzzy Use Case Point*.

2. Mengimplementasikan hasil dari pengolahan data berdasarkan *Fuzzy Use Case Point* pada pembuatan system estimasi proyek pengembangan perangkat lunak.

1.4 Batasan Masalah

Batasan masalah digunakan untuk menghindari penyimpangan dalam proses penelitian dan pembangunan Sistem Estimasi *Effort* Proyek Pengembangan Perangkat Lunak Menggunakan *Fuzzy Use Case Point*. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem yang di bangun hanya dapat merekomendasikan estimasi waktu dan SDM (Sumber Daya Manusia) yang dibutuhkan berdasarkan *Fuzzy Use Case Point*.
2. Sistem yang di bangun ini bekerja menggunakan inputan berdasarkan *Use Case*
3. Sistem yang dibangun berbasis *website*.

1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

1. Pendahuluan
Bab ini memuat uraian tentang latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika penulisan skripsi yang masing-masing tertuang secara eksplisit dalam sub bab tersendiri.
2. Tinjauan Pustaka
Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi, teori dan informasi yang berkaitan dengan masalah dalam penelitian.
3. Metodologi Penelitian
Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.
4. Pengembangan Sistem
Bab ini akan menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem yang digunakan dalam proses pengembangan

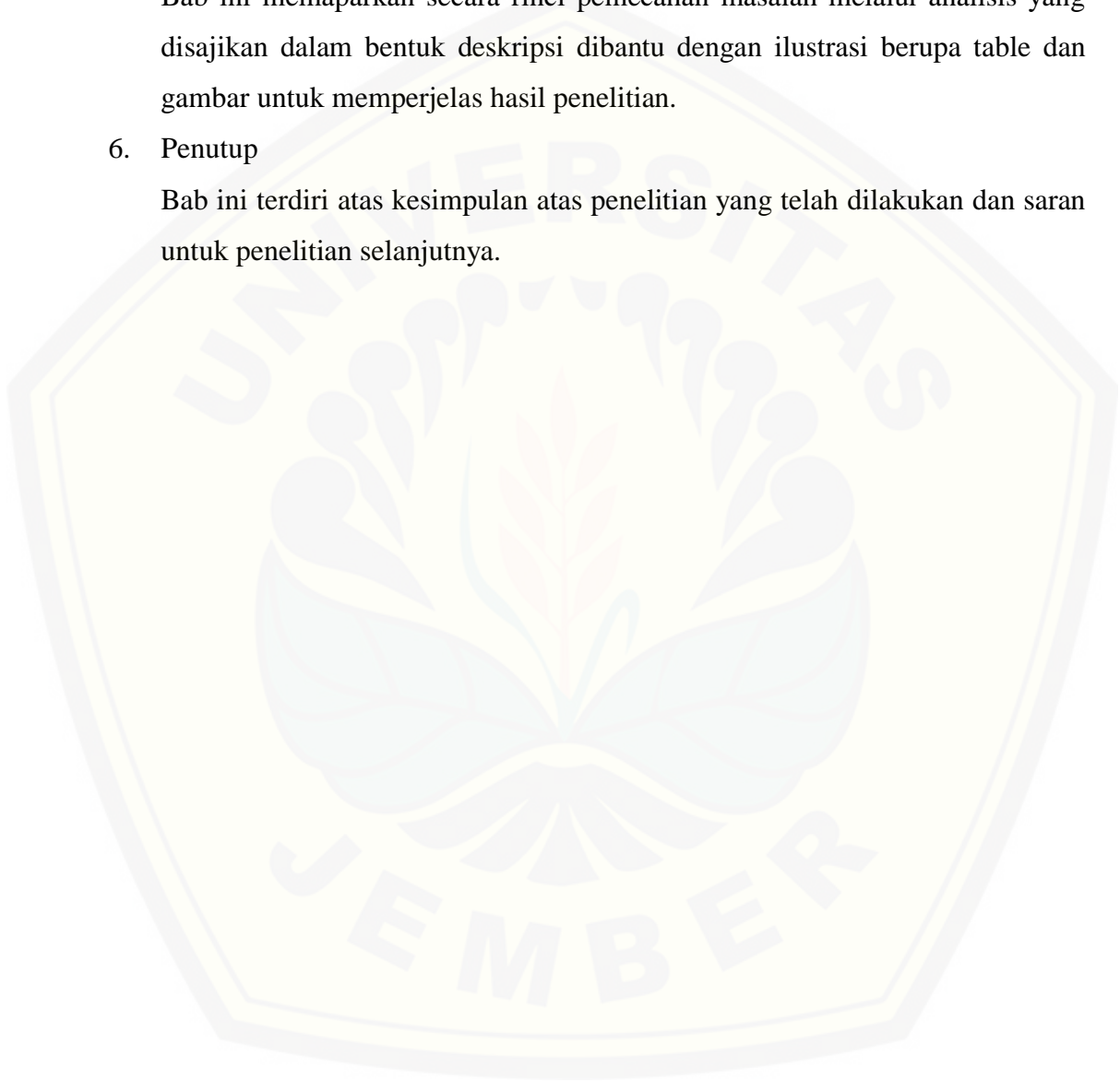
pencocokan pada graf untuk sistem pengambilan keputusan dalam penempatan bidang kerja. Tahapan analisis hingga pengujian dilakukan sesuai dengan metode pengembangan *waterfall*.

5. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yang disajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa table dan gambar untuk memperjelas hasil penelitian.

6. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.



BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dijelaskan teori-teori dan pustaka yang digunakan dalam penelitian. Teori yang dibahas adalah teori estimasi pengembangan proyek perangkat lunak, *Use Case Point*, logika *fuzzy*.

2.1 Proyek Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak adalah suatu strategi pengembangan yang memadukan proses, metode, dan perangkat (*tools*) yang bertujuan untuk mendapatkan perangkat lunak yang berkualitas. Metode-metode pengembangan perangkat lunak, memberikan teknik untuk membangun perangkat lunak yang berkaitan dengan serangkaian tugas yang luas yang menyangkut analisis kebutuhan, konstruksi program, desain, pengujian, dan pemeliharaan. Pengembangan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan/memperbaiki sistem yang telah ada (Budi, Yoga Siswa, & Abijono, 2016)

Sistem yang lama perlu diperbaiki atau diganti disebabkan karena beberapa hal, yaitu:

1. Adanya permasalahan-permasalahan (*problems*) yang timbul di sistem yang lama
2. Untuk meraih kesempatan-kesempatan (*opportunities*)
3. Adanya instruksi-instruksi (*directives*)

2.2 Use Case Diagram

Use Case diagram adalah suatu model yang sangat fungsional dalam sebuah sistem yang menggunakan aktor dan *Use Case* sedangkan pengertian dari *Use Case* sendiri adalah layanan atau fungsi-fungsi yang tersedia pada sistem untuk penggunaannya (samiran, 2017).

Use Case diagram menggambarkan efek fungsionalitas yang telah diharapkan oleh sistem. *Use Case diagram* dapat sangat membantu bila kita sedang menyusun Requirement sebuah sistem, mengkomunikasikan sebuah rancangan aplikasi

dengan konsumen, serta merancang *test case* untuk semua fitur yang ada pada sistem (samiran, 2017).

Use Case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem secara keseluruhan yang akan dibuat. *Use Case* diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dengan pengertian yang cepat, *Use Case* diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Urva & Siregar, 2015).

2.3 Use Case Point

Use Case Point adalah merupakan suatu metode estimasi dalam pengembangan perangkat lunak (Karner, 1993). Metode ini memberikan sebuah estimasi yang mendekati estimasi sebenarnya yang dihasilkan dari pengalaman pembuatan atau pengembangan perangkat lunak.

Pada penelitian (Carroll, Edward R, 2005), disebutkan “setelah menerapkan proses yang cukup besar, ratusan proyek perangkat lunak (rata-rata 60 *man-hours*), ditunjukkan metrik yang membuktikan akurasi yang memperkirakan kurang dari 9% dari deviasi sebenarnya untuk biaya diperkirakan pada 95% dari proyek-proyek tersebut (Dewi, Subriadi, & Sholiq, 2015)

Tahapan perhitungan yang dirumuskan (Karner, 1993) adalah sebagai berikut:

1. Menghitung nilai *Unadjusted Use Case Point* (UUCP)

Rumus yang digunakan yaitu

$$UUCP = UUCW + UAW \dots\dots\dots(1)$$

Dimana untuk mendapatkan UUCP, harus dihitung terlebih dahulu *Unadjusted Use Case Weight* (UUCW) dan *Unadjusted Actor Weight* (UAW). Masing-masing *Use Case* memiliki tipe dan bobot. Tipe *Use Case* ditentukan berdasarkan jumlah transaksi per *Use Case*, dalam hal ini digunakan metode *Roundtrip* (Jacobson, Christorson, Jonsson, & Overgaard, 1992) tipe dan pembobotan usecase yang dapat dilihat pada Tabel 2.1

Tabel 2.1 Tipe dan pembobotan *Use Case* (Karner, 1993)

Type Use Case	Bobot	Keterangan
Sederhana	5	Jumlah transaksi tidak lebih dari 3
Sedang	10	Jumlah transaksi antara 4 dan 7
Sulit	15	Jumlah transaksi lebih dari 7

UUCW = (Total Use Case Tipe Sederhana x 5) + (Total Use Case Tipe Sedang x 10) + (Total Use Case Tipe Sulit x 15).

2. Menghitung *Technical Complexity Factor* (TCF)

Menghitung faktor kompleksitas teknis yaitu dengan cara tiap-tiap *technical factor* dikalikan dengan bobot yang sudah di tetapkan Pada Tabel 2.2 lalu dijumlahkan secara keseluruhan dengan rumus persamaan 2, sebagai berikut :

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

.....(2)Dimisalkan M adalah graf bipartit dengan

Tabel 2.2 Data Nilai Technical Factor (*Karner, 1993*)

Kode	No.	Deskripsi	Bobot	Nilai (0 - 5)	TF
T1	1.	<i>Distributed System Required</i>	2		
T2	2.	<i>Response Time is Important</i>	1		
T3	3.	<i>End User Efficiency</i>	1		
T4	4.	<i>Complex Internal Processing Required</i>	1		
T5	5.	<i>Reusable Code Must Be A Focus</i>	1		
T6	6.	<i>Installation easy</i>	0,5		
T7	7.	<i>Usability</i>	0,5		
T8	8.	<i>Cross-platform support</i>	2		
T9	9.	<i>Easy to change</i>	1		
T10	10.	<i>Highly concurrent</i>	1		
T11	11.	<i>Custom security</i>	1		
T12	12.	<i>Dependence on third-part code</i>	1		
T13	13.	<i>User training</i>	1		

<i>Technical Factor (TF)</i>	
--------------------------------	--

3. Menghitung *Environmental Complexity Factor* (ECF)

Menghitung faktor kompleksitas lingkungan yaitu dengan cara setiap *environmental factor* dikalikan dengan bobot yang sudah di tetapkan Pada Tabel 2.3 lalu dijumlahkan secara keseluruhan dengan rumus persamaan 3, sebagai berikut :

$$ECF = 1,4 + (-0,03 * EF \text{ (Environmental Factor)}) \dots\dots\dots(3)$$

Tabel 2.3 Environmental Factor (*Karner, 1993*)

Kode	No.	Deskripsi	Bobot	Nilai (0 - 5)	EF
E1	1	<i>Familiarity with the Project</i>	1,5		
E2	2	<i>Application Experience</i>	0,5		
E3	3	<i>OO Programming Experience</i>	1		
E4	4	<i>Lead Analyst Capability</i>	0,5		
E5	5	<i>Motivation</i>	1		
E6	6	<i>Stable Requirements</i>	2		
E7	7	<i>Part Time Staff</i>	-1		
E8	8	<i>Difficult Programming Language</i>	-1		
<i>EF</i>					

4. Menghitung *Use Case Point*

Apabila tiga langkah diatas selesai dihitung, maka selanjutnya adalah menghitung nilai UCP. Dari perolehan nilai UCP tersebut, maka *actual Effort* dapat dihitung dengan mengalikan UCP dengan *Effort rate* yang telah ditentukan dengan rumus persamaan 4, sebagai berikut :

$$UCP = UUCP \text{ (Unadjusted Use Case Point) } *TCF \text{ (Technical Complexity Factor) } *ECF \text{ (Environmental Complexity Factor) } \dots\dots\dots(4)$$

Hasil UCP sudah menggambarkan ukuran dari sistem yang akan dibuat, langkah selanjutnya menghitung berapa banyak *effort* atau tenaga yang dibutuhkan.

Nilai *effort* dapat diketahui dari hasil UCP dibagi dengan nilai PHM (*Person Hour Multiplier*) (Hariyanto, 2010). Nilai PHM diperoleh dengan aturan sebagai berikut:

- F1 = Jumlah E1 sampai E6 yang < 3
- F2 = Jumlah E7 sampai E8 yang > 3
- Jika $F1 + F2 \leq 2$ maka PHM = 20
- Jika $F1 + F2 = 3$ atau 4 maka PHM = 28

Jika $F1 + F2 > 4$ maka proyek harus dibatalkan

Ket :

1. F1 : Kondisi Pertama
2. F2 : Kondisi Ke Dua
3. E1 – E13 : Kode pada tabel ECF (*Environmental Complexity Factor*)
4. PHM (*Person Hour Multiplier*)

2.4 Logika Fuzzy

Logika Fuzzy adalah teknik / metode yang dipakai untuk mengatasi hal yang tidak pasti pada masalah – masalah yang mempunyai banyak jawaban. Pada dasarnya Fuzzy logic merupakan logika bernilai banyak/ multivalued logic yang mampu mendefinisikan nilai diantara keadaan yang konvensional seperti benar atau salah, ya atau tidak, putih atau hitam dan lain-lain. Penalaran Logika Fuzzy menyediakan cara untuk memahami kinerja sistem dengan cara menilai input dan output sistem dari hasil pengamatan. Logika Fuzzy menyediakan cara untuk menggambarkan kesimpulan pasti dari informasi yang samar-samar, ambigu dan tidak tepat. Fuzzy logic Pertama kali dikembangkan oleh Lotfi A. Zadeh tahun 1965 (Much. Djunaidi, Andista, & Setiawan, 2005)

2.5 Metode Fuzzy Use Case Point

Konsep logika *fuzzy* diperkenalkan oleh Prof. Lotfi Zadeh dari Universitas California di Berkeley pada 1965. Penamaan *Fuzzy Use Case Point* dikemukakan oleh Hariyanto dan Wahono pada penelitiannya yang berjudul Estimasi Proyek Pengembangan Perangkat Lunak dengan *Fuzzy Use Case Points* pada tahun 2015.

Metode *fuzzy Use Case Points* merupakan modifikasi metode *Use Case Points* dengan menggunakan logika *fuzzy* untuk memberikan nilai klasifikasi *Use Case* (Hariyanto, 2010), Tercantum pada Tabel 2.4 berikut :

Tabel 2.4 Tabel Fuzzy Use Case Point (Hariyanto & wahono, 2015)

Transaksi Pada <i>Use Case</i>	Nilai Fuzzy <i>Use Case Point</i>
1	5
2	5
3	6,45
4	7,50
5	8,55
6	10
7	11,4
8	12,5
9	13,6
10	15
11	15
12	15

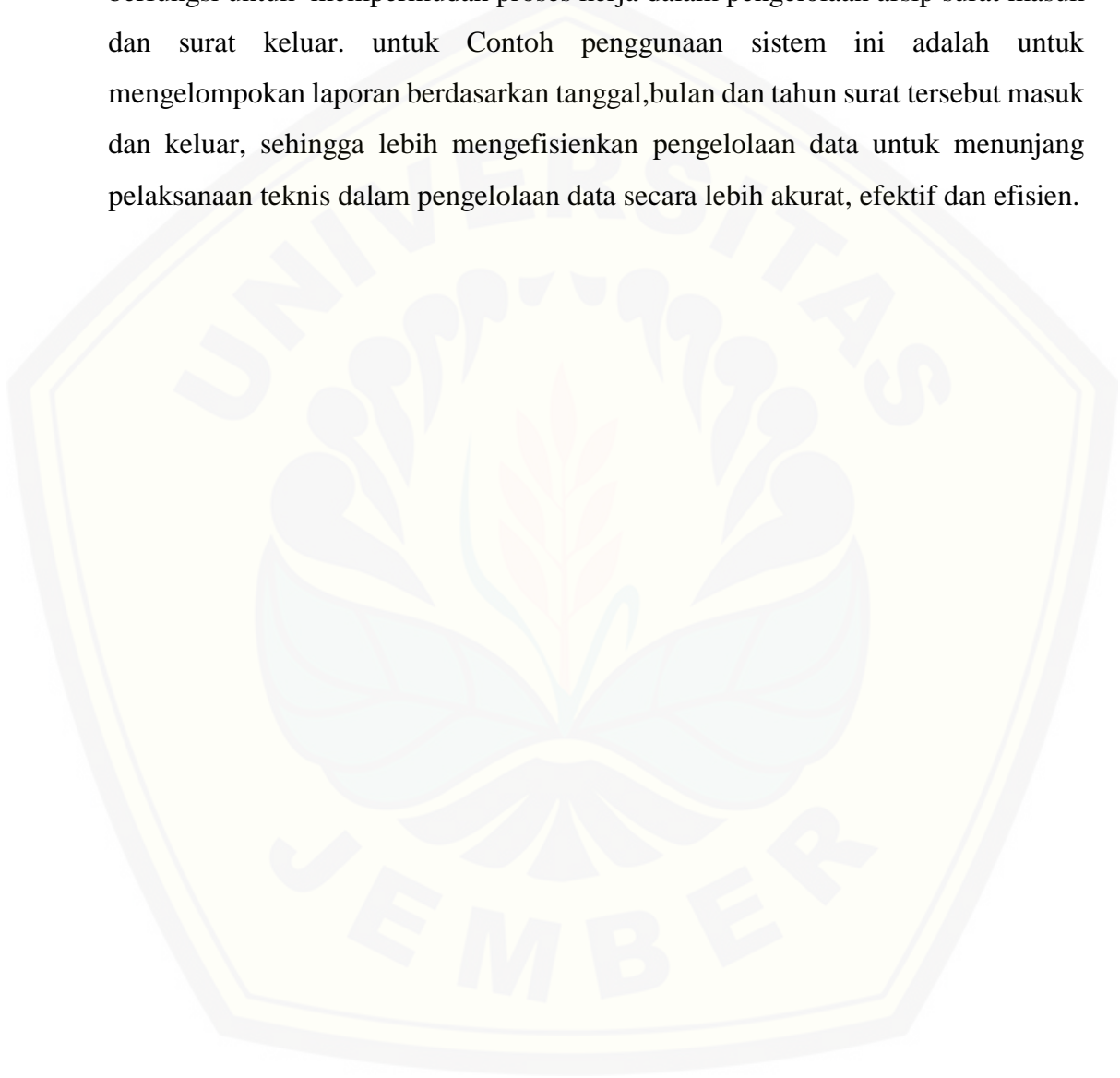
Nilai *Fuzzy Use Case Point* dapat diperoleh dengan mengalikan *Unadjusted Actor Weight*, *Unadjusted Use Weight*, *Technical Complexity Factor*, dan *Environmental Complexity Factor*.

2.7 Pengertian Kearsipan

Kearsipan merupakan bagian yang sangat penting dari pekerjaan kantor. Setiap pekerjaan dan kegiatan di perkantoran memerlukan data dan informasi. Salah satu sumber data adalah arsip, karena arsip adalah bukti dan rekaman dari suatu kegiatan. Arsip biasanya diolah dengan cara manual dan dengan cara elektronik menggunakan komputer. (Ni'matussholihah, 2015).

2.8 Sistem Manajemen Penyuratan

Pada penelitian ini penulis meneliti Sistem Kearsipan. sistem ini memiliki beberapa fitur diantaranya agenda masuk dan agenda keluar. Fitur-fitur tersebut berfungsi untuk mempermudah proses kerja dalam pengelolaan arsip surat masuk dan surat keluar. untuk Contoh penggunaan sistem ini adalah untuk mengelompokkan laporan berdasarkan tanggal, bulan dan tahun surat tersebut masuk dan keluar, sehingga lebih mengefisienkan pengelolaan data untuk menunjang pelaksanaan teknis dalam pengelolaan data secara lebih akurat, efektif dan efisien.



BAB 3. METODOLOGI PENELITIAN

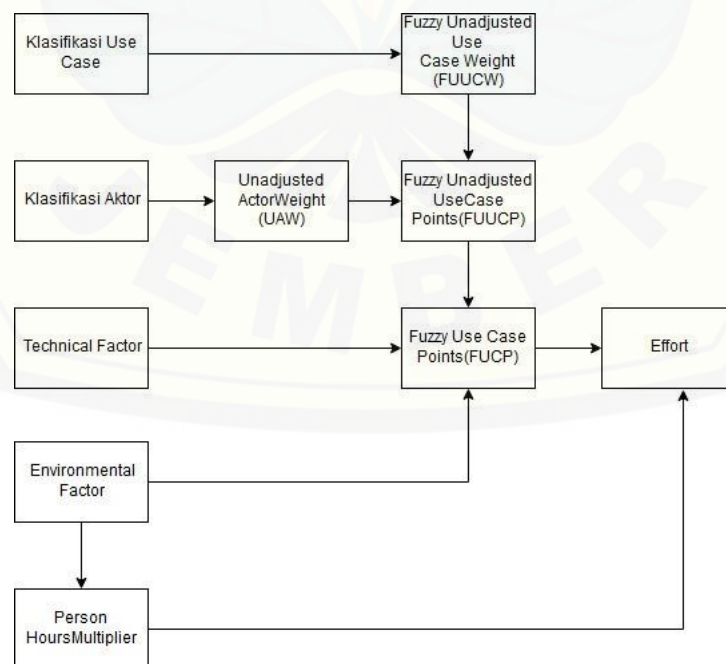
Bab ini menjelaskan tentang metode yang digunakan dalam merancang dan membangun sistem informasi ini.

3.1 Waktu Penelitian

Penelitian dilakukan dengan menggunakan Informasi SMP dari Perpustakaan Daerah Kabupaten Jember. Waktu penelitian dilakukan selama 5 bulan, dimulai bulan Mei 2017 sampai dengan September 2017.

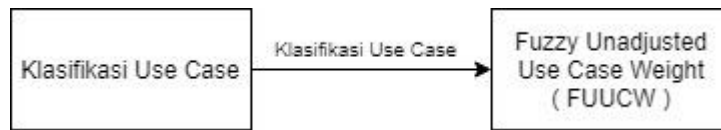
3.2 Tahapan *Fuzzy Use Case Point*

Penelitian ini dilakukan dalam beberapa tahapan diantaranya tahap pengumpulan data yang akan dianalisis kemudian dibuat perancangan serta di implementasikan. Tahapan penelitian ini digunakan untuk mencapai tujuan penelitian yaitu membantu dalam proses pembangunan sistem. Tahap penelitian dapat dilihat pada Gambar 3.1,



Gambar 3.1 Tahap penelitian

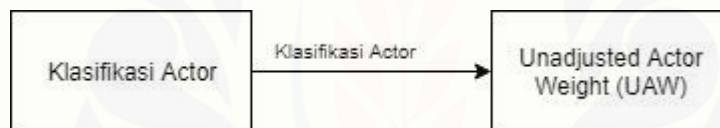
3.2.1 Klasifikasi Use Case Point



Gambar 3.2 Klasifikasi Use Case

Unadjusted Use Case Weight menunjukkan kompleksitas *use case* pada Sistem Perpustakaan yang diukur dari jumlah transaksi dalam sebuah *use case*. Setiap *Use Case* di dalam sistem memiliki nilai *Fuzzy Unadjusted Use Case Weight* diperoleh dari penjumlahan bobot tiap *use case*. Pada Gambar 3.2 Menunjukkan Proses pengelompokan *Use Case* berdasarkan jumlah transaksi dalam sebuah *Use Case* sesuai dengan tabel 2.4.

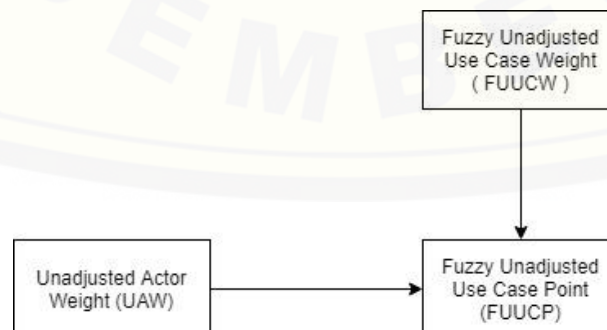
3.2.2 Klasifikasi Aktor



Gambar 3.3 Klasifikasi Aktor

Nilai UAW adalah total nilai yang didapatkan dari total aktor dari masing – masing tipe yang ada pada *use case* dikalikan bobot masing – masing tipe atau tingkat kompleksitasnya, Kriteria Tipe, bobot sesuai dengan tabel 2.2.

3.2.3 Fuzzy Unadjusted Use Case Point



Gambar 3.4 Fuzzy Unadjusted UseCase Point

ukuran kompleksitas software yang didapatkan dari penjumlahan nilai bobot *use case* yang tercantum tabel 2.4 atau *Fuzzy unadjusted use case weight*

(FUUCW) dan nilai bobot aktor yang ada pada use case yang tercantum tabel 2.2 atau unadjusted actor weight (UAW) sesuai Persamaan 1.

3.2.4 *Technical Factor*

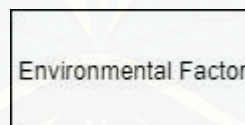


Gambar 3.5 *Technical Factor*

TCF merupakan salah satu faktor yang diterapkan pada perkiraan ukuran perangkat lunak untuk memperhitungkan pertimbangan teknis dari sistem. Faktor ini menggambarkan ekspektasi yang diharapkan user terhadap software yang akan dibangun. Technical complexity factor dibobot tergantung dari bagaimana tingkat kesulitan sistem yang akan dibangun

Nilai TF di dapatkan melalui wawancara kepada Narasumber menggunakan pertanyaan – pertanyaan sesuai tabel 2.2. Kemudian untuk Mencari Nilai TCF Menggunakan Persamaan 2.

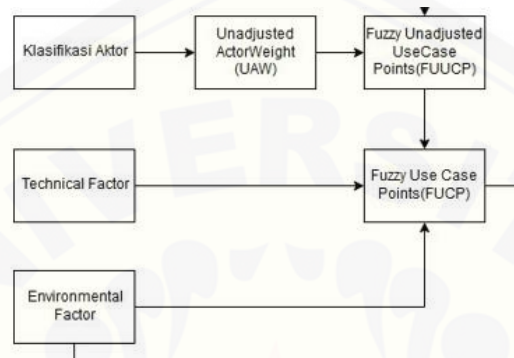
3.2.5 *Environmental Factor*



Gambar 3.6 *Environmental Factor*

Untuk menghitung EF, nilai dari masing-masing faktor lingkungan dihitung berdasarkan tingkat pengalaman tim. Hal ini ditentukan dengan menetapkan skor antara 0 (tidak ada pengalaman) dan 5 (ahli) untuk masing-masing 8 faktor lingkungan yang tercantum dalam tabel 2.3 Kemudian untuk mencari ECF menggunakan Persamaan 3.

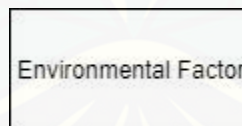
3.2.6 Fuzzy Use Case Point



Gambar 3.7 Fuzzy Use Case Point

FUCP didapatkan berdasarkan perhitungan dari *Fuzzy Unadjusted Use Case Point*, *Technical Factor*, *Environmental Factor* dengan mengalikan ketiganya maka di dapatkan nilai FUCP Menggunakan Persamaan 4.

3.2.7 Person Hours Multiplier



Gambar 3.8 Person Hours Multiplier

Nilai PHM diperoleh berdasarkan data dari *Environmental Factor* dengan aturan (Karner, 1993) sebagai berikut:

- F1 = Jumlah E1 sampai E6 yang < 3
- F2 = Jumlah E7 sampai E8 yang > 3
- Jika $F1 + F2 \leq 2$ maka PHM = 20
- Jika $F1 + F2 = 3$ atau 4 maka PHM = 28
- Jika $F1 + F2 > 4$ maka proyek harus dibatalkan

Keterangan :

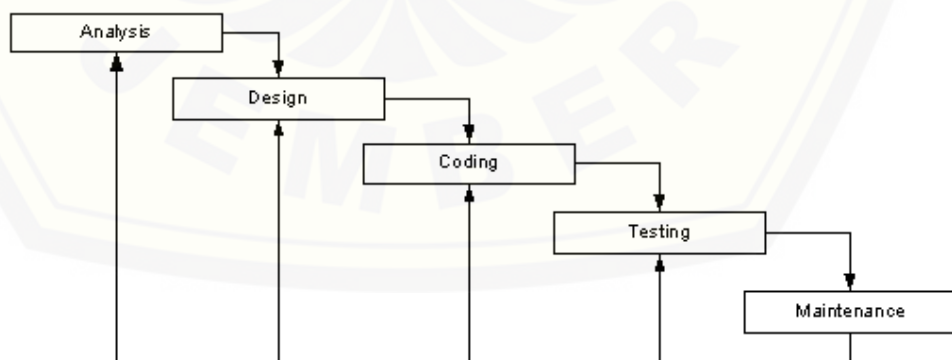
5. F1 : Kondisi Pertama
6. F2 : Kondisi Ke Dua
7. E1 – E13 : Kode pada tabel ECF (*Environmental Complexity Factor*)
8. PHM (*Person Hour Multiplier*)

3.2.8 Effort

Hasil FUCP sudah menggambarkan ukuran dari sistem yang akan dibuat, langkah selanjutnya menghitung berapa banyak *effort* yaitu tenaga dan waktu yang dibutuhkan. Nilai *effort* dapat diketahui dari hasil FUCP dibagi dengan nilai PHM (*Person Hour Multiplier*) (Dewi, Subriadi, & Sholih, 2015).

3.3 Pengembangan

Penelitian ini dikembangkan dengan menggunakan model pengembangan *Software Development Life Cycle (SDLC) waterfall*. *Waterfall* digunakan karena keuntungan dari model ini yaitu kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap, sehingga tidak berfokus pada tahapan tertentu selain itu bila kebutuhan sistem dapat didefinisikan dengan baik maka pembangunan sistem akan berjalan lancar. Model *Waterfall* merupakan model yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode, *test*, dan pemeliharaan (Pressman, 2001). Model *Waterfall* dapat dilihat pada Gambar 3.9,



Gambar 3.9 Model *Waterfall*

Sumber : (Pressman, 2001)

3.3.1 Analisis Kebutuhan

Analisis kebutuhan adalah proses perancangan perangkat lunak yang dimulai dari proses merumuskan solusi dari setiap permasalahan yang muncul. Tahapan yang dilakukan dalam proses analisis kebutuhan meliputi tahap pengumpulan data, analisis data, serta menentukan kebutuhan fungsional dan non fungsional dari sistem yang dibangun. Tahap yang digunakan adalah sebagai berikut:

a. Tahap Pengumpulan Data

Pengumpulan data yang akan dilakukan pada penelitian ini adalah sebagai berikut:

1. Wawancara

Tahap wawancara merupakan tahap pengumpulan data yang dilakukan dengan melakukan komunikasi secara langsung dengan narasumber terkait untuk memperoleh data-data yang dibutuhkan untuk penelitian diantaranya data untuk mengklasifikasikan *use case* dan *actor*, data untuk menemukan nilai *Technical Factor* dan *Enviromental Factor*. Pada penelitian ini melakukan wawancara dengan dua narasumber yaitu Pak Agung dan Pak Fathur selaku *programmer* di Perpustakaan Daerah Kab. Jember untuk mendapatkan data yang diperlukan dalam penelitian ini.

b. Tahap Pengolahan Data

Tahap pengolahan data dimulai dengan menelaah data secara keseluruhan yang telah diperoleh dari tahap pengumpulan data. Pengumpulan data diperoleh dari hasil wawancara dengan narasumber, data yang telah terkumpul digunakan untuk mengidentifikasi kebutuhan fungsional dan kebutuhan non-fungsional sistem.

3.3.2 Desain

Tahap ini meliputi tahap desain sistem dengan pembuatan diagram menggunakan *tools UML Visual Paradigm*. Diagram-diagram tersebut akan digunakan sebagai acuan pembuatan sistem pada tahap implementasi. Diagram-diagram yang digunakan yang digunakan sebagai berikut:

1. *Business process* merupakan diagram yang menggambarkan proses dari sebuah sistem yang meliputi *input*, *output*, dan *goal* yang merupakan tujuan dari sebuah sistem yang dibangun.
2. *Use Case Diagram* menggambarkan fungsionalitas sistem dan hubungan antara sistem dengan aktor berkaitan dengan fungsi atau tugas yang dilakukan oleh aktor. *Use Case Diagram* juga menggambarkan hak akses dari pengguna.
3. *Scenario* menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu event tertentu.
4. *Activity Diagram* mendeskripsikan aktifitas yang dibentuk dalam suatu operasi
5. *Sequence Diagram* digunakan untuk menunjukkan rangkaian pesan yang dikirim dan diterima antar *object*.
6. *Class Diagram* digunakan untuk menggambarkan relasi antar objek dan struktur sematik yang umum.
7. *Entity Realtionship Diagram* digunakan untuk menggambarkan relasi data yang terdapat dalam *database*.

3.3.3 Implementasi Sistem

Implementasi perangkat lunak ini dilakukan dengan mengacu pada perancangan desain sistem yang telah dibuat sebelumnya. Implementasi perangkat lunak menggunakan bahasa pemrograman PHP yang akan digunakan untuk membangun sistem berbasis website dan manajemen basis data dengan menggunakan *Database Management System (DBMS) MySQL*.

3.3.4 Pengujian Sistem

Merebaknya isu kualitas pengembangan *software*, yang dipicu oleh tingkat kecenderungan gagalnya proyek-proyek pengembangan *software* yang cukup tinggi. Pengujian program yang digunakan untuk sistem informasi ini yaitu dengan menggunakan metode pengujian *White Box Testing* dan *Black Box Testing*. Pengujian menggunakan *White Box Testing (Glass box)* dan *Black Box Testing*

Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar.

3.3.5 Pemeliharaan

Pemeliharaan merupakan tahap terakhir dalam metode *waterfall*. Sistem yang berhasil dibuat tersebut masih membutuhkan adanya pemeliharaan. Pemeliharaan yang dimaksud dalam hal ini adalah bertujuan untuk peningkatan kinerja dari sistem yang digunakan sebagai kebutuhan baru. Sehingga apabila nantinya user menemukan adanya *bug* atau gangguan pada sistem tersebut, *user* dapat melaporkan secara langsung kepada pihak pengembang sistem (*developer*) untuk segera ditangani.

BAB 4. PENGEMBANGAN SISTEM

Bab ini akan menguraikan mengenai analisis kebutuhan sistem, desain, implementasi, dan pengujian sistem yang digunakan dalam proses pengembangan sistem penunjang keputusan penempatan bidang kerja. Tahapan analisis hingga pengujian dilakukan sesuai dengan model pengembangan *waterfall*.

4.1 Analisis Kebutuhan Sistem

Berdasarkan metode pengembangan sistem model *waterfall*, tahapan awal yang dilakukan adalah tahapan analisis. Analisis kebutuhan yang dilakukan sesuai dengan data yang diperoleh pada tahapan yang telah dijelaskan pada bab metodologi penelitian subbab 3.3.1. Tahapan analisis ini dilakukan untuk memperoleh kebutuhan-kebutuhan dari sistem yang dibangun, baik berupa kebutuhan fungsional maupun kebutuhan non-fungsional.

4.1.1 *Statement of Purpose (SOP)*

Sistem estimasi *effort* proyek pengembangan perangkat lunak menggunakan *fuzzy Use Case Point* ini akan diimplementasikan pada Perpustakaan Daerah Kabupaten Jember. Aktor dalam sistem ini adalah Assessor disetiap proyek yang dijalankan untuk diestimasi. Sistem ini menggunakan *Fuzzy Use Case Point* untuk menghitung nilai *effort rate* dalam suatu proyek pengembangan sistem. Sistem ini mengolah data dengan mengklasifikasikan *Use Case* dan aktor yang telah ditentukan. Selain itu sistem akan menghitung nilai aktual *effort* sistem melakukan analisis korelasi dan persamaan linier sehingga memunculkan *effort rate*.

4.1.2 Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem mampu difungsikan sesuai dengan tujuan dan kebutuhan

pengguna terhadap sistem itu sendiri (Arista, A. B., & Pantjawati, 2014).

Kebutuhan fungsional dari sistem ini yaitu:

1. Assessor mampu melakukan *login* untuk *user* yang akan masuk sistem.
2. Assessor mampu mengelola data proyek
3. Assessor mampu mengelola data UnAjusted Actor Weight.
4. Assessor mampu mengelola data Fuzzy Un-Adjusted *Use Case* Weight.
5. Assessor mampu mengelola data Technical Complexity.
6. Assessor mampu mengelola data Environmental Complexity.
7. Assessor dapat menampilkan hasil estimasi proyek.
8. Assessor mampu melakukan *logout* untuk *user* yang akan keluar dari sistem.

4.1.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan fitur-fitur yang dimiliki untuk mendukung sistem dalam memenuhi fungsionalitasnya untuk dapat memenuhi kebutuhan dari pengguna. Kebutuhan non-fungsional dari sistem ini yaitu:

1. Sistem memiliki batasan hak akses pengguna dengan menggunakan *username* dan *password*.
2. Sistem berbasis *website*.

4.1.4 Fungsi Sistem

Fungsi utama dari sistem yang dibangun dalam penelitian ini terletak pada fitur login sesuai dengan hak akses dari setiap pengguna dalam sistem. Ketika pengguna melakukan login, maka sistem melakukan autentifikasi *username* dan *password* pengguna. Selanjutnya sistem akan menyajikan tampilan sistem sesuai dengan hak akses pengguna yang meliputi:

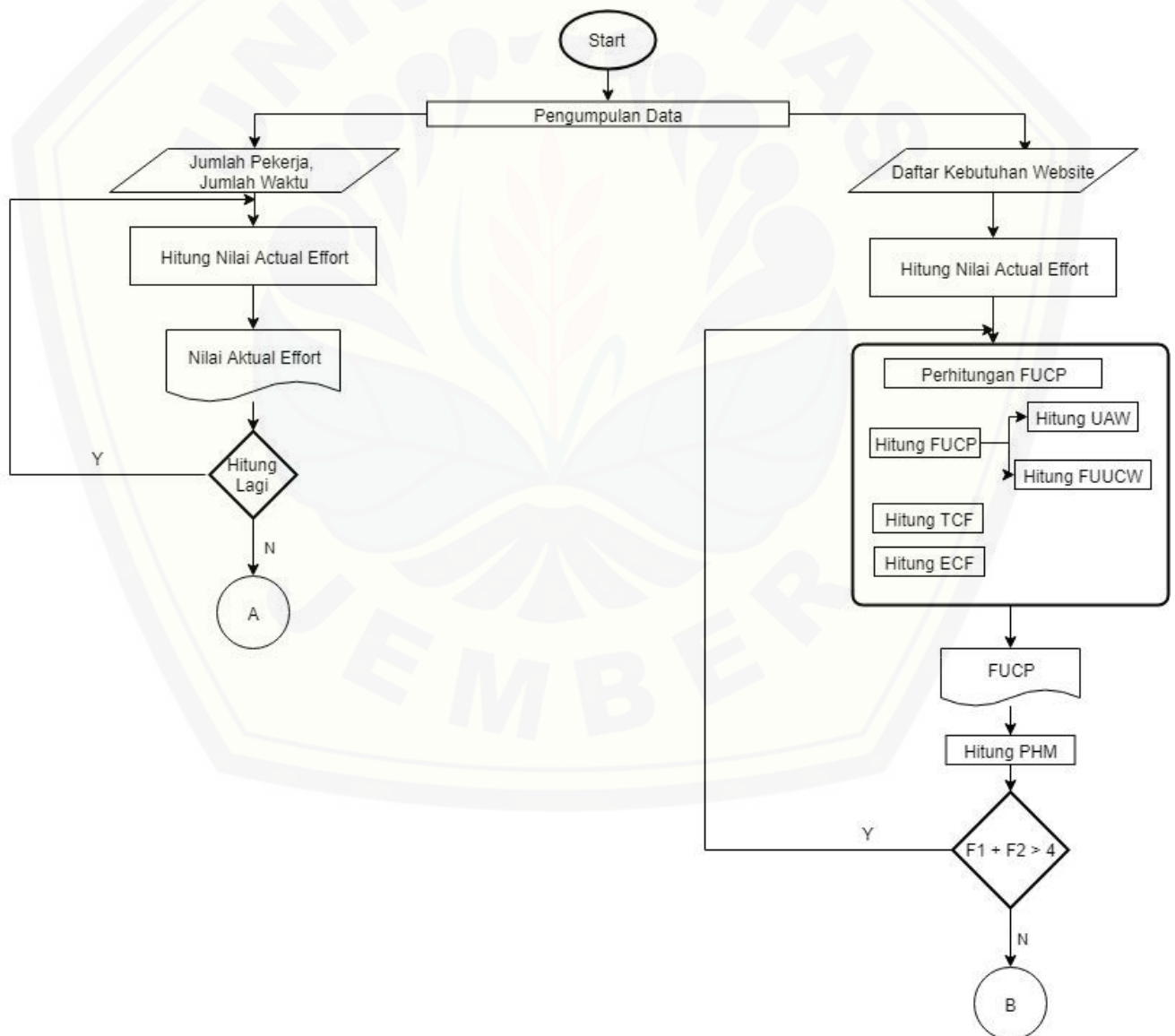
1. Assessor

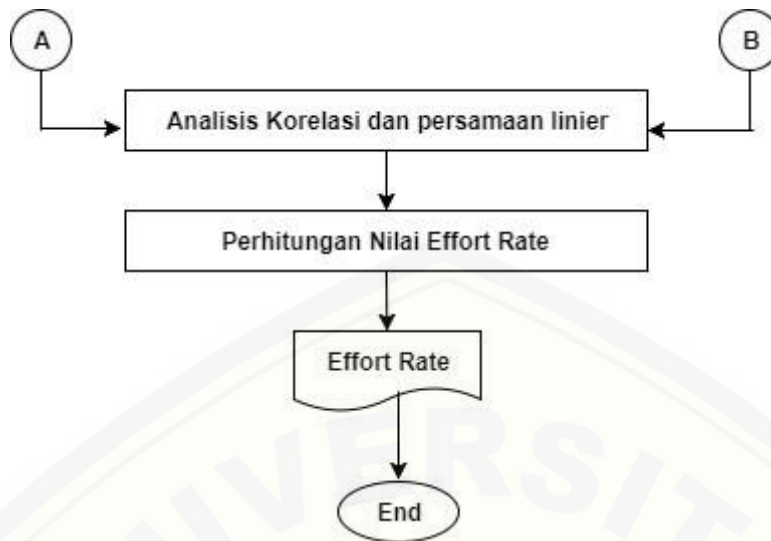
Assessor merupakan pengguna sistem yang telah terdaftar dan ketika login berhasil maka sistem akan menampilkan halaman *dashboard* dan halaman-halaman yang sesuai dengan level Assessor. Assessor dapat mengakses halaman *dashboard* data unajusted actor weight, halaman fuzzy un-adjusted *Use Case* weight, halaman

technical complexity, halaman environmental complexity. Setiap assessor bertanggung jawab atas semua proyek yang dibuat.

Desain Sistem

Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem yaitu tahap perencanaan pembangunan sistem yang dapat digambarkan dengan desain sistem. Desain pada sistem ini meliputi *business process*, *use case diagram*, *use case skenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*. Alur pengolahan data dapat dilihat pada Gambar 4.1





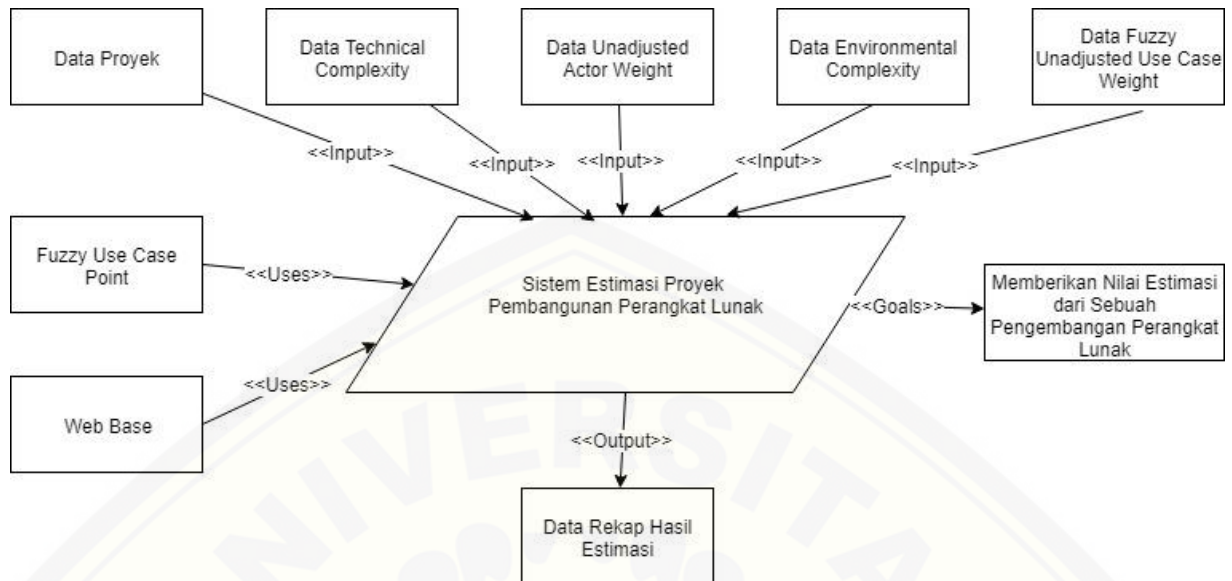
Gambar 4.1 *Flowchart* sistem

Gambar 4.1 menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. menjelaskan bagaimana proses data diolah, mulai dari pengumpulan data sampai mendapatkan *effort rate*.

4.2.1 *Business Process*

Business process merupakan diagram yang menggambarkan kebutuhan data yang dibutuhkan oleh sistem. *Business process* didalamnya terdiri atas beberapa poin diantaranya:

1. *Input* : Data yang dimasukkan ke dalam sistem
2. *Output* : Data yang dihasilkan oleh sistem
3. *Goal* : Tujuan dibangun suatu sistem
4. *Used* : *Platform* yang menjadi basis sistem
5. *Process* : Sistem yang bekerja



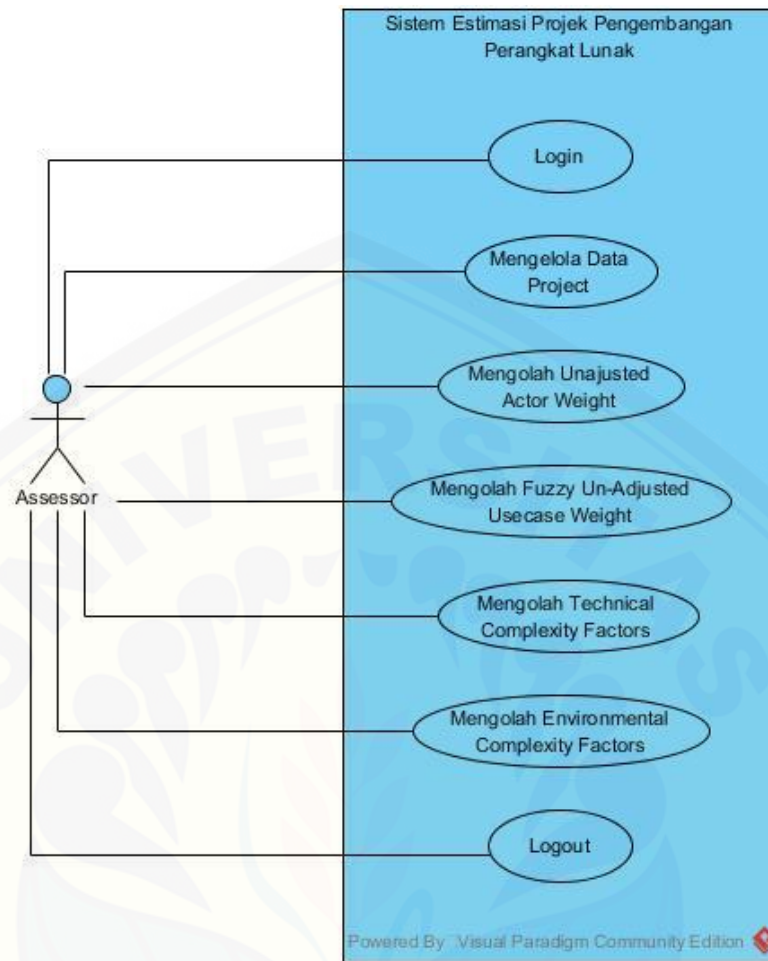
Gambar 4.2 Bussiness Process

Gambar 4.2 merupakan *business process* dari sistem estimasi proyek pembangunan perangkat lunak. *Business process* menjelaskan proses *input*, *output*, *goal*, dan *uses* yang diaplikasikan ke dalam sistem.

Output data hasil estimasi proyek pembangunan perangkat lunak didapatkan dari pengolahan data proyek dan pengolahan evaluasi estimasi effort Unadjusted actor weight, fuzzy Unadjusted *Use Case* weight, technical complexity, dan environmental complexity.

4.2.2 Use Case Diagram

Use case diagram merupakan pemodelan yang dibuat untuk dapat menggambarkan interaksi antara aktor dengan sistem estimasi effort proyek pembangunan perangkat lunak. Melalui *use case diagram* dapat diketahui interaksi yang dapat dilakukan aktor terhadap sistem sesuai dengan hak akses yang dimiliki oleh masing-masing aktor atau pengguna. *Use case diagram* ditunjukkan pada Gambar 4.3,



Gambar 4.3 Use case Diagram

Gambar 4.3 menunjukkan *use case diagram* sistem estimasi proyek pembangunan perangkat lunak dengan Assessor sebagai aktor. Setiap Assessor memegang kendali atas tiap-tiap proyek yang dibuat.

Use case sistem estimasi proyek pembangunan perangkat lunak ini mempunyai penjelasan berupa tabel definisi aktor yang menggambarkan tugas-tugas aktor dalam mengoperasikan sistem tersebut dapat dilihat pada Tabel 4.1 Definisi Aktor. Penjelasan lainnya yaitu disebut definisi *use case* yang menggambarkan fungsionalitas dari setiap *Use Case* dapat dilihat pada Tabel 4.2 Definisi *Use case*.

Tabel 4.1 Definisi Aktor

No.	Aktor	Definisi Tugas
1.	Assessor	Melakukan pengelolaan data dari sebuah sistem termasuk data proyek, data unadjusted actor weight, data fuzzy un-adjusted <i>Use Case</i> weight, data technical complexity, data environmental complexity

Tabel 4.2 Definisi *Use Case*

No.	<i>Use Case</i>	Deskripsi
1.	<i>Login</i>	<i>Use Case login</i> merupakan <i>Use Case</i> yang digunakan untuk mengelola data autentikasi pengguna yaitu assessor
2.	Mengelola Data Project	<i>Use Case</i> mengelola data project merupakan mengelola data proyek yang akan dibuat yang meliputi <i>add</i> , <i>view</i> dan cetak data proyek.
3.	Mengolah Unadjusted Actor Weight	<i>Use Case</i> mengolah unadjusted actor weight merupakan mengolah data unadjusted actor weight yang meliputi <i>add</i> , <i>view</i> dan cetak data unadjusted actor weight.
4.	Mengolah <i>Fuzzy unadjusted Use Case weight</i>	<i>Use Case</i> mengolah <i>fuzzy unadjusted Use Case weight</i> merupakan mengolah data <i>fuzzy unadjusted Use Case weight</i> yang meliputi <i>add</i> , <i>view</i> dan cetak data <i>fuzzy unadjusted Use Case weight</i> .
5.	Mengolah Technical Complexity Factors	<i>Use Case</i> mengolah technical complexity factors merupakan mengolah data technical complexity factors yang meliputi <i>add</i> , <i>view</i> dan cetak data technical complexity factors
6.	Mengolah Environmental Complexity Factors	<i>Use Case</i> mengolah environmental complexity factors merupakan mengolah data environmental complexity factors yang meliputi <i>add</i> , <i>view</i> dan cetak data environmental complexity factors
7.	<i>Logout</i>	<i>Use Case logout</i> merupakan <i>Use Case</i> yang digunakan untuk keluar dari sistem.

4.2.3 Skenario Sistem

Skenario sistem berfungsi untuk menjelaskan alur dari sebuah sistem serta alur alternatif yang dilakukan oleh para aktor yang menggunakan sistem ini. Skenario sistem sesuai dengan yang ada pada *Use case diagram* seperti pada Gambar 4.3.

1. Skenario Login

Skenario Masuk merupakan alur dari aksi aktor dan reaksi sistem jika akan masuk dalam sistem. Penjelasan urutan aksi aktor dan reaksi sistem pada skenario utama dan skenario alternatif skenario *use case* login ditunjukkan pada lampiran A.1.

2. Skenario Mengelola Data Project

Skenario mengelola data project merupakan alur aksi aktor dan reaksi sistem jika aktor Assessor akan mengelola data proyek pada menu dashboard. Penjelasan urutan aksi aktor dan reaksi sistem pada skenario utama dan skenario alternatif scenario *use case* mengelola project ditunjukkan pada lampiran A.2.

3. Skenario Mengolah *Unadjusted Actor Weight*

Skenario mengolah *unadjusted actor weight* merupakan alur aksi aktor dan reaksi sistem jika aktor Assessor akan mengolah unadjusted actor wight pada menu evaluation di submenu data unadjusted actor weight. Penjelasan skenario *use case* mengelola data jabatan ditunjukkan pada Tabel 4.3.

Tabel 4.3 Skenario Mengelola Anjusted Actor Weight

No. Use Case	UC03
Nama Use Case	Mengelola Unadjusted Actor Weight
Aktor	Assessor
Deskripsi Singkat	Assessor akan mengolah data anjusted actor weight
Prakondisi	- Assessor telah Login - Data Aktor telah diinputkan
Pascakondisi	Data Unadjusted actor weight berhasil di proses
Aliran normal	
Mengolah Data Anjusted Actor Weight	
Aksi Aktor	Reaksi Sistem
1. Klik menu evaluation	

Dilanjutkan

Lanjutan

2. Klik submenu anjusted actor weight	
	3. Menampilkan halaman data <i>Unajusted actor weight</i> meliputi form penilaian estimasi fuzzy beserta tombol submit juga tabel hasil estimasi anjusted actor weight beserta tombol print.
4. Memilih Project	
5. Mengisi form penilaian estimasi anjusted actor weight	
6. Klik tombol submit	
	7. Menyimpan data
	8. Menampilkan hasil estimasi anjusted actor weight pada tabel

4. Skenario Mengolah *Fuzzy unajusted Use Case weight*

Skenario mengolah *fuzzy unajusted Use Case weight* merupakan alur aksi aktor dan reaksi sistem jika aktor Assessor akan mengolah data *fuzzy unajusted Use Case weight* pada menu evaluation di submenu *fuzzy unajusted Use Case weight*. Penjelasan skenario *use case* mengolah *fuzzy unajusted Use Case weight* ditunjukkan pada Tabel 4.4.

Tabel 4.4 Skenario Mengelola *Fuzzy unajusted Use Case weight*

No. <i>Use Case</i>	UC04
Nama <i>Use Case</i>	Mengelola <i>Fuzzy unajusted Use Case weight</i>
Aktor	Assessor
Deskripsi Singkat	Assessor akan mengolah data <i>fuzzy unajusted Use Case weight</i>
Prakondisi	Data <i>Use Case</i> telah di inputkan

Dilanjutkan

Lanjutan

Pascakondisi	Data <i>fuzzy unadjusted Use Case weight</i> berhasil di proses
Aliran normal	
Mengolah Data <i>Fuzzy unadjusted Use Case weight</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu evaluation	
2. Klik submenu <i>fuzzy unadjusted Use Case weight</i>	
	3. Menampilkan halaman data <i>fuzzy unadjusted Use Case weight</i> meliputi form penilaian estimasi fuzzy beserta tombol submit juga tabel hasil estimasi fuzzy beserta tombol print.
4. Memilih Project	
5. Mengisi form penilaian estimasi fuzzy	
6. Klik tombol submit	
	7. Menyimpan data
	8. Menampilkan hasil estimasi fuzzy pada tabel

5. Skenario Mengolah Technical Complexity Factors

Skenario mengolah technical complexity factors merupakan alur aksi aktor Assessor akan mengolah data technical complexity factors pada menu evaluation di submenu technical complexity factors. Penjelasan cenario *use case* technical complexity factors ditunjukkan pada Tabel 4.5.

Tabel 4.5 Mengolah Technical Complexity Factor

No. <i>Use Case</i>	UC05
Nama <i>Use Case</i>	Mengelola Technical Comlexity Factor
Aktor	Assessor

Dilanjutkan

Lanjutan

Deskripsi Singkat	Assessor akan mengolah data environmental technical factor
Prakondisi	Assesor telah login
Pascakondisi	Data environmental technical factor berhasil di proses
Aliran normal	
Mengolah Data Technical Complexity Factor	
Aksi Aktor	Reaksi Sistem
1. Klik menu evaluation	
2. Klik submenu technical complexity	
	3. Menampilkan halaman data technical complexity meliputi form penilaian estimasi technical complexity beserta tombol submit juga tabel hasil estimasi technical complexity beserta tombol print.
4. Memilih Project	
5. Mengisi form penilaian estimasi technical complexity	
6. Klik tombol submit	
	7. Menyimpan data
	8. Menampilkan hasil estimasi environmental complexity pada tabel technical complexity

6. Skenario Mengolah Environmental Complexity Factors

Skenario mengolah environmental complexity factors merupakan alur aksi aktor Assessor akan mengolah data environmental complexity factors pada menu evaluation di submenu environmental complexity factors. Penjelasan skenario *use case* mengolah environmental complexity factors ditunjukkan pada Tabel 4.6.

Tabel 4.6 Mengolah Environmental Complexity Factor

No. Use Case	UC06
---------------------	-------------

Nama <i>Use Case</i>	Mengelola Environmental Complexity Factor
Aktor	Assessor
Deskripsi Singkat	Assessor akan mengolah data environmental complexity factor
Prakondisi	Assesor telah login
Pascakondisi	Data environmental complexity factor berhasil di proses
Aliran normal	
Mengolah Data Environmental Complexity Factor	
Aksi Aktor	Reaksi Sistem
1. Klik menu evaluation	
2. Klik submenu environmental complexity	
	3. Menampilkan halaman data environmental complexity meliputi form penilaian estimasi environmental complexity beserta tombol submit juga tabel hasil estimasi environmental complexity beserta tombol print.
4. Memilih Project	
5. Mengisi form penilaian estimasi environmental complexity	
6. Klik tombol submit	
	7. Menyimpan data
	8. Menampilkan hasil estimasi environmental complexity pada tabel

7. Skenario Logout

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario utama dan skenario alternatif skenario *use case* logout ditunjukkan pada lampiran A.3.

4.2.4 Sequence Diagram

Sequence Diagram pada sistem ini digunakan untuk menggambarkan interaksi yang terjadi antara obyek di dalam sistem yang disusun pada sebuah urutan

dan rangkaian waktu. *Sequence Diagram* diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

1. *Sequence Diagram* Login

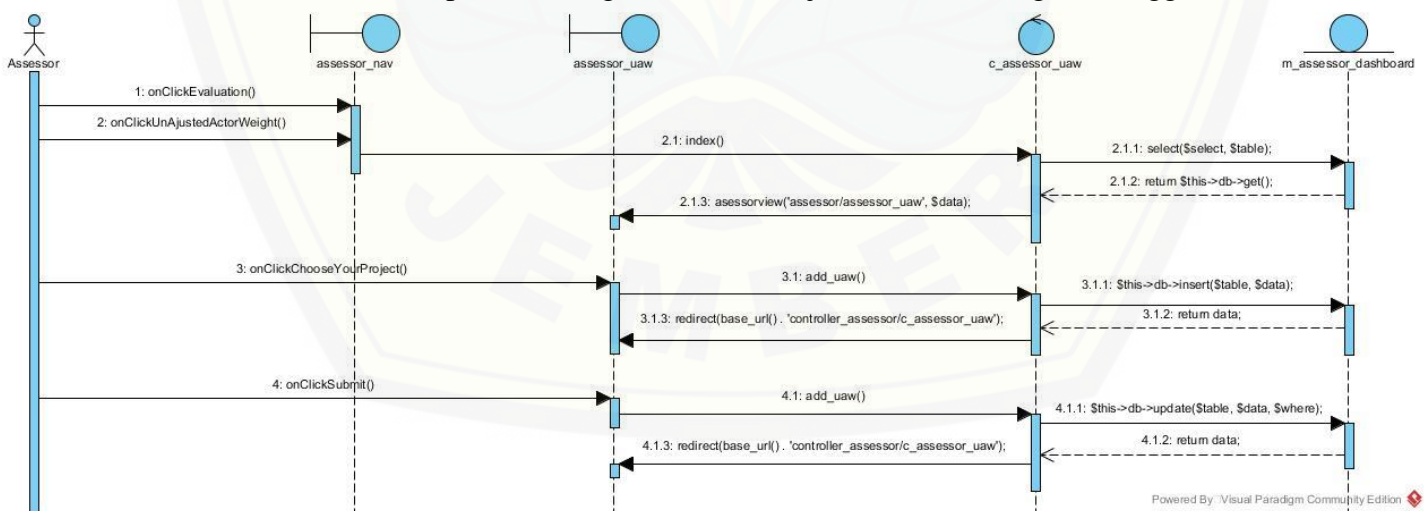
Sequence diagram masuk merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses login. Penggambaran *sequence diagram* masuk digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.1.

2. *Sequence Diagram* Mengelola Project

Sequence diagram mengelola data proyek merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses mengelola data proyek. Penggambaran *sequence diagram* mengelola data karyawan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.2.

3. *Sequence Diagram* Mengolah UnAjusted Actor Weight

Sequence diagram mengolah unadjusted actor weight merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses mengolah data unadjusted actor weight. Penggambaran



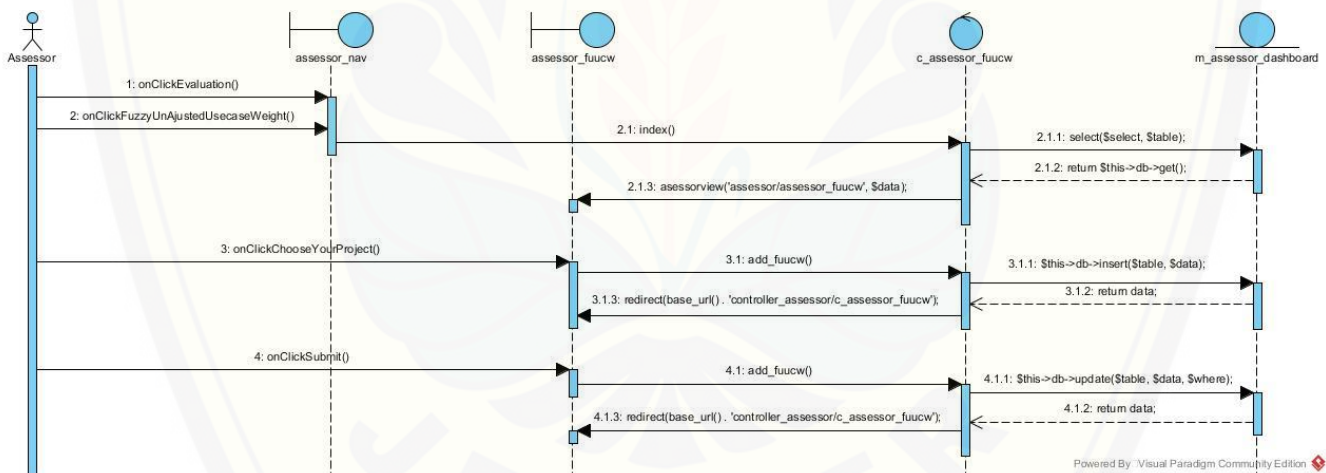
sequence diagram mengolah data unadjusted actor weight digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.4,

Gambar 4.4 *Sequence Diagram* Mengolah Unadjusted Actor Weight

Gambar 4.4 menggambarkan alur *method* dalam proses mengolah Unadjusted actor weight. Pada *sequence* ini terdapat class *view* *assessor_nav*, class *view* *assessor_uaw*, class *controller* *c_assessor_uaw*, class *model* *m_assessor_dashboard* serta di dalam class tersebut terdapat beberapa *method* yang dipanggil.

4. Sequence Diagram Mengolah Fuzzy unajusted Use Case weight

Sequence diagram mengolah *fuzzy unajusted Use Case weight* merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses mengolah data *fuzzy unajusted Use Case weight*. Penggambaran *sequence diagram* mengolah *fuzzy unajusted Use Case weight* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.5,

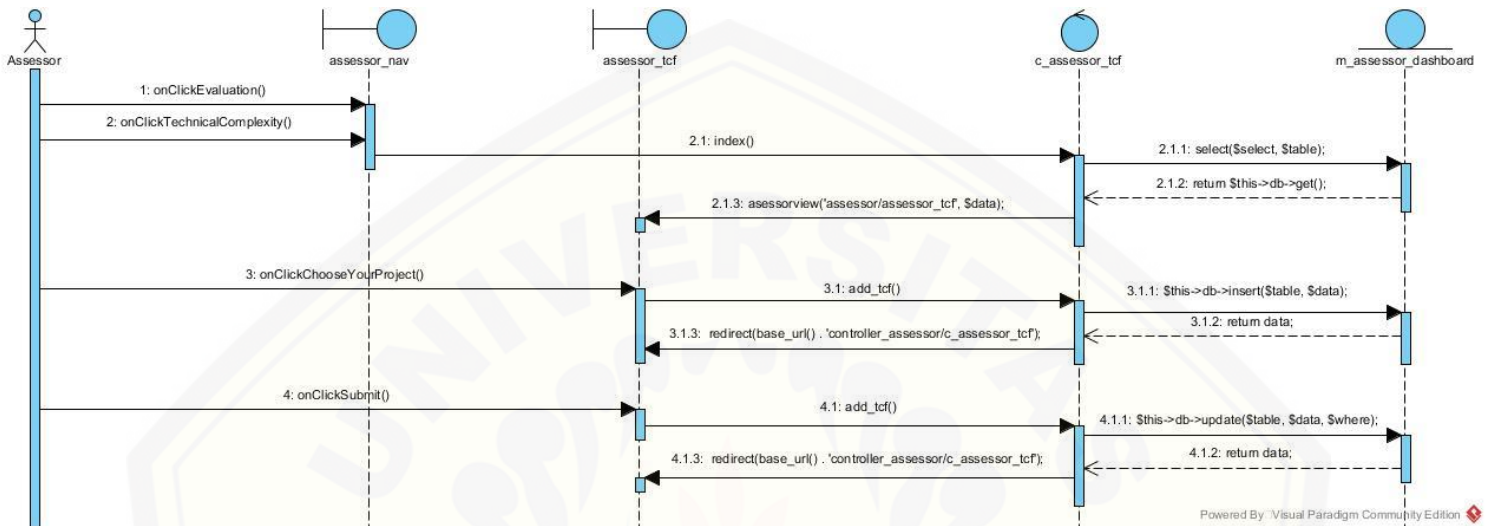


Gambar 4.5 Sequence Diagram Mengolah Fuzzy Unadjusted Use Case Weight

Gambar 4.5 menggambarkan alur *method* dalam proses mengolah fuzzy Unadjusted Use Case weight. Pada *sequence* ini terdapat class *view* *assessor_nav*, class *view* *assessor_fuuw*, class *controller* *c_assessor_fuuw*, class *model* *m_assessor_dashboard* serta di dalam class tersebut terdapat beberapa *method* yang dipanggil.

5. Sequence Diagram Mengolah Technical Complexity Factors

Penggambaran *sequence diagram* mengolah technical complexity factors digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.6,

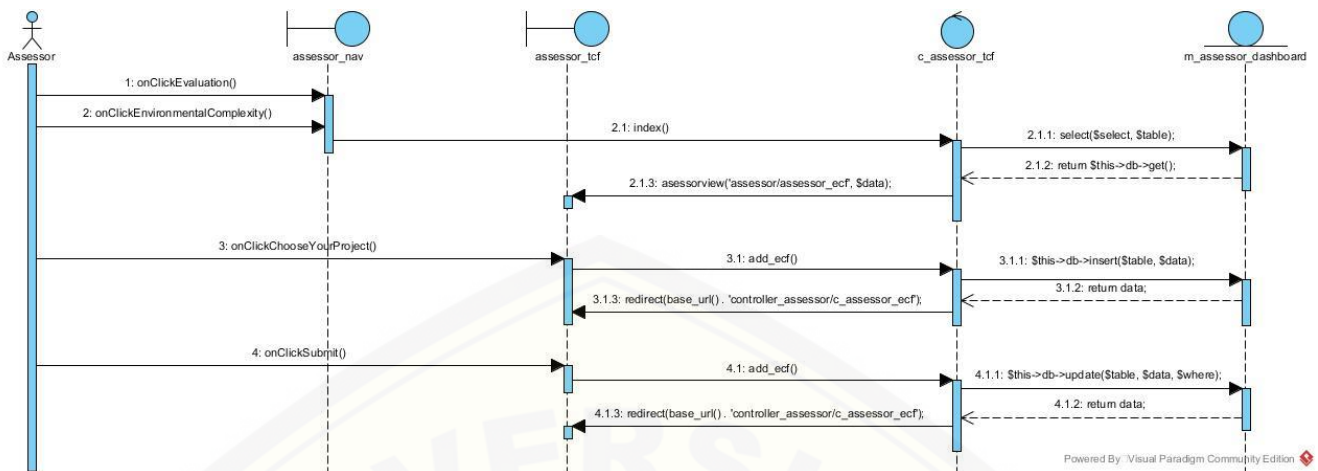


Gambar 4.6 *Sequence Diagram* Mengolah Technical Complexity

Gambar 4.6 menggambarkan alur *method* dalam proses mengolah technical complexity. Pada *sequence* ini terdapat class *view* *assessor_nav*, class *view* *assessor_tcf*, class *controller* *c_assessor_tcf*, class *model* *m_assessor_dashboard* serta di dalam class tersebut terdapat beberapa *method* yang dipanggil.

6. *Sequence Diagram* Environmental Complexity Factors

Penggambaran *sequence diagram* environmental complexity factors digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.7,



Gambar 4.7 *Sequence Diagram* Mengolah Environmental Complexity

Gambar 4.7 menggambarkan alur *method* dalam proses mengolah environmental complexity. Pada *sequence* ini terdapat class *view* `assessor_nav`, class *view* `assessor_ecf`, class *controller* `c_assessor_ecf`, class *model* `m_assessor_dashboard` serta di dalam class tersebut terdapat beberapa *method* yang dipanggil.

7. *Sequence Diagram* Logout

Penggambaran *sequence diagram* logout digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Lampiran B.3.

4.2.5 *Activity Diagram*

Activity diagram pada sistem ini berfungsi untuk menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana masing-masing alir berakhir.

1. *Activity diagram* Login

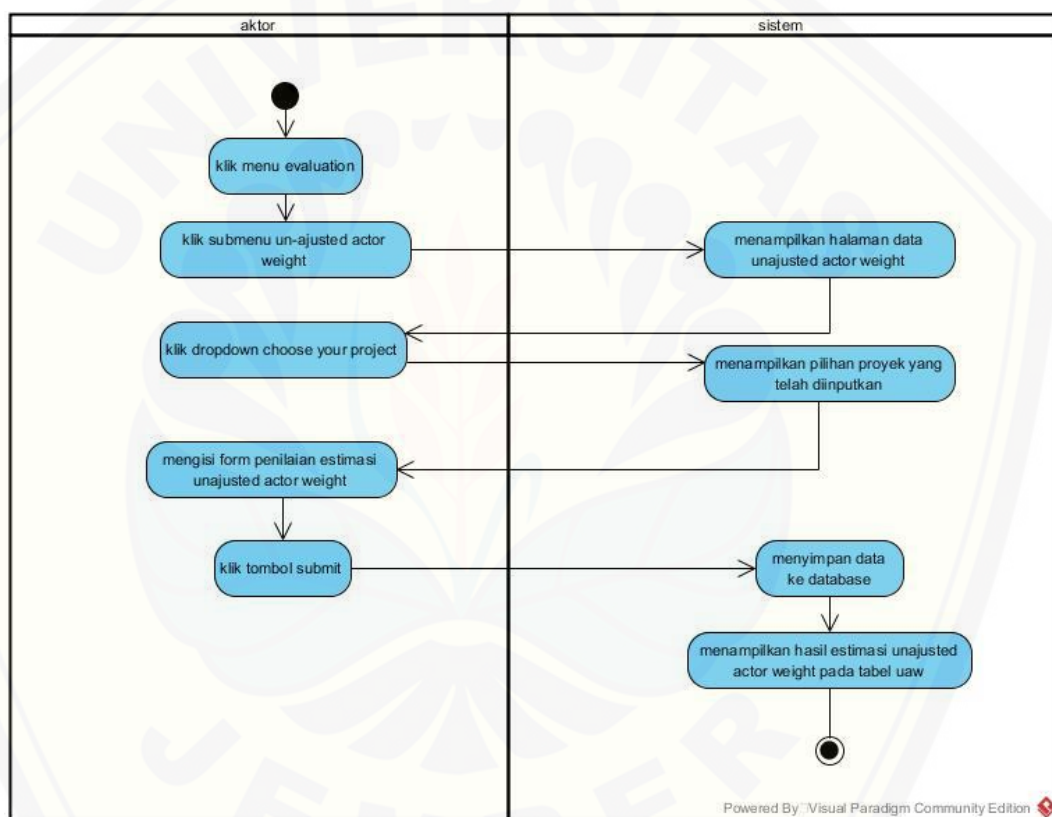
Activity diagram ini menggambarkan aktivitas login jika ingin mengakses sistem informasi ini dengan *email* dan *password* yang telah disediakan serta login sesuai hak akses yang ada seperti yang ditunjukkan pada lampiran C.1

2. Activity diagram Mengelola Project

Activity diagram ini menggambarkan alur aktivitas yang dilakukan oleh aktor dan sistem jika ingin mengakses dan mengelola project dimulai dari tambah, lihat dan cetak seperti yang ditunjukkan pada lampiran C.2

3. Activity diagram Mengolah UnAjusted Actor Weight

Activity diagram ini menggambarkan alur aktivitas yang dilakukan oleh aktor dan sistem jika ingin mengakses dan mengolah unadjusted actor weight dimulai dari tambah, lihat dan cetak seperti yang ditunjukkan pada Gambar 4.8,

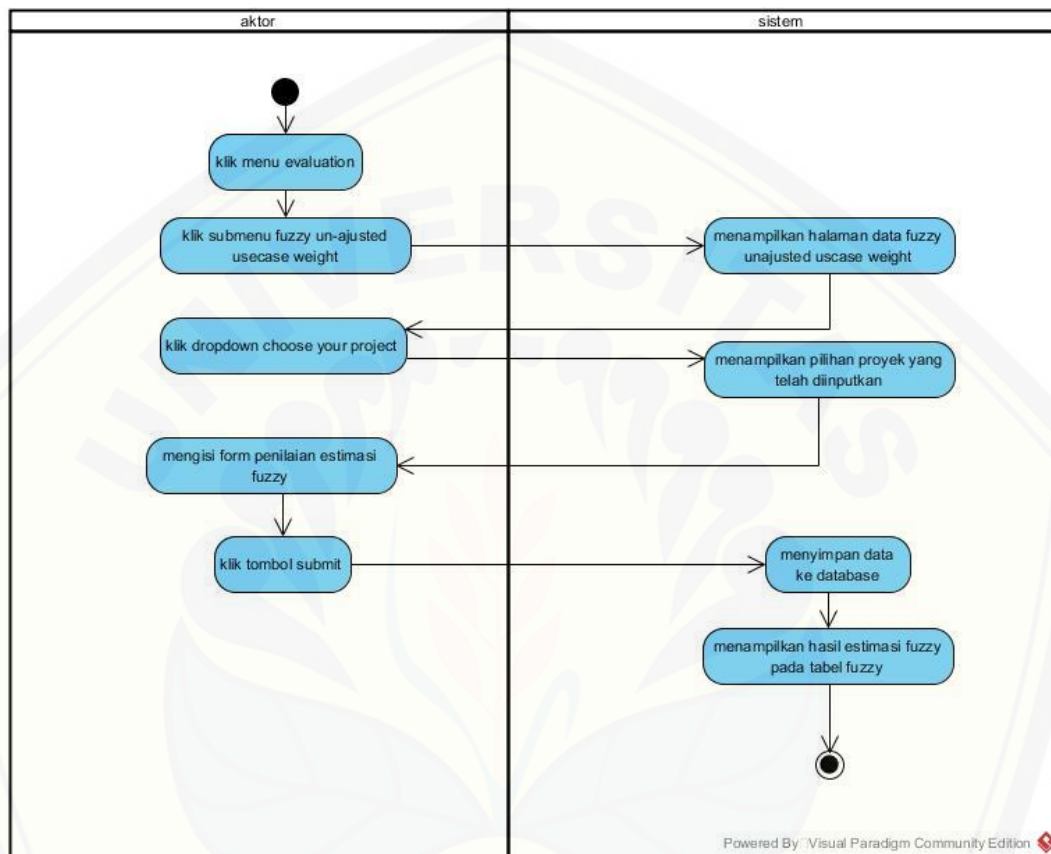


Gambar 4.8 Activity Diagram Mengolah Unadjusted Actor Weight

Gambar 4.8 menggambarkan activity diagram mengolah unadjusted actor weight dimana pada activity diagram ini menggambarkan aktivitas actor yaitu assessor jika ingin mengolah unadjusted actor weight untuk melengkapi langkah-langkah evaluasi guna mendapatkan hasil estimasi proyek pengembangan perangkat lunak.

4. Activity diagram Mengolah Fuzzy unadjusted Use Case weight

Activity diagram ini menggambarkan alur aktivitas yang dilakukan oleh aktor dan sistem jika ingin mengakses dan mengolah *fuzzy unadjusted Use Case weight* seperti yang ditunjukkan pada Gambar 4.9,

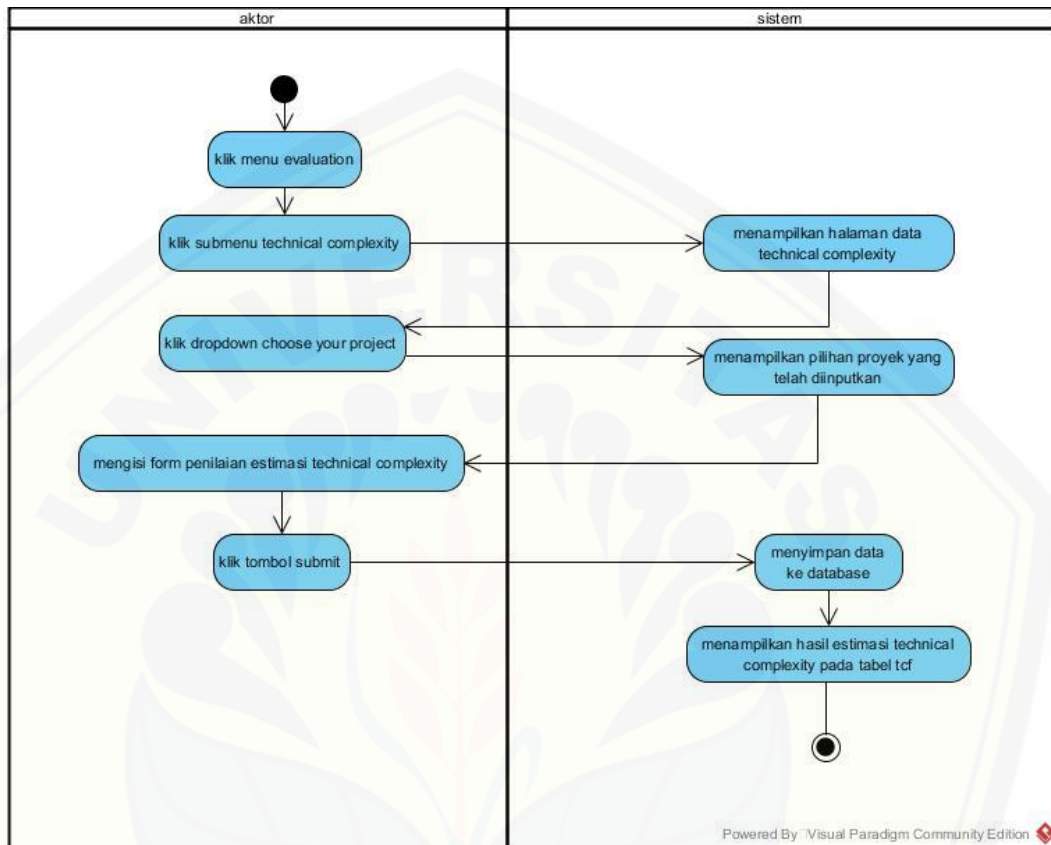


Gambar 4.9 Activity Diagram Mengolah Fuzzy Unadjusted Use Case Weight

Gambar 4.9 menggambarkan *activity diagram* mengolah fuzzy unadjusted Use Case weight dimana pada *activity diagram* ini menggambarkan aktivitas actor yaitu assessor jika ingin mengolah fuzzy unadjusted Use Case weight untuk melengkapi langkah-langkah evaluasi guna mendapatkan hasil estimasi proyek pengembangan perangkat lunak.

5. Activity diagram Mengolah Technical Complexity Factors

Activity diagram ini menggambarkan alur aktivitas yang dilakukan oleh aktor dan sistem jika ingin Mengolah Technical Complexity Factors dimulai dari tambah, lihat dan cetak seperti yang ditunjukkan pada Gambar 4.10,

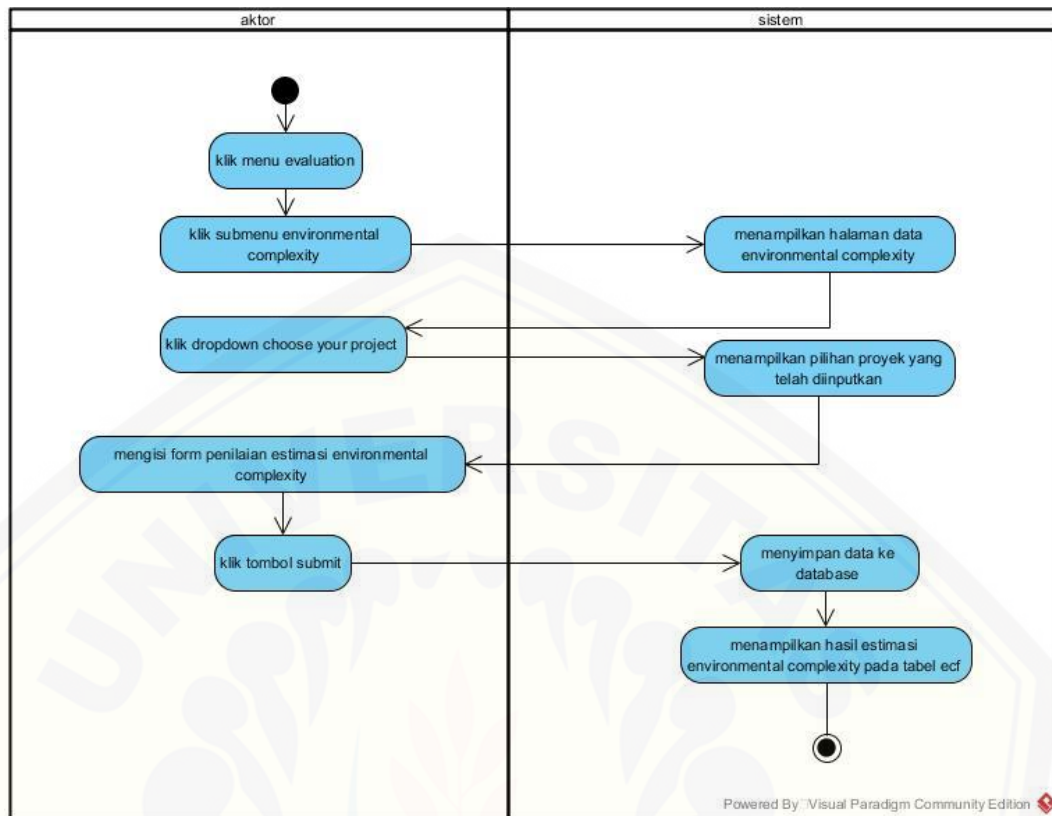


Gambar 4.10 *Activity Diagram* Mengolah Technical Complexity

Gambar 4.10 menggambarkan *activity diagram* mengolah technical complexity dimana pada *activity diagram* ini menggambarkan aktivitas actor yaitu assessor jika ingin mengolah technical complexity untuk melengkapi langkah-langkah evaluasi guna mendapatkan hasil estimasi proyek pengembangan perangkat lunak.

6. *Activity diagram* Mengolah Environmental Complexity Factors

Activity diagram ini menggambarkan alur aktivitas yang dilakukan oleh aktor dan sistem jika ingin mengolah Environmental Complexity Factors dimulai dari tambah, lihat dan cetak seperti yang ditunjukkan pada Gambar 4.11,



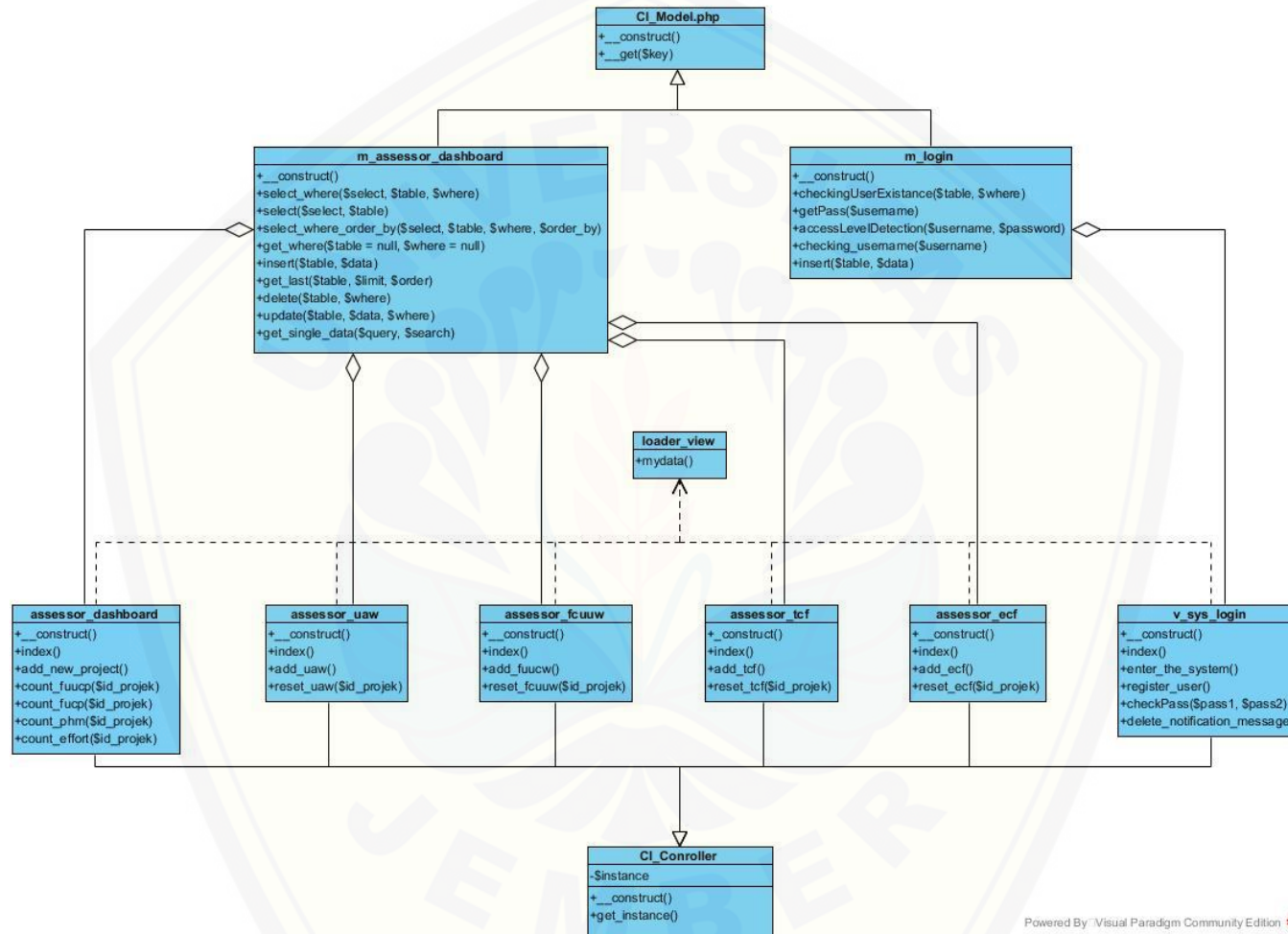
Gambar 4.11 *Activity Diagram* Mengolah Environmental Complexity

Gambar 4.11 menggambarkan *activity diagram* mengolah environmental complexity dimana pada *activity diagram* ini menggambarkan aktivitas actor yaitu assessor jika ingin mengolah environmental complexity untuk melengkapi langkah-langkah evaluasi guna mendapatkan hasil estimasi proyek pengembangan perangkat lunak.

7. *Activity diagram* Keluar

Activity diagram ini menggambarkan alur aktivitas yang dilakukan oleh aktor dan sistem jika ingin keluar dari sistem seperti yang ditunjukkan pada lampiran C.3.

4.2.6 Class Diagram



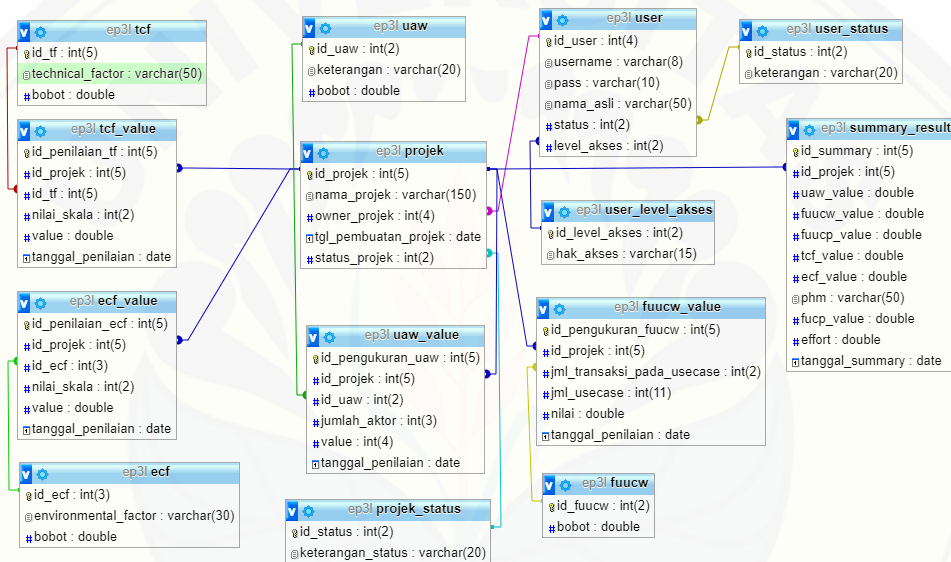
5

6 Gambar 4.12 Class Diagram

Gambar 4.12 menggambarkan tentang relasi antar class di dalam sistem. Relasi terjadi antar *controller*, *model*, dan *view*. Berdasarkan gambar tersebut bisa dipahami keterkaitan dan ketergantungan antar class di dalam sistem.

6.1.1 Entity Relationship Diagram

Entity Relationship Diagram (ERD) pada sistem pengambilan keputusan penempatan bidang kerja ini menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD aplikasi ditunjukkan pada gambar 4.13.



Gambar 4.13 *Entity Relationship Diagram*

Gambar 4.13 hanya terdapat relasi antar tabel *one to many*, misal pada tabel *projek* terdapat atribut *id_projek* yang menjadi *primary key* dan pada tabel *uaw_value* juga terdapat atribut *id_projek* yang menjadi *foreign key*.

4.3 Penulisan Kode Program

Setelah tahap perancangan selesai, tahap selanjutnya dalam penelitian ini yaitu tahap pengimplementasian desain perancangan ke dalam bahasa pemrograman. Bahasa pemrograman yang dipakai adalah bahasa pemrograman *PHP* (*Hypertext Preprocessor*) dan menggunakan database *MySQL*. Dalam optimasi pencocokan pada graf untuk pengambilan keputusan penempatan bidang kerja ini menggunakan

framework CodeIgniter untuk memudahkan dalam pengembangan dan penulisan *coding*. Pada tahap implementasi perancangan ini menjelaskan tentang fitur-fitur yang terdapat pada sistem. Fitur-fitur tersebut meliputi mengelola data proyek, mengolah data Unadjusted actor weight, mengolah data fuzzy unadjusted *Use Case* weight, mengolah data technical complexity dan mengolah data environmental complexity.

4.4 Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian *whitebox* terlebih dahulu kemudian akan dilanjutkan dengan pengujian *blackbox*.

4.4.1 Pengujian White Box

Pengujian *white box* pada sistem ini dengan cara menggambar diagram alir, menghitung kompleksitas siklomatiknya (CC), dan membuat tabel pengujian *test case*. Pengujian kode program pada proses pertama dimana Unadjusted actor weight diakumulasikan bersama dengan nilai fuzzy Unadjusted *Use Case* weight pada method *count_fuucp* yang ditunjukkan pada Gambar 4.14, gambar diagram alir ditunjukkan pada Gambar 4.15 dan tabel pengujian alur *test case* ditunjukkan pada Tabel 4.7.

```

75 function count_fuucp($id_projek) {
76     $uaw_value = $this->m_assessor_dashboard->get_single_data("SELECT uaw_value FROM summary_result WHERE id_projek = '$id_projek'", 'uaw_value');
77     $fuucw_value = $this->m_assessor_dashboard->get_single_data("SELECT fuucw_value FROM summary_result WHERE id_projek = '$id_projek'", 'fuucw_value');
78     $fuucp_value = $uaw_value + $fuucw_value;
79     $this->m_assessor_dashboard->update("summary_result", array('fuucp_value' => $fuucp_value), "id_projek = '$id_projek'");
80     $this->session->set_flashdata('message_table_success', 'Fuzzy Un-Ajusted Usecase Points Value Added to Database');
81     redirect(base_url() . 'controller_assessor/c_assessor_dashboard');
82 }

```

Gambar 4.14 Kode Program Proses Pertama pada method *count_fuucp*



Gambar 4.15 Diagram Alir method *count_fuucp*

Berdasarkan flowchart yang telah dibuat, maka perhitungan cyclometric complexity adalah sebagai berikut

$$\begin{aligned} \text{Cyclomatic Complexity (CC)} &= E - N + 2 \\ &= 2 - 3 + 2 = 1 \end{aligned}$$

Diketahui bahwa *cyclomatic complexity method* proses1 berjumlah 1. Dari hasil dapat diambil kesimpulan bahwa function `count_fuucp` secara umum merupakan method yang tidak rumit. Hal ini dikarenakan nilai CC kurang dari 10. Test case dari *method* `count_fuucp` dapat dilihat pada Tabel 4.7

Tabel 4.7 Pengujian *Testcase* Proses Fuzzy Unadjusted *Use Case Point*

<i>Test Case 1</i>	Menghitung unadjusted <i>Use Case Point</i> dengan menjumlahkan hasil perhitungan unadjusted actor weight dengan unadjusted <i>Use Case</i> weight
Target yang diharapkan	Menampilkan hasil perhitungan unadjusted <i>Use Case Point</i>
Hasil Pengujian	Benar
Path/Jalur	75 – 76 – 77 – 78 – 79 – 80 – 81 – 82

Pengujian selanjutnya adalah pengujian kode program pada proses kedua dimana hasil dari FUUCP dikalikan dengan perhitungan hasil technical complexity dan juga hasil perhitungan environmental complexity pada method `count_fucp` yang ditunjukkan pada Gambar 4.16, gambar diagram alir ditunjukkan pada Gambar 4.17 dan tabel pengujian alur *test case* ditunjukkan pada Tabel 4.8.

```

84 function count_fucp($id_projek) {
85     $fucp_value = $this->m_assessor_dashboard->get_single_data("SELECT fucp_value FROM summary_result WHERE id_projek = '$id_projek'", 'fucp_value');
86     $tcf_value = $this->m_assessor_dashboard->get_single_data("SELECT tcf_value FROM summary_result WHERE id_projek = '$id_projek'", 'tcf_value');
87     $ecf_value = $this->m_assessor_dashboard->get_single_data("SELECT ecf_value FROM summary_result WHERE id_projek = '$id_projek'", 'ecf_value');
88     $fucp_value = $fucp_value * $tcf_value * $ecf_value;
89     $this->m_assessor_dashboard->update("summary_result", array('fucp_value' => $fucp_value), "id_projek = '$id_projek'");
90     $this->session->set_flashdata('message_table_success', 'Fuzzy Usecase Points Value Added to Database');
91     redirect(base_url() . 'controller_assessor/c_assessor_dashboard');
92 }

```

Gambar 4.16 Kode Program Proses Kedua pada method `count_fucp`



Gambar 4.17 Diagram Alir Proses method `count_fucp`

Berdasarkan flowchart yang telah dibuat, maka perhitungan cyclometric complexity adalah sebagai berikut

$$\begin{aligned} \text{Cyclomatic Complexity (CC)} &= E - N + 2 \\ &= 2 - 3 + 2 \\ &= 1 \end{aligned}$$

Diketahui bahwa *cyclomatic complexity method count_fucp* berjumlah 1. Dari hasil dapat diambil kesimpulan bahwa *method count_fucp* secara umum merupakan method yang tidak rumit. Hal ini dikarenakan nilai CC kurang dari 10. Test case dari *method count_fucp* dapat dilihat pada Tabel 4.8

Tabel 4.8 Pengujian *Testcase* Proses Fuzzy *Use Case Point*

<i>Test Case 1</i>	Menghitung <i>Use Case Point</i> dengan mengalikan hasil perhitungan unadjusted <i>Use Case Point</i> dengan hasil perhitungan technical factor serta dengan environmental factor
Target yang diharapkan	Menampilkan hasil perhitungan <i>Use Case Point</i>
Hasil Pengujian	Benar
Path/Jalur	84 – 85 – 86 – 87 – 88 – 89 – 90 – 91 – 92

Pengujian selanjutnya adalah pengujian kode program pada *method count_phm* yang ditunjukkan pada Gambar 4.18, gambar diagram alir ditunjukkan pada Gambar 4.19 dan tabel pengujian alur *test case* ditunjukkan pada Tabel 4.9.

```

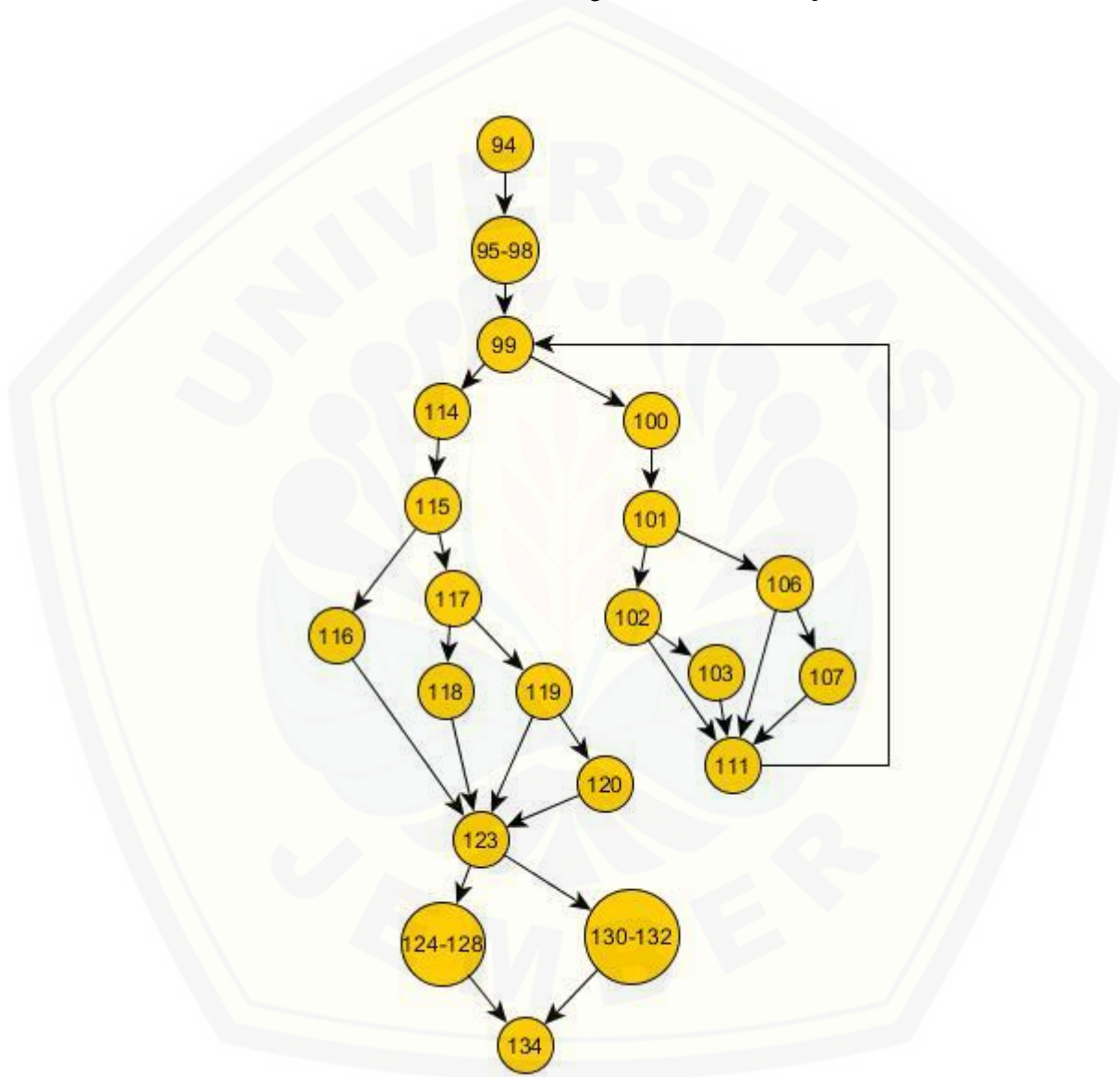
94 function count_phm($id_projek) {
95     $data['ecf'] = $this->m_assessor_dashboard->select_where_order_by("nilai_skala", "ecf_value", "id_projek = '$id_projek'", "id_ecf ASC");
96     $phm1 = 0;
97     $phm2 = 0;
98     $i = 1;
99     foreach ($data['ecf']->result_array() as $value) {
100         $nilai = $value['nilai_skala'];
101         if ($i < 7) {
102             if ($nilai <= 3) {
103                 $phm1++;
104             }
105             else {
106                 if ($nilai > 3) {
107                     $phm2++;
108                 }
109             }
110         }
111         $i++;
112     }
113
114     $phm_value = null;
115     if (($phm1 + $phm2) <= 2) {
116         $phm_value = "20";
117     } elseif (($phm1 + $phm2) == 3 || ($phm1 + $phm2) == 4) {
118         $phm_value = "28";
119     } elseif (($phm1 + $phm2) >= 4) {
120         $phm_value = "Please CANCEL project";
121     }

```

```

123 if ($phm_value == "Please CANCEL Project") {
124     $this->m_assessor_dashboard->update("projek", array('status_projek' => 2), "id_projek = '$id_projek'");
125     $this->db->set('tanggal_summary', 'NOW()', FALSE);
126     $this->m_assessor_dashboard->update("summary_result", array('phm' => $phm_value), "id_projek = '$id_projek'");
127     $this->session->set_flashdata('message_table_success', 'Person Hour Multiplier Value Added to Database');
128     redirect(base_url() . 'controller_assessor/c_assessor_dashboard');
129 } else {
130     $this->m_assessor_dashboard->update("summary_result", array('phm' => $phm_value), "id_projek = '$id_projek'");
131     $this->session->set_flashdata('message_table_success', 'Person Hour Multiplier Value Added to Database');
132     redirect(base_url() . 'controller_assessor/c_assessor_dashboard');
133 }
134 }
135

```

Gambar 4.18 Kode Program method *count_phm*Gambar 4.19 Diagram Alir method *count_phm*

Berdasarkan flowchart yang telah dibuat, maka perhitungan cyclometric complexity adalah sebagai berikut

$$\begin{aligned}
 \text{Cyclomatic Complexity (CC)} &= E - N + 2 \\
 &= 28 - 21 + 2 \\
 &= 9
 \end{aligned}$$

Diketahui bahwa *cyclomatic complexity method* proses1 berjumlah 3. Dari hasil dapat diambil kesimpulan bahwa *method count_phm* secara umum merupakan method rumit. Hal ini dikarenakan nilai CC sama dengan 10. Test case dari *method pindah* dapat dilihat pada Tabel 4.9

Tabel 4.9 Pengujian Testcase method *count_phm*

<i>Test Case 1</i>	Jika nilai F1 pada tabel environmental factor lebih kecil sama dengan 3 dan jumlah F1 ditambah F2 kurang dari sama dengan 2
Target yang diharapkan	Menampilkan hasil perhitungan <i>Use Case Point</i>
Hasil Pengujian	Benar
Path/Jalur	94 – 95 – 96 – 97 – 98 – 99 – 100 – 101 – 102 – 103– 111 – 99 – 114 – 115 – 116 – 123 – 130 – 131 – 132 – 134
<i>Test Case 2</i>	Jika nilai F1 pada tabel environmental factor lebih besar dengan 3 dan jumlah F1 ditambah F2 kurang dari sama dengan 2
Target yang diharapkan	Menampilkan proses akhir pencocokan
Hasil Pengujian	Benar
Path/Jalur	94 – 95 – 96 – 97 – 98 – 99 – 100 – 101 – 106 – 107– 111 – 99 – 114 – 115 – 116 – 123 – 130 – 131 – 132 – 134
<i>Test Case 3</i>	Jika nilai F1 pada tabel environmental factor lebih kecil sama dengan 3 dan jumlah F1 ditambah F2 lebih besar 2 dan jika jumlah F1 ditambah F2 sama dengan 3 atau 4
Target yang diharapkan	Menampilkan proses akhir pencocokan
Hasil Pengujian	Benar
Path/Jalur	94 – 95 – 96 – 97 – 98 – 99 – 100 – 101 – 102 – 103– 111 – 99 – 114 – 115 – 117 – 118 – 123 – 130 – 131 – 132 – 134
<i>Test Case 4</i>	Jika nilai F1 pada tabel environmental factor lebih kecil sama dengan 3 dan jumlah F1 ditambah F2 lebih besar 2 dan jika jumlah F1 ditambah F2 tidak sama dengan 3 atau 4 dan jika jumlah F1 ditambah F2 lebih besar sama dengan 4

Target yang diharapkan	Menampilkan proses akhir pencocokan
Hasil Pengujian	Benar
Path/Jalur	94 – 95 – 96 – 97 – 98 – 99 – 100 – 101 – 102 – 103– 111 – 99 – 114 – 115 – 117 – 119 – 120 – 123 – 124 – 125 – 126 – 127 – 128 – 134
<i>Test Case 5</i>	Jika nilai F1 pada tabel environmental factor lebih besar 3 dan jumlah F1 ditambah F2 lebih besar 2 dan jika jumlah F1 ditambah F2 sama dengan 3 atau 4
Target yang diharapkan	Menampilkan proses akhir pencocokan
Hasil Pengujian	Benar
Path/Jalur	94 – 95 – 96 – 97 – 98 – 99 – 100 – 101 – 106 – 107– 111 – 99 – 114 – 115 – 117 – 118 –123 – 130 – 131 – 132 – 134
<i>Test Case 6</i>	Jika nilai F1 pada tabel environmental factor lebih besar 3 dan jumlah F1 ditambah F2 lebih besar 2 dan jika jumlah F1 ditambah F2 tidak sama dengan 3 atau 4 dan jika jumlah F1 ditambah F2 tidak lebih besar sama dengan 4
Target yang diharapkan	Menampilkan proses akhir pencocokan
Hasil Pengujian	Benar
Path/Jalur	94 – 95 – 96 – 97 – 98 – 99 – 100 – 101 – 106 – 107– 111 – 99 – 114 – 115 – 117 – 119 – 120 – 123 – 130 – 131 – 132 – 134
<i>Test Case 7</i>	Jika nilai F1 pada tabel environmental factor lebih kecil sama dengan 3 dan jumlah F1 ditambah F2 lebih besar 2 dan jika jumlah F1 ditambah F2 tidak sama dengan 3 atau 4 dan jika jumlah F1 ditambah F2 tidak lebih besar sama dengan 4
Target yang diharapkan	Menampilkan proses akhir pencocokan
Hasil Pengujian	Benar
Path/Jalur	94 – 95 – 96 – 97 – 98 – 99 – 100 – 101 – 102 – 103– 111 – 99 – 114 – 115 – 117 – 119 –123 – 130 – 131 – 132 – 134
<i>Test Case 8</i>	Jika nilai F1 pada tabel environmental factor lebih kecil sama dengan 3 dan jumlah F1 ditambah F2 lebih besar 2 dan jika jumlah F1 ditambah F2 tidak sama dengan 3 atau 4 dan jika jumlah F1 ditambah F2 tidak lebih besar sama dengan 4
Target yang diharapkan	Menampilkan proses akhir pencocokan

Hasil Pengujian	Benar
Path/Jalur	94 – 95 – 96 – 97 – 98 – 99 – 100 – 101 – 102 – 103– 111 – 99 – 114 – 115 – 117 – 119 – 120 – 123 – 124 – 125 – 126 – 127 – 128 – 134
<i>Test Case 9</i>	Jika nilai phm sudah diketahui dan proyek dicancel
Target yang diharapkan	Menampilkan proses akhir pencocokan
Hasil Pengujian	Benar
Path/Jalur	94 – 95 – 96 – 97 – 98 – 99 – 100 – 101 – 102 – 103– 111 – 99 – 114 – 115 – 116 – 123 – 124 – 125 – 126 – 127 – 128 – 134

Pengujian selanjutnya adalah pengujian kode program pada proses environmental complexity pada *method* pindah yang ditunjukkan pada Gambar 4.20, gambar diagram alir ditunjukkan pada Gambar 4.21 dan tabel pengujian alur *test case* ditunjukkan pada Tabel 4.10

```

136 function count_effort($id_projek) {
137     $fucp = $this->m_assessor_dashboard->get_single_data("SELECT fucp_value FROM summary_result WHERE id_projek = '$id_projek'", 'fucp_value');
138     $phm_get_from_db = $this->m_assessor_dashboard->get_single_data("SELECT phm FROM summary_result WHERE id_projek = '$id_projek'", 'phm');
139     $phm = doubleval($phm_get_from_db);
140
141     $effort = $fucp * $phm;
142
143     $this->m_assessor_dashboard->update("projek", array('status_projek' => 2), "id_projek = '$id_projek'");
144
145     $this->db->set('tanggal_summary', 'NOW()', FALSE);
146     $this->m_assessor_dashboard->update("summary_result", array('effort' => $effort), "id_projek = '$id_projek'");
147     $this->session->set_flashdata('message_table_success', 'EFFORT Value Added to Database');
148     redirect(base_url() . 'controller_assessor/c_assessor_dashboard');
149 }
    
```

Gambar 4.10 Kode Program Proses Environmental Complexity



Gambar 4.21 Diagram Alir Proses Environmental Complexity

Berdasarkan flowchart yang telah dibuat, maka perhitungan cyclometric complexity adalah sebagai berikut

$$\begin{aligned}
 \text{Cyclomatic Complexity (CC)} &= E - N + 2 \\
 &= 9 - 8 + 2 \\
 &= 3
 \end{aligned}$$

Diketahui bahwa *cyclomatic complexity method* proses1 berjumlah 3. Dari hasil dapat diambil kesimpulan bahwa *method count_effort* secara umum merupakan method yang tidak rumit. Hal ini dikarenakan nilai CC kurang dari 10. Test case dari *method* pindah dapat dilihat pada Tabel 4.10

Tabel 4.10 Pengujian Testcase method count_effort

<i>Test Case 1</i>	Menghitung nilai effort dengan mengalikan hasil perhitungan unadjusted <i>Use Case Point</i> dengan hasil nilai person hour multiplier
Target yang diharapkan	Menampilkan hasil perhitungan effort
Hasil Pengujian	Benar
Path/Jalur	136 – 137 – 138 – 139 – 140 – 141 – 142 – 143 – 144 – 145 – 146 – 147 – 148 – 149

4.4.2 Pengujian *Black Box*

Pengujian *black box* berfungsi untuk menguji sistem dari segi spesifikasi fungsional sistem dengan tujuan mengetahui apakah fungsi-fungsi, inputan, dan keluaran sistem sesuai dengan spesifikasi yang dibutuhkan oleh pengguna. Hasil pengujian dengan metode *black box* dapat dilihat pada Tabel 4.11,

Tabel 4.11 Pengujian *Blackbox*

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Login	Mengisi kolom <i>email</i> dan <i>password</i> lalu klik tombol <i>log in</i>	Menampilkan dashboard <i>user</i>	[√] Benar [] Salah
		Kolom <i>username</i> dan <i>password</i> kosong	Menampilkan <i>span</i> “Harap isi bidang ini”	[√] Benar [] Salah
		Kolom <i>password</i> kosong	Menampilkan <i>span</i> “Harap isi bidang ini”	[√] Benar [] Salah
		Kolom <i>email</i> kosong	Menampilkan <i>span</i> “Harap isi bidang ini”	[√] Benar [] Salah
		<i>Email</i> atau <i>password</i> salah	Menampilkan <i>alert</i> “ <i>LogIn FAILED, WRONG username / password</i> ”	[√] Benar [] Salah
2.	Mengelola Data Project	Klik menu dashboard	Menampilkan halaman data proyek	[√] Benar [] Salah
		Mengisi tambah data proyek lalu klik submit	Menambahkan data ke database dan menampilkan kembali halaman data proyek	[√] Benar [] Salah

Dilanjutkan

Lanjutan

3.	Mengolah Unjusted Actor Weight	Klik menu evaluation	Menampilkan sub menu kriteria	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
		Klik sub menu unjusted actor weight	Menampilkan halaman unjusted actor weight	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
		Klik dropdown choose you project	Menampilkan pilihan proyek yang telah diinputkan	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
		Mengisi form penilaian dan klik tombol submit	Menyimpan data dan menampilkan halaman hasil estimasi	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
4.	Mengelola Fuzzy Unjusted Use Case Weight	Klik menu evaluation	Menampilkan sub menu kriteria	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
		Klik sub menu fuzzy unjusted Use Case weight	Menampilkan halaman fuzzy unjusted Use Case weight	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
		Klik dropdown choose you project	Menampilkan pilihan proyek yang telah diinputkan	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
		Mengisi form penilaian dan klik tombol submit	Menyimpan data dan menampilkan halaman hasil estimasi	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
5.	Mengolah Technical Complexity	Klik menu evaluation	Menampilkan sub menu kriteria	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah

Dilanjutkan

Lanjutan

		Klik sub menu technical complexity	Menampilkan halaman technical complexity	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
		Klik dropdown choose you project	Menampilkan pilihan proyek yang telah diinputkan	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
		Mengisi form penilaian dan klik tombol submit	Menyimpan data dan menampilkan halaman hasil estimasi	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
6.	Mengolah Environmental Complexity	Klik menu evaluation	Menampilkan sub menu kriteria	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
		Klik sub menu environmental complexity	Menampilkan halaman environmental complexity	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
		Klik dropdown choose you project	Menampilkan pilihan proyek yang telah diinputkan	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
		Mengisi form penilaian dan klik tombol submit	Menyimpan data dan menampilkan halaman hasil estimasi	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah
7.	Logout	Klik <i>Logout</i> pada dropdown pengguna	Menampilkan halaman utama sistem	<input checked="" type="checkbox"/> Benar <input type="checkbox"/> Salah

BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari hasil penelitian yang telah dilakukan oleh peneliti adalah sebagai berikut:

1. Menghitung estimasi proyek pengembangan perangkat lunak dengan menggunakan metode *fuzzy Use Case Point* memberikan kenaikan akurasi waktu bila di bandingkan dengan realita yang ada. Nilai estimasi yang didapatkan melalui *fuzzy Use Case Point* yaitu 1.338,5918 Man Per Hours, dengan mengacu pada Standar jam kerja wajib pekerja perminggu mengacu pada UU No. 13 th. 2003 tentang ketenagakerjaan (UU ketenagakerjaan) yaitu 40 jam kerja wajib per minggu atau 173 jam per bulan, berdasarkan wawancara yang di lakukan penulis pengembangan sistem ini dikerjakan 3 orang yang memiliki peranannya masing-masing yaitu project leader, analis system dan programmer sehingga didapatkan nilai estimasinya yaitu 1.338,5918 di bagi 3 menjadi 446.197 jam atau 2.6 bulan. Dibandingkan dengan realita pembuatan Proyek pengembangan perangkat lunak pada perustakaan daerah kabupaten jember dengan menggunakan 3 SDM, proyek tersebut dapat terselesaikan \pm 5 bulan. Sehingga dapat disimpulkan bahwa estimasi menggunakan *fuzzy Use Case Point* dapat memajemen waktu secara efektif dan efisien.
2. Implementasi metode *fuzzy Use Case Point* pada System estimasi proyek pengembangan ini Mampu mengestimasi waktu dan SDM yang digunakan dalam pembuatan sistem ini secara efektif dan efisien. Dalam mengestimasi sistem ini melalui beberapa tahap. Tahap pertama adalah mendapatkan nilai Fuzzy Unadjusted Use Case Point (FUUCP), untuk mendapatkan nilai UUCP maka diperlukan perhitungan terlebih dahulu nilai Unadjusted Actor Weight (

UAW) dan Unadjusted Use Case Weight (FUUCW). Tahap kedua adalah menentukan nilai Technical Complexity Factor (TCF) dan Environmental Complexity Factor (ECF) . Tahap ke Tiga menentukan nilai FUCP yang didapatkan berdasarkan perhitungan FUUCP , TCF dan ECF. Berdasarkan data pada ECF, dapat diketahui nilai PHM adalah 20 , Sehingga didapatkan Nilai Effort 1.338,5918 Man Per Hours. Sedangkan untuk perhitungan cost estimation tentang jam kerja wajib pekerja perminggu mengacu pada UU No. 13 th. 2003 tentang ketenagakerjaan (UU ketenagakerjaan) yaitu 40 jam kerja wajib per minggu atau 173 jam per bulan, berdasarkan wawancara yang dilakukan penulis Pengembangan sistem ini dikerjakan 3 orang yang memiliki jobs desk masing-masing yaitu project leader, analis system dan programmer sehingga didapatkan nilai estimasinya yaitu 1.338,5918 di bagi 3 menjadi 446.197 jam atau 2.6 bulan system ini dapat terselesaikan secara Efektif dan efisien.

3. Sistem estimasi proyek pengembangan perangkat dibangun dengan berbasis web dengan *user* yang memiliki hak akses yaitu assessor. Pengembangan sistem ini menggunakan *software development life cycle waterfall* yang tahapannya dimulai dari analisis kebutuhan, desain sistem, pengkodean sistem, pengujian sistem, dan pemeliharaan sistem. Analisis kebutuhan dilakukan untuk mendapatkan kebutuhan sistem baik fungsional maupun non fungsional. Analisis kebutuhan harus dilakukan secara mendetail. Desain sistem dilakukan dengan merancang diagram yang akan digunakan sebagai panduan dalam *coding* sistem. Tahap *coding* sistem dilakukan dengan menggunakan *framework* CodeIgniter yang merupakan *framework* dengan bahasa pemrograman PHP dan menggunakan MySQL sebagai *database management system*. Tahap pengujian yang dilakukan adalah *white box* dan *black box*.

6.2 Saran

1. Pengembangan lebih lanjut pada penelitian ini diharapkan dapat menambahkan fitur perhitungan anggaran keuangan berdasarkan effort yang telah dihasilkan sehingga akan lebih memudahkan perusahaan dalam mengestimasi rencana

anggaran yang dikeluarkan dalam pembuatan suatu system sehingga anggaran keuangan menjadi lebih efektif dan efisien.



DAFTAR PUSTAKA

- Aklani, S. A., & Pratama, A. (2014). Implementasi dan Perancangan Perpustakaan Online (Digital Library). 6.
- Arista, M., A. B., T., & Pantjawati, S. (2014). *AnalaisiS Perancangan System*.
- Asratian, A. S., Denley , T. M., & Häggkvist, R. (1998). *Bipartite Graphs and their Applications*. New York: Cambridge University Press.
- Bafadal, I. (2015). *Pengelolaan Perpustakaan Sekolah*. Jakarta: Bumi Aksara.
- Budi, D. S., Yoga Siswa, T. A., & Abijono, H. (2016). Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak. *Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak*.
- Bull Survey. (1998). Failure Causes.
- Dewi, R. S., Subriadi, A. P., & Sholiq. (2015). USE CASE POINT - ACTIVITY-BASED COSTING: METODE BARU UNTUK MENGESTIMASI BIAYA PENGEMBANGAN PERANGKAT LUNAK. 323.
- diana, T., samosir, D. H., & Widiyasa, I. M. (2008). PENGEMBANGAN DIGITAL LIBRARY PERPUSTAKAAN UNIVERSITAS ATMAJAYA YOGYAKARTA. *PENGEMBANGAN DIGITAL LIBRARY PERPUSTAKAAN UNIVERSITAS ATMAJAYA YOGYAKARTA*.
- Ermaya, S. K. (2015). APLIKASI INVENTARIS BUKU UNTUK PERPUSTAKAAN BERBASIS WEB.
- Galorath, D. D. (2016). The 10 Step Software Estimation Process For Successful Software Planning, Measurement and Control.
- Hariyanto, & wahono. (2015). Estimasi Proyek perangkat Lunak. *Estimasi Proyek perangkat Lunak*.

- Hariyanto, M. (2010). Estimasi Proyek Pengembangan Perangkat Lunak dengan Fuzzy Use Case Point. 10.
- Hartono, J. (1999). *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Offset.
- Indah Puji Hartatik, S. M. (2014). *Buku Praktis Mengembangkan SDM*. Yogyakarta: Laksana.
- Jacobson, I., Christorson, M., Jonsson, P., & Overgaard, G. (1992). *Object-Oriented Software Engineering. A Use Case Driven Approach (Revised Printing ed.)*. Addison - Wesley.
- Karner, G. (1993). *Resource Estimasi for objectory Projects, Objective System SF AB*. Torshamnsgatan.
- Kocay, W., & Kreher, D. L. (2004). *Graphs, Algorithms, and Optimization, Second Edition*. CRC Press.
- KPMG Canada. (1997). [http://www.it-cortex.com/Stat_Failure_Cause.htm#The%20KPMG%20Canada%20Survey%20\(1997\)](http://www.it-cortex.com/Stat_Failure_Cause.htm#The%20KPMG%20Canada%20Survey%20(1997)). Diambil kembali dari [http://www.it-cortex.com/Stat_Failure_Cause.htm#The%20KPMG%20Canada%20Survey%20\(1997\)](http://www.it-cortex.com/Stat_Failure_Cause.htm#The%20KPMG%20Canada%20Survey%20(1997)).
- Kurniawan, W., Sholiq, & Sutanto, T. (2013). PENENTUAN EFFORT RATE PADA ESTIMASI EFFORT MENGGUNAKAN METODE USE CASE POINT UNTUK PENGEMBANGAN PERANGKAT LUNAK WEBSITE KEPEMERINTAHAN. 61.
- Kusrini, & M, A. (2014). Sistem Pendukung Keputusan Evaluasi Kinerja Karyawan Untuk Promosi Jabatan.
- Much. Djunaidi, Andista, F. W., & Setiawan, E. (2005). PENENTUAN JUMLAH PRODUKSI DENGAN APLIKASI METODE FUZZY – MAMDANI.

PENENTUAN JUMLAH PRODUKSI DENGAN APLIKASI METODE FUZZY – MAMDANI.

Ni'matussholihah, A. (2015). GOOGLE DRIVE FOR STORING ARCHIVES.

Pressman, R. S. (2001). *Software Engineering a practitioner's approach* (5 ed.). New York, America: McGraw-Hill.

Rahman, A., Abrori, M., & Musafi, N. S. (2014). Penyelesaian Matching Graph Dengan Menggunakan Metode Hungarian Dan Penerapannya Pada Penempatan Karyawan Di Suatu Perusahaan.

samiran, F. w. (2017). Use Case Diagram. *Use Case Diagram*, 7.

Slamin. (2009). *DESAIN JARINGAN: Pendekatan Teori Graf*. Jember: Jember University Press.

Sprague, R. H., & Watson, H. J. (1993). *Decision Support System: Putting Theory Into Practice*. Englewood Cliffs: N.J. Prentice Hall.

Standish Group. (2011). chaos report.

Suharto, I. (1997). *Manajemen Proyek*. Jakarta: Erlangga.

Turban, E., & Aronson, J. E. (2001). *Decision Support System and Intelligence System. 6th edition*. Prentice Hall: Upper Saddle River.

Urva, G., & Siregar, H. F. (2015). Pemodelan UML E-Marketing Minyak Goreng.

wijaya, S. f. (2011). PENGARUH TEKNOLOGI INFORMASI DAN PERUBAHAN ORGANISASI. *PENGARUH TEKNOLOGI INFORMASI DAN PERUBAHAN ORGANISASI*.

Wijaya, S. F. (t.thn.). PENGARUH TEKNOLOGI INFORMASI DAN PERUBAHAN ORGANISASI. 2011.

Y. C, A. D. (2015). Kondisi Perpustakaan Umum Daerah Kabupaten Jember oleh Pelajar Berdasarkan Standar Nasional Perpustakaan. 11.

- Aklani, S. A., & Pratama, A. (2014). Implementasi dan Perancangan Perpustakaan Online (Digital Library). 6.
- Arista, M., A. B., T., & Pantjawati, S. (2014). *AnalaisiS Perancangan System*.
- Asratian, A. S., Denley , T. M., & Häggkvist, R. (1998). *Bipartite Graphs and their Applications*. New York: Cambridge University Press.
- Bafadal, I. (2015). *Pengelolaan Perpustakaan Sekolah*. Jakarta: Bumi Aksara.
- Budi, D. S., Yoga Siswa, T. A., & Abijono, H. (2016). Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak. *Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak*.
- Bull Survey. (1998). Failure Causes.
- Dewi, R. S., Subriadi, A. P., & Sholiq. (2015). USE CASE POINT - ACTIVITY-BASED COSTING: METODE BARU UNTUK MENGESTIMASI BIAYA PENGEMBANGAN PERANGKAT LUNAK. 323.
- diana, T., samosir, D. H., & Widiyasa, I. M. (2008). PENGEMBANGAN DIGITAL LIBRARY PERPUSTAKAAN UNIVERSITAS ATMAJAYA YOGYAKARTA. *PENGEMBANGAN DIGITAL LIBRARY PERPUSTAKAAN UNIVERSITAS ATMAJAYA YOGYAKARTA*.
- Ermaya, S. K. (2015). APLIKASI INVENTARIS BUKU UNTUK PERPUSTAKAAN BERBASIS WEB.
- Galorath, D. D. (2016). The 10 Step Software Estimation Process For Successful Software Planning, Measurement and Control.
- Hariyanto, & wahono. (2015). Estimasi Proyek perangkat Lunak. *Estimasi Proyek perangkat Lunak*.

- Hariyanto, M. (2010). Estimasi Proyek Pengembangan Perangkat Lunak dengan Fuzzy Use Case Point. 10.
- Hartono, J. (1999). *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Offset.
- Indah Puji Hartatik, S. M. (2014). *Buku Praktis Mengembangkan SDM*. Yogyakarta: Laksana.
- Jacobson, I., Christorson, M., Jonsson, P., & Overgaard, G. (1992). *Object-Oriented Software Engineering. A Use Case Driven Approach (Revised Printing ed.)*. Addison - Wesley.
- Karner, G. (1993). *Resource Estimasi for objectory Projects, Objective System SF AB*. Torshamnsgatan.
- Kocay, W., & Kreher, D. L. (2004). *Graphs, Algorithms, and Optimization, Second Edition*. CRC Press.
- KPMG Canada. (1997). [http://www.it-cortex.com/Stat_Failure_Cause.htm#The%20KPMG%20Canada%20Survey%20\(1997\)](http://www.it-cortex.com/Stat_Failure_Cause.htm#The%20KPMG%20Canada%20Survey%20(1997)). Retrieved from [http://www.it-cortex.com/Stat_Failure_Cause.htm#The%20KPMG%20Canada%20Survey%20\(1997\)](http://www.it-cortex.com/Stat_Failure_Cause.htm#The%20KPMG%20Canada%20Survey%20(1997)).
- Kurniawan, W., Sholiq, & Sutanto, T. (2013). PENENTUAN EFFORT RATE PADA ESTIMASI EFFORT MENGGUNAKAN METODE USE CASE POINT UNTUK PENGEMBANGAN PERANGKAT LUNAK WEBSITE KEPEMERINTAHAN. 61.
- Kusrini, & M, A. (2014). Sistem Pendukung Keputusan Evaluasi Kinerja Karyawan Untuk Promosi Jabatan.
- Much. Djunaidi, Andista, F. W., & Setiawan, E. (2005). PENENTUAN JUMLAH PRODUKSI DENGAN APLIKASI METODE FUZZY – MAMDANI.

*PENENTUAN JUMLAH PRODUKSI DENGAN APLIKASI METODE
FUZZY – MAMDANI.*

Ni'matussholihah, A. (2015). GOOGLE DRIVE FOR STORING ARCHIVES.

Pressman, R. S. (2001). *Software Engineering a practitioner's approach* (5 ed.).
New York, America: McGraw-Hill.

Rahman, A., Abrori, M., & Musafi, N. S. (2014). Penyelesaian Matching Graph
Dengan Menggunakan Metode Hungarian Dan Penerapannya Pada
Penempatan Karyawan Di Suatu Perusahaan.

samiran, F. w. (2017). Use Case Diagram. *Use Case Diagram*, 7.

Slamin. (2009). *DESAIN JARINGAN: Pendekatan Teori Graf*. Jember: Jember
University Press.

Sprague, R. H., & Watson, H. J. (1993). *Decision Support System: Putting
Theory Into Practice*. Englewood Cliffs: N.J. Prentice Hall.

Standish Group. (2011). chaos report.

Suharto, I. (1997). *Manajemen Proyek*. Jakarta: Erlangga.

Turban, E., & Aronson, J. E. (2001). *Decision Support System and Intelligence
System. 6th edition*. Prentice Hall: Upper Saddle River.

Urva, G., & Siregar, H. F. (2015). Pemodelan UML E-Marketing Minyak Goreng.

wijaya, S. f. (2011). PENGARUH TEKNOLOGI INFORMASI DAN
PERUBAHAN ORGANISASI. *PENGARUH TEKNOLOGI INFORMASI
DAN PERUBAHAN ORGANISASI*.

Wijaya, S. F. (n.d.). PENGARUH TEKNOLOGI INFORMASI DAN
PERUBAHAN ORGANISASI. 2011.

Y. C, A. D. (2015). Kondisi Perpustakaan Umum Daerah Kabupaten Jember oleh
Pelajar Berdasarkan Standar Nasional Perpustakaan. 11.

LAMPIRAN

LAMPIRAN A

A.1 Skenario Login

Tabel 1 Skenario Login

No. Usecase	UC01
Nama Usecase	Login
Aktor	Assessor
Deskripsi Singkat	Assessor akan mengakses sistem
Prakondisi	<i>Username</i> dan <i>password</i> yang akan digunakan untuk login
Pascakondisi	Berhasil <i>login</i>
Aliran normal	
Aksi Aktor	Reaksi Sistem
1. Masukkan email dan password	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>email</i> dan <i>password</i>
	4. Menampilkan halaman <i>dashboard</i> assessor
Alur alternatif	
Username dan Password Kosong	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>email</i> dan <i>password</i>
	4. Menampilkan <i>span</i> “Harap isi bidang ini”
	5. Menampilkan halaman <i>login</i> sistem
Alur alternatif	
Password Kosong	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>email</i> dan <i>password</i>

Dilanjutkan

Lanjutan

	4. Menampilkan span “ Harap isi bidang ini ”
	5. Menampilkan halaman <i>login</i> sistem
Alur alternatif Username Kosong	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>email</i> dan <i>password</i>
	4. Menampilkan <i>span</i> “Harap isi bidang ini”
	5. Menampilkan halaman <i>login</i> sistem
Alur alternative Username atau password salah	
2. Klik tombol <i>login</i>	
	3. Sistem mengecek <i>email</i> dan <i>password</i>
	4. Menampilkan <i>alert</i> “ <i>LogIn FAILED, WRONG username / password</i> ”
	5. Menampilkan halaman <i>login</i> sistem

A.2 Skenario Mengelola Data Proyek

Tabel 2 Skenario Mengelola Data Proyek

No. Usecase	UC02
Nama Usecase	Mengelola data proyek
Aktor	Assessor
Deskripsi Singkat	Admin akan mengelola data proyek
Prakondisi	Menu Dashboard
Pascakondisi	Data proyek berhasil ditambahkan
Aliran normal Tambah/input data proyek	
Aksi Aktor	Reaksi Sistem

Dilanjutkan

Lanjutan

1. Klik menu dashboard	
	2. Menampilkan halaman data proyek
3. Mengisi tambah data proyek	
4. Klik tombol submit	
	5. Menyimpan data ke database
	6. Menampilkan halaman data proyek beserta tabel list proyek
Alur Alternatif Data Ada Yang Kosong	
4. Klik tombol submit	
	5. Menampilkan span “harap isi bidang ini”
Aliran normal Lihat Hasil Akhir Estimasi Dari Tiap-tiap Langkah Evaluasi	
1. Klik menu dashboard	
	2. Menampilkan halaman data proyek meliputi tabel dengan atribut nomor, id proyek, nama proyek, tanggal dibuat, nilai unjusted actor weight, nilai fuzzy unjusted usecase weight, nilai technical complexity, nilai environmental complexity, PHM, effort, tanggal diselesaikan proyek tersebut.

A.3 Skenario Keluar

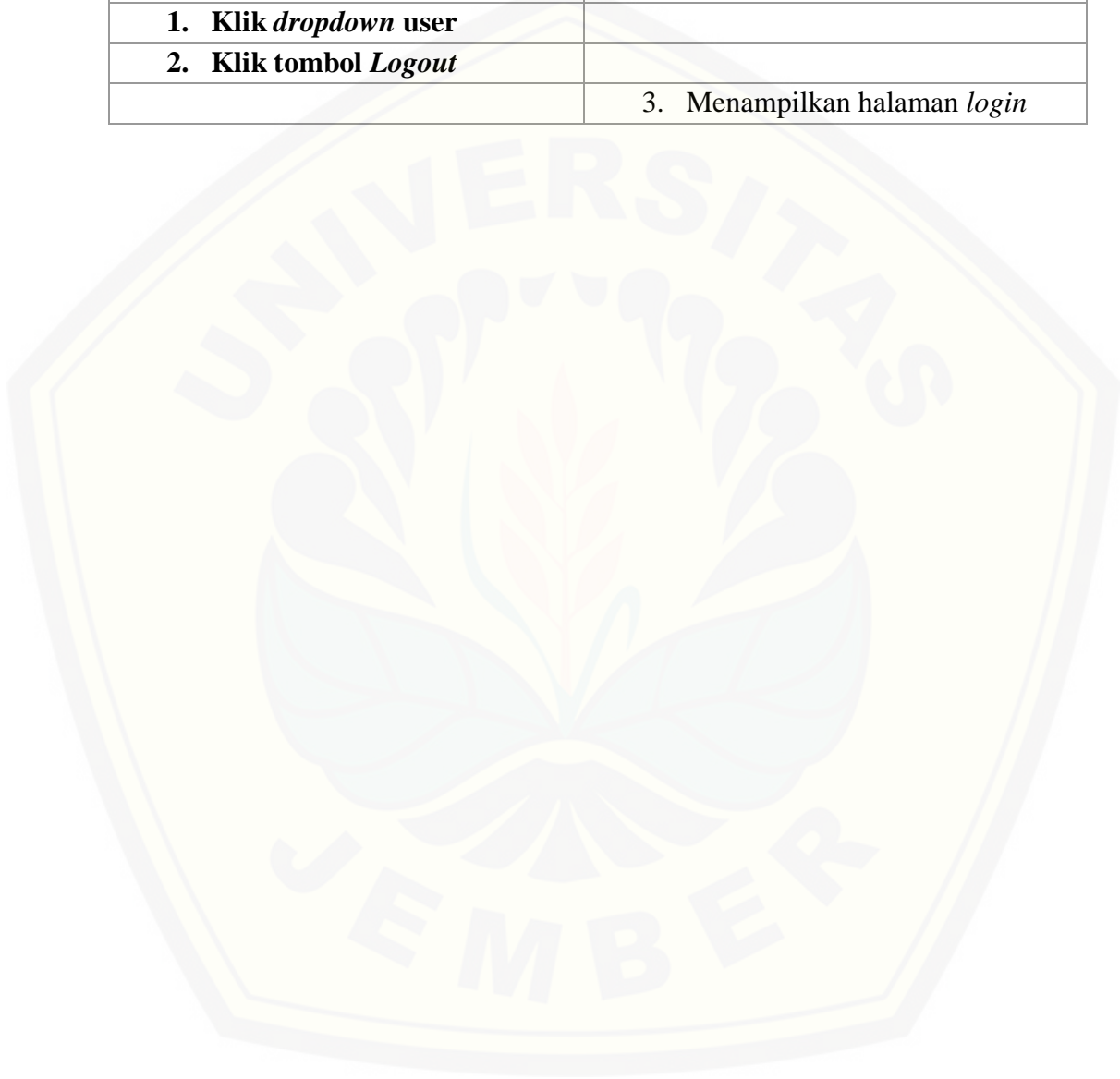
Tabel 3 Skenario Logout

No. Usecase	UC7
Nama Usecase	Logout
Aktor	Assessor
Deskripsi Singkat	Fitur untuk keluar dari sistem.
Prakondisi	Halaman Dashboard Assessor

Dilanjutkan

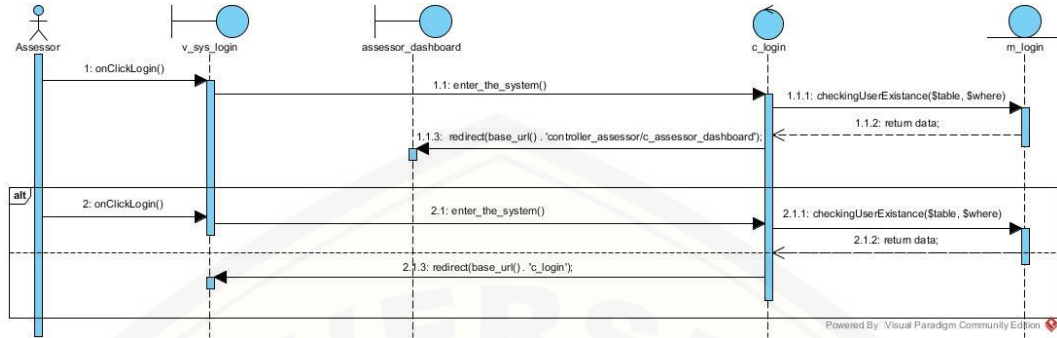
Lanjutan

Pascakondisi	Berhasil keluar dari sistem dan berada pada halaman <i>login</i>
Aliran normal Keluar	
Aksi Aktor	Reaksi Sistem
1. Klik <i>dropdown user</i>	
2. Klik tombol <i>Logout</i>	
	3. Menampilkan halaman <i>login</i>



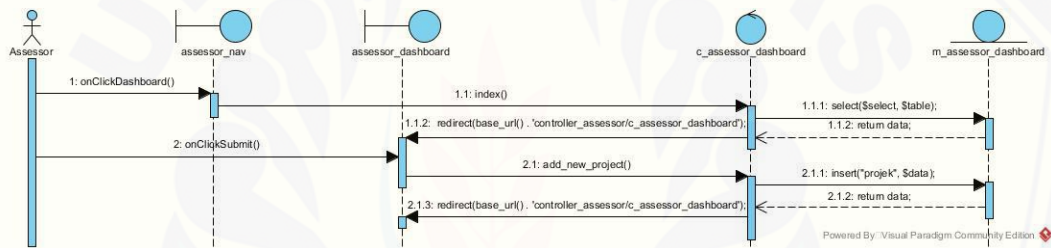
LAMPIRAN B

B.1 Sequence Diagram Login



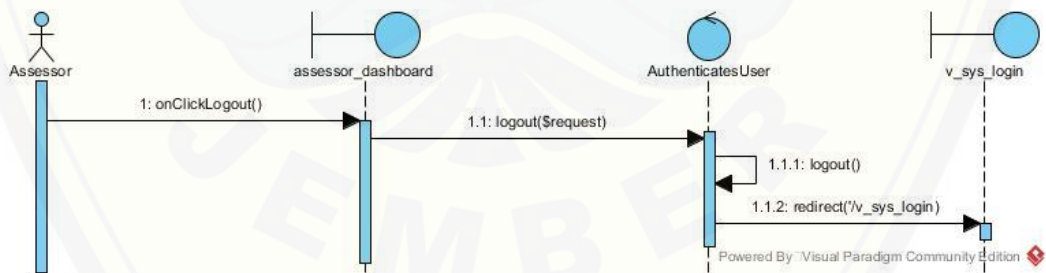
Gambar 1 Sequence Diagram Login

B.2 Sequence Diagram Mengelola Data Proyek



Gambar 2 Sequence Diagram Mengelola Data Proyek

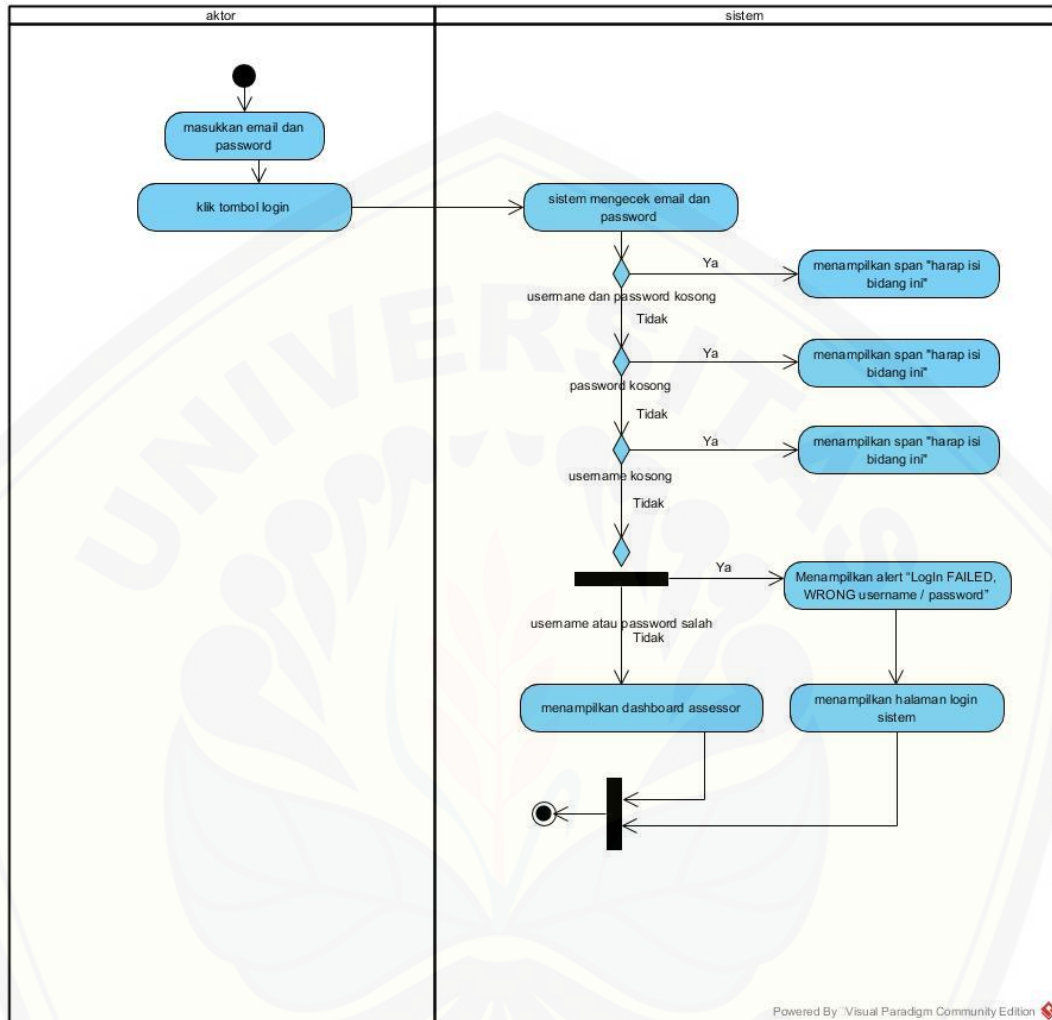
B.3 Sequence Diagram Logout



Gambar 3 Sequence Diagram Logout

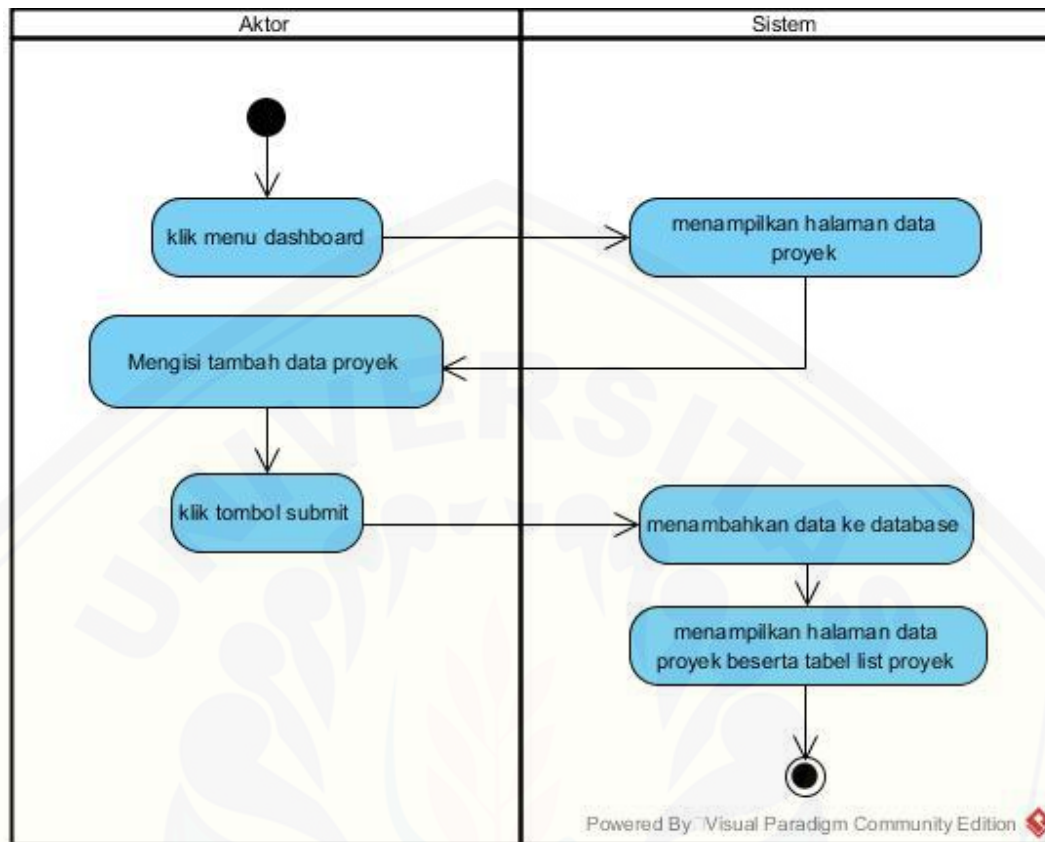
LAMPIRAN C

C.1 Activity Diagram Login



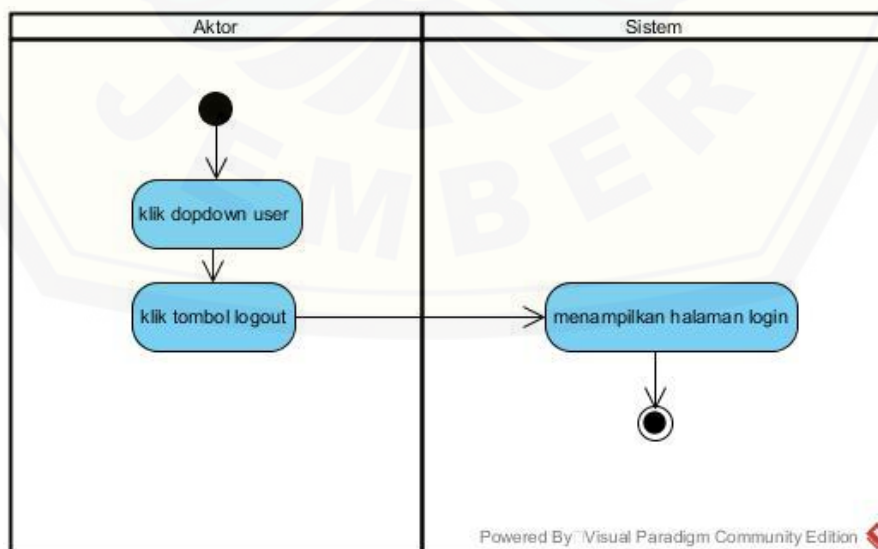
Gambar 1 Activity Diagram Masuk

C.2 Activity Diagram Mengelola Data Proyek



Gambar 4 Activity Diagram Mengelola Data Proyek

C.12 Activity Diagram Logout



Gambar 20 Activity Diagram Logout

