

repository.ub.ac.id

Implementasi Metode *COCOMO II* Untuk Estimasi Biaya Pengembangan Perangkat Lunak di CV. Profile Image Studio

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Amru Nizar Maqдум
NIM: 145150400111046



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

UNIVERSITAS
BRAWIJAYA

PENGESAHAN

Implementasi Metode *COCOMO II* Untuk Estimasi Biaya Pengembangan Perangkat Lunak di CV. Profile Image Studio

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :

Amru Nizar Maqdam

NIM: 145150400111046

Skripsi ini telah diuji dan dinyatakan lulus pada 16 Mei 2019

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II

Andi Reza P., S.Kom., M.MT.
NIK:2016078611281001

Widhy Hayuhardhika N.P., S.Kom., M.Kom.
NIK:2017128704092001

Mengetahui

Ketua Jurusan Sistem Informasi



Dr. Eng. Herman Tolle, S.T., M.T.
NIP:197408232000121001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 16 Mei 2019



Amru Nizar Magdum

NIM: 145150400111046

PRAKATA

Puji dan syukur penulis ucapkan kepada Tuhan Yesus Kristus yang memberikan penulis berkat dan kekuatan dalam menyelesaikan skripsi ini. Skripsi ini dapat terselesaikan tidak lepas dari dukungan dari banyak pihak yang telah menolong penulis. Oleh karena itu, pada kesempatan ini penulis hendak mengucapkan terimakasih kepada:

1. Bapak Andi Reza Perdanakusuma, S.Kom., M.MT, selaku Dosen Pembimbing I yang telah memberikan banyak waktu bimbingan maupun masukan selama pengerjaan skripsi agar penulis dapat menyelesaikan skripsi ini dengan sebaik-baiknya
2. Bapak Widhy Hayuhardhika Nugraha P., S.Kom., M.Kom. selaku Dosen Pembimbing II yang telah memberikan ide, masukan, serta waktu dan perhatiannya supaya penulis dapat menyelesaikan skripsi ini dengan sebaik-baiknya.
3. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB. selaku ketua program studi sistem informasi Universitas Brawijaya.
4. Bapak Ismiarta Akrunanda, S.T., M.Sc., Ph.D selaku sekretaris jurusan sistem informasi Universitas Brawijaya.
5. Bapak Herman Tolle, Dr. Eng., S.T, M.T selaku ketua jurusan sistem informasi Universitas Brawijaya.
6. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D Firdaus selaku dekan Fakultas Ilmu Komputer Universitas Brawijaya.
7. Orangtua dan keluarga tercinta penulis yang selalu memberikan doa serta motivasi.
8. Pihak CV. Profile Image Studio yang telah membantu mendukung penulis menyelesaikan skripsi ini.

Demikianlah yang penulis dapat sampaikan. Dia akhir kata penulis mengucapkan syukur atas selesainya skripsi ini, dan penulis berterimakasih kepada semua pihak yang telah menolong penulis, baik dosen pembimbing dan sahabat yang tidak dapat penulis sebutkan satu per satu. Semoga Tuhan memberkati kita semua.

Malang, 16 Mei 2019

Penulis

amrumaqdum@gmail.com

ABSTRAK

Amru Nizar Maqdam, Implementasi Metode *COCOMO II* Untuk Estimasi Biaya Pengembangan Perangkat Lunak di CV. Profile Image Studio

Pembimbing: Andi Reza Perdanakusuma, S.Kom., M.MT. dan Widhy Hayuhardhika Nugraha P., S.Kom., M.Kom.

Estimasi biaya perangkat lunak memegang peranan yang penting dalam hal pembuatan anggaran untuk proyek perangkat lunak. Dalam hal ini penentuan estimasi harga perangkat lunak di CV. Profile Image Studio belum mempunyai metode estimasi yang spesifik, sehingga sering menjadikan perusahaan mengalami kerugian secara finansial maupun waktu. Berangkat dari hal tersebut maka dalam penelitian ini akan mengimplementasikan metode *COCOMO II (Constructive Cost Model)* untuk menghitung biaya perangkat lunak sebagai saran kepada CV. Profile Image Studio. Estimasi biaya dapat diperoleh setelah mendapatkan nilai estimasi effort pengembangan sistem, sumber daya manusia dan juga estimasi waktu pengembangan sistem. Metode *COCOMO II* diimplementasikan kepada 2 sistem yang sudah selesai dikembangkan oleh CV. Profile Image Studio. Pada akhir penelitian ini, dilakukan perbandingan biaya pengembangan sistem, yaitu dengan membandingkan hasil estimasi biaya yang diperoleh menggunakan metode *COCOMO II (Constructive Cost Model)* dengan alokasi biaya yang dianggarkan oleh CV. Profile Image Studio. Adapun hasil dari implementasi metode *COCOMO II* dalam menghitung biaya perangkat lunak didapatkan bahwa total estimasi biaya sistem DBA ticketing sebesar Rp.61.344.000 yang dikerjakan oleh 3 orang dalam waktu 9 bulan sedangkan untuk sistem pintu air sebesar Rp.36.352.000 yang dikerjakan oleh 2 orang dalam waktu 8 bulan. Selisih estimasi biaya untuk seluruh perangkat lunak adalah sebesar 37,24% dengan selisih waktu sebesar 52,94% dan selisih SDM sebesar 40%. Sehingga metode *COCOMO II* akan lebih tepat dalam memperkirakan estimasi biaya pada suatu proyek yang pada akhirnya dapat meminimalisir kerugian yang diterima oleh perusahaan.

Kata kunci: estimasi biaya, estimasi biaya perangkat lunak, *COCOMO II*.

ABSTRACT

Amru Nizar Maqdam, Implementation *COCOMO II* Method For Development Cost Estimation In CV. Profile Image Studio

Supervisor: Andi Reza Perdanakusuma, S.Kom., M.MT. and Widhy Hayuhardhika Nugraha P., S.Kom., M.Kom.

Software cost estimation is an important aspect of an software project for budgeting. In the process of determining the price of a system in CV. The Profile Image Studio does not have specific estimation. because of that make often the company receive losses both financially and time. Based on this case, the cost estimation in this study will implement *COCOMO II (Constructive Cost Mode III)* method is implemented in calculating software costs suggestion for CV. Profile Image Studio. Cost estimates can be obtained after obtaining estimated effort, number of human resources and estimated system development time. *COCOMO II (Constructive Cost Model)* method is implemented in 2 systems that have been developed by CV. Studio Profile Image. At the end of this study, the cost comparison of system development was done by comparing estimated cost results obtained using the *COCOMO II (Constructive Cost Model)* with the allocation of costs by CV. Studio Profile Image. The results of applying the *COCOMO II (Constructive Cost Model II)* method in calculating software costs, it was found that the total estimated cost of the DBA ticket system was Rp.61.344.000 carried out by 3 people in 9 month while pintu air system was Rp.36.352.000 carried out by 2 people in 8 month. Difference in estimated cost for all software is 37,24% with time difference is 52,94 and people difference is 40%. So the *COCOMO II* method will be more precise in estimating cost of a project that can be minimize the losses received by the company.

Keywords: *cost estimation, software cost estimation, COCOMO II.*

DAFTAR ISI

PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN ORISINALITAS.....	iii
prakata	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN.....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Visi dan Misi perusahaan.....	5
2.1.1 Visi.....	5
2.1.2 Misi.....	5
2.2 Pengertian Proyek.....	5
2.3 Pengertian Manajemen Proyek Teknologi Informasi	6
2.4 Estimasi biaya perangkat lunak	6
2.5 Metode COCOMO II	7
2.6 <i>Data Flow Diagram</i>	7
2.7 <i>Function Point</i>	8
2.8 <i>Scale factor</i>	11
2.9 <i>Effort Multiplier</i>	13
2.10 <i>Effort Estimation</i>	18
2.11 Estimasi Biaya	18



2.12 Agile Software Development	19
BAB 3 METODOLOGI PENELITIAN	20
3.1 Sistematika Penelitian	20
3.2 Identifikasi Masalah	20
3.3 Studi Literatur	21
3.4 Pengumpulan Data.....	21
3.4.1 Wawancara.....	21
3.4.2 Observasi data	21
3.5 Analisis DFD (<i>Data Flow Diagram</i>)	21
3.6 Menghitung UFP (<i>Unadjusted Function Point</i>).....	22
3.7 Menghitung <i>Size</i>	23
3.8 Menghitung <i>Scale factor</i>	23
3.9 Menghitung <i>Effort multipliers</i>	23
3.10 Menghitung Effort Estimation	23
3.11 Analisis Hasil.....	24
3.12 Kesimpulan Saran.....	24
BAB 4 HASIL Data	25
4.1 Hasil Wawancara.....	25
4.2 Hasil lembar penilaian faktor eksponen (<i>Scale factor</i>)	25
4.3 Hasil lembar penilaian <i>Effort multipliers</i>	28
4.4 <i>Data Flow Diagram</i> (DFD).....	32
4.4.1 DFD level 0 Perangkat Lunak <i>DBA Ticketing</i>	32
4.4.2 DFD level 1 Perangkat Lunak <i>DBA Ticketing</i>	33
4.4.3 DFD level 2 Mengelola Pemesanan <i>DBA Ticketing</i>	34
4.4.4 DFD level 2 mengelola laporan keuangan <i>DBA Ticketing</i>	34
4.4.5 DFD level 2 mengelola agen dan deposit <i>DBA Ticketing</i>	35
4.4.6 DFD level 2 pembuatan tiket <i>DBA Ticketing</i>	36
4.4.7 DFD level 0 Perangkat Lunak Sistem informasi Pintu Air.....	36
4.4.8 DFD level 1 Perangkat Lunak Sistem informasi Pintu Air.....	37
4.4.9 DFD level 2 Mengelola pintu air	38
4.4.10 DFD level 2 mengelola laporan keuangan.....	38
4.4.11 DFD level 2 mengelola profil	39



4.4.12 DFD level 2 mengelola berita	40
4.4.13 DFD level 2 mengelola pengaduan	40
BAB 5 PEMBAHASAN.....	42
5.1 Skala Proyek.....	42
5.2 Analisis Function Point	42
5.2.1 Analisis Function Point Perangkat Lunak <i>DBA Ticketing</i>	43
5.2.2 Analisis Function Point Perangkat Lunak Sistem Pintu Air	45
5.3 Analisis Perhitungan UFP.....	48
5.3.1 Analisis Perhitungan UFP Perangkat Lunak <i>DBA Ticketing</i>	48
5.3.2 Analisis Perhitungan UFP Perangkat Lunak Sistem Informasi Pintu Air.....	57
5.4 Analisis perhitungan Faktor Skala	66
5.4.1 Perhitungan Faktor Skala Perangkat Lunak <i>DBA Ticketing</i>	66
5.4.2 Perhitungan Faktor Skala Perangkat Lunak Sistem Pintu Air	67
5.5 Analisis Perhitungan <i>Effort estimation</i>	67
5.5.1 Analisis Perhitungan <i>Effort Estimation</i> Perangkat Lunak <i>DBA Ticketing</i>	68
5.5.2 Analisis perhitungan <i>Effort Estimation</i> Perangkat lunak Sistem Pintu Air	69
5.6 Menghitung Estimasi Biaya Proyek.....	70
5.6.1 Analisis perhitungan Estimasi Biaya Perangkat lunak <i>DBA Ticketing</i>	71
5.6.2 Analisis Perhitungan Estimasi Biaya Perangkat lunak Sistem Pintu Air	71
5.7 Pembahasan Estimasi waktu, SDM, dan Biaya	72
5.7.1 Estimasi Sumberdaya Manusia (SDM)	72
5.7.2 Estimasi Waktu	73
5.7.3 Estimasi Biaya	73
BAB 6 Penutup.....	74
6.1 Kesimpulan	74
6.2 Saran.....	75
Daftar Referensi.....	76



LAMPIRAN A Hasil Wawancara	78
LAMPIRAN B Lembar Validasi PDM dan Data flow diagram	80
LAMPIRAN C Lembar penilaian scale factor	81
LAMPIRAN D effort multiplier	83



DAFTAR TABEL

Tabel 2. 1 Data Flow Diagram.....	7
Tabel 2. 2 komponen function point	8
Tabel 2. 3 Klasifikasi tipe data	9
Tabel 2. 4 Klasifikasi Bobot kompleksitas	10
Tabel 2. 5 QSM language Table	10
Tabel 2. 6 Bobot scale factor	13
Tabel 2. 7 Deskripsi Effort Multiplier	13
Tabel 2. 8 Tabel nilai <i>effort multipliers</i>	16
Tabel 4. 1 Scale factors Perangkat Lunak DBA ticketing.....	26
Tabel 4. 2 Scale Factors Perangkat Lunak Sistem Informasi Pintu Air.....	27
Tabel 4. 3 Effort multipliers Proyek Perangkat lunak DBA Ticketing.....	28
Tabel 4. 4 <i>Effort Multipliers</i> Proyek Perangkat Lunak Sistem Informasi Pintu Air	30
Tabel 5. 2 Analisis Function Point Perangkat Lunak DBA Ticketing proses mengelola pembelian	43
Tabel 5. 3 Analisis Function Point Perangkat Lunak DBA Ticketing proses mengelola laporan keuangan.....	43
Tabel 5. 4 Analisis Function Point Perangkat Lunak DBA Ticketing proses mengelola agen dan deposit	44
Tabel 5. 5 Analisis Function Point Perangkat Lunak DBA Ticketing proses mengelola pembuatan tiket.....	45
Tabel 5. 6 Analisis Function Point Perangkat Lunak Sistem Pintu Air proses pengelolaan data air	45
Tabel 5. 7 Analisis Function Point Perangkat Lunak Sistem Pintu Air proses pengolahan laporan keuangan.....	46
Tabel 5. 8 Analisis Function Point Perangkat Lunak Sistem Pintu Air proses pengolahan profil.....	46
Tabel 5. 9 Analisis Function Point Perangkat Lunak Sistem Pintu Air proses pengolahan publikasi	47

Tabel 5. 10 Analisis Function Point Perangkat Lunak Sistem Pintu Air proses pengolahan pengaduan	47
Tabel 5. 11 Analisis DET,RET, Dan FTR Perangkat Lunak DBA Ticketing proses mengelola pembelian.....	48
Tabel 5. 12 Perhitungan UFP Perangkat Lunak DBA Ticketing pada proses mengelola pembelian	50
Tabel 5. 13 Analisis DET,RET, Dan FTR Perangkat Lunak DBA Ticketing proses mengelola laporan keuangan	50
Tabel 5. 14 Perhitungan UFP Perangkat Lunak DBA Ticketing pada proses mengelola laporan keuangan	52
Tabel 5. 15 Analisis DET,RET, Dan FTR Perangkat Lunak DBA Ticketing proses mengelola mengelola agen dan saldo	52
Tabel 5. 16 Perhitungan UFP Perangkat Lunak DBA Ticketing pada proses mengelola laporan agen dan saldo	54
Tabel 5. 17 Analisis DET,RET, Dan FTR Perangkat Lunak DBA Ticketing proses mengelola tiket.....	54
Tabel 5. 18 Perhitungan UFP Perangkat Lunak DBA Ticketing pada proses mengelola tiket.....	56
Tabel 5. 19 Proses Perhitungan KSLOC Perangkat Lunak DBA Ticketing	56
Tabel 5. 20 Analisis DET,RET, dan FTR perangkat lunak Sistem Informasi Pintu Air proses mengelola pintu air.....	57
Tabel 5. 21 Perhitungan UFP perangkat lunak Sistem Informasi Pintu Air pada proses mengelola pintu air.....	58
Tabel 5. 22 Analisis DET,RET, dan FTR perangkat lunak Sistem Informasi Pintu Air proses mengelola laporan keuangan	59
Tabel 5. 23 Perhitungan UFP perangkat lunak Sistem Informasi Pintu Air pada proses mengelola laporan keuangan	60
Tabel 5. 24 Analisis DET,RET, dan FTR perangkat lunak Sistem Informasi Pintu Air proses mengelola profil	60
Tabel 5. 25 Perhitungan UFP perangkat lunak Sistem Informasi Pintu Air pada proses mengelola profil	62
Tabel 5. 26 Analisis DET,RET, dan FTR perangkat lunak Sistem Informasi Pintu Air proses mengelola publikasi	62
Tabel 5. 27 Perhitungan UFP perangkat lunak Sistem Informasi Pintu Air pada proses mengelola publikasi	63
Tabel 5. 28 Analisis DET,RET, Dan FTR Perangkat Lunak Sistem Informasi Pintu Air proses mengelola pengaduan	64



Tabel 5. 29 Perhitungan UFP Perangkat Lunak Sistem Informasi Pintu Air Pada proses mengelola pengaduan 65

Tabel 5. 30 Proses Perhitungan KSLOC Perangkat Lunak Sistem Informasi Pintu Air 65

Tabel 5. 31 Perhitungan Nilai Faktor Skala Perangkat Lunak DBA Ticketing 66

Tabel 5. 32 Perhitungan Nilai Faktor Skala Perangkat Lunak Sistem Pintu Air 67

Tabel 5. 33 Perhitungan Nilai Faktor Skala Perangkat Lunak DBA Ticketing 68

Tabel 5. 34 perhitungan nilai faktor skala perangkat lunak Sistem Pintu Air 69



DAFTAR GAMBAR

Gambar 3. 1 Metodologi Penelitian.....	20
Gambar 3. 2 contoh DFD level 1 sistem.....	22
Gambar 4. 1 DFD level 0 DBA ticketing.....	32
Gambar 4. 2 DFD level 1 DBA ticketing.....	33
Gambar 4. 3 DFD level 2 pemesanan DBA ticketing.....	34
Gambar 4. 4 DFD level 2 mengelola laporan keuangan DBA ticketing.....	34
Gambar 4. 5 DFD level 2 mengelola agen dan deposit DBA ticketing.....	35
Gambar 4. 6 DFD level 2 pembuatan tiket DBA ticketing.....	36
Gambar 4. 7 DFD level 0 Perangkat Lunak Sistem informasi Pintu Air.....	36
Gambar 4. 8 DFD level 1 Perangkat Lunak Sistem informasi Pintu Air.....	37
Gambar 4. 9 DFD level 2 mengelola pintu air	38
Gambar 4. 10 DFD level 2 mengelola laporan keuangan.....	38
Gambar 4. 11 DFD level 2 mengelola profil	39
Gambar 4. 12 DFD level 2 mengelola berita	40
Gambar 4. 13 DFD level 2 mengelola pengaduan	40

DAFTAR LAMPIRAN

LAMPIRAN A Hasil Wawancara	78
A.1 Hasil Wawancara	78
A.2 Validasi Wawancara	79
LAMPIRAN B Lembar Validasi PDM dan Data flow diagram	80
LAMPIRAN C Lembar penilaian scale factor	81
LAMPIRAN D effort multiplier	83



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Berkembang pesatnya dunia komputer maka diikuti oleh perkembangan perangkat lunak. Maka proyek perangkat lunak otomatis akan semakin bertambah. Beberapa proyek perangkat lunak ada yang gagal karena perusahaan kurang mementingkan manajemen proyek perangkat lunak (maswinandar,2016). Itulah yang mendasari pentingnya penerapan manajemen proyek perangkat lunak bagi suatu perusahaan, organisasi, dan pemerintahan (Schwalbe, 2014).

Estimasi biaya perangkat lunak sendiri adalah analisis yang digunakan bagi manajer untuk menentukan proyek perangkat lunak yang diketahui dari faktor estimasi biaya yang muncul saat proyek tersebut direncanakan dan juga dikerjakan. Analisis estimasi biaya pembuatan perangkat lunak ini juga merupakan sub bagian dari manajemen proyek perangkat lunak yang banyak dipakai sebagai pendukung dalam mengambil keputusan sehingga selanjutnya dalam pelaksanaan dapat digunakan bantuan metode untuk estimasi biaya pembuatan perangkat lunak.

CV. Profile Image Studio merupakan perusahaan di malang yang bergerak dalam bidang IT sejak 2011, yang menjadi produk utama adalah pembuatan konsep dan pengembangan website untuk bisnis digital. Perusahaan ini menerima pengerjaan perangkat lunak dari swasta maupun dari instansi pemerintah.

Saat ini dalam penentuan proyek di CV. Profile Image Studio perusahaan hanya melihat dari aspek teknis, biaya yang ditawarkan oleh klien dan juga dari kesanggupan jumlah sumberdaya perusahaan untuk mengerjakan proyek atau dikenal dengan istilah *Guesstimate* (Marchewka, 2003). Menggunakan metode *Guesstimate* akan menyebabkan estimasi biaya bisa lebih atau kurang dari biaya sesungguhnya, sedangkan menggunakan metode *parametric* hasil dari estimasi akan lebih akurat. Pihak CV. Profile Image Studio menjelaskan ketika mereka akan mengembangkan suatu perangkat lunak tetapi mereka belum memiliki pengalaman dalam membuat perangkat lunak tersebut sebelumnya maka mereka akan sedikit kesulitan ketika menggunakan metode *Guesstimate*. Sehingga alangkah lebih baik apabila terdapat sebuah metode pendekatan yang dapat memberikan hasil dalam menghitung estimasi usaha (*effort*) dan juga estimasi biaya. Karena utamanya penentu dari berhasil atau tidaknya suatu pengerjaan proyek IT adalah tentang estimasi biaya (B. Prasetyo, 2006)

Penelitian ini bertujuan untuk mengimplementasikan salah satu model estimasi biaya yang populer adalah model *COCOMO (Constructive Cost Model) II* yang mana model ini digunakan untuk menghitung nilai usaha dan hasilnya nanti untuk menghitung estimasi waktu, sumber daya manusia serta biaya pengembangan sistem. *COCOMO II* merupakan hasil pengembangan dari *COCOMO 81'* yang dikembangkan oleh B. Boehm pada 1981 dan dipublikasikan pada tahun 1997. Model ini memiliki kelebihan yaitu bisa memprediksi dengan

kelas waktu, biaya, dan juga pegawai yang dibutuhkan untuk menyelesaikan suatu proyek IT (Merlo, 2002).

Metode *COCOMO II* dipilih untuk digunakan karena metode tersebut dapat memudahkan dalam melihat biaya, waktu serta dasar dari semua sisi proyek dalam *software* dengan mudah dapat dipahami sehingga nantinya resiko proyek dapat dikurangi. Hasil ini berdasarkan penelitian yang dilakukan oleh T.N.Sharma berjudul "*Analysis Software Cost Estimation using COCOMO II*". Dari penelitian disimpulkan bahwa metode *COCOMO II* memudahkan dalam mengklarifikasikan hasil yang muncul dengan tidak hanya memperkiraan biaya dan durasi proyek, tetapi juga meminta untuk memverifikasi semua sisi dasar dari suatu perangkat lunak (T.N.Sharma, 2011).

Dari penjelasan diatas, penelitian ini akan mengimplementasikan metode *COCOMO II* di CV. Profile Image Studio, yang nantinya hasil estimasi dari penelitian ini akan digunakan sebagai masukan bagi pihak CV. Profile Image Studio dalam menghitung biaya.

1.2 Rumusan Masalah

Setelah dijelaskan latar belakang pada penelitian ini, maka penulis akan merumuskan masalah pada penelitian ini, yaitu:

1. Bagaimana perhitungan estimasi biaya proyek pembuatan perangkat lunak dengan metode *COCOMO II* ?
2. Bagaimana perhitungan estimasi waktu proyek untuk pembuatan sebuah *software* menggunakan metode *COCOMO II* ?
3. Bagaimana menghitung estimasi SDM (Sumber Daya Manusia) proyek pembuatan *software* menggunakan metode *COCOMO II* ?
4. Bagaimana hasil estimasi biaya menggunakan metode *COCOMO II* dengan *actual cost* dari CV. Profile Image Studio ?

1.3 Tujuan

Tujuan dari dilakukannya penelitian ini adalah :

1. Mendapatkan hasil perkiraan estimasi biaya pembuatan perangkat lunak menggunakan *COCOMO II*.
2. Mendapatkan hasil perkiraan estimasi waktu pembuatan perangkat lunak menggunakan *COCOMO II*.
3. Mendapatkan hasil perkiraan estimasi sumber daya manusia pembuatan perangkat lunak menggunakan *COCOMO II*.
4. Mendapatkan perbandingan hasil perhitungan antara menggunakan Metode *COCOMO II* dengan *actual cost*.

1.4 Manfaat

Penelitian ini mempunyai manfaat sebagai berikut:

1. Mendapatkan tambahan keilmuan mengenai estimasi biaya menggunakan metode *COCOMO II*.
2. Untuk tambahan informasi tentang bagaimana pertimbangan pengambilan keputusan oleh pihak CV. Profile Image Studio untuk estimasi biaya pembuatan perangkat lunak selanjutnya.

1.5 Batasan Masalah

Batasan yang peneliti gunakan dalam menyusun skripsi ini yaitu:

1. Data yang peneliti gunakan adalah data proyek dari CV. Profile Image Studio.
2. Metode estimasi biaya yang digunakan adalah metode *COCOMO II*.
3. Metode yang digunakan adalah *COCOMO II* sub-model *Post-Architecture Design Model*.

1.6 Sistematika Pembahasan

Berikut ini adalah Penjelasan masing-masing bab skripsi pada penelitian ini, yaitu:

Bab 1 : Pendahuluan

Berisi penjelasan tentang pendahuluan, masalah penelitian, tujuan penelitian, manfaat penelitian, batasan masalah penelitian, dan juga sistematika pembahasan.

Bab 2 : Dasar Teori

menjelaskan referensi yang digunakan, dapat berupa jurnal ilmiah juga buku yang sesuai digunakan untuk penelitian ini.

Bab 3 : Metodologi Penelitian

Menjelaskan metode, data dan sumber yang digunakan, analisis data dan analisis hasil yang akan digunakan dalam penelitian ini.

Bab 4: Hasil Penelitian

Menjelaskan tentang hasil perolehan data observasi dan wawancara serata lembar penilaian yang diisi oleh tim developer CV. Profile Image Studio.

Bab 5: Analisis

Menjelaskan tentang skala dari proyek dan hasil estimasi biaya, waktu, dan sumber daya manusia dalam pengembangan perangkat lunak dengan metode *COCOMO II*.

Bab 6: Pembahasan

Menjelaskan tentang hasil dari estimasi biaya, waktu, dan SDM (sumber daya manusia) pada pengembangan *software* menggunakan metode *COCOMO II* dengan actual cost dari CV. Profile Image Studio.

Bab 7: Penutup

Disini merupakan hasil kesimpulan dari penelitian yang sudah dilaksanakan.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Visi dan Misi perusahaan

2.1.1 Visi

Menjadi perusahaan terdepan yang dapat memberi nilai keunggulan bagi ekosistem bisnis di Indonesia dengan menggunakan pendekatan berbasis teknologi.

2.1.2 Misi

1. Menjalinkan hubungan kerjasama dengan beraneka ragam bidang industri
2. Menghadirkan potensi baru dengan merangkai padu teknologi termutakhir
3. Membangun standar industri yang ideal dalam hal pelayanan jasa teknologi informasi

2.2 Pengertian Proyek

Proyek adalah bagaimana cara untuk menyelesaikan suatu pekerjaan dengan asumsi itu adalah sesuatu yang besar (Cotterel, 2006). Proyek memiliki karakteristik sebagai berikut :

1. Tidak rutin melibatkan pekerjaan
2. Memerlukan perencanaan
3. Dapat melihat objek yang spesifik atau dapat membuat produk yang spesifik
4. Mempunyai beberapa fase dalam penyelesaian pekerjaan
5. Membatasi penggunaan sumberdaya
6. Kompleks dan besar itu adalah proyek

Proyek adalah cara untuk menghasilkan suatu produk atau layanan yang unik yang bersifat sementara pada suatu usaha (Schwalbe, 2014).

Proyek memiliki atribut yaitu :

1. Proyek harus menghasilkan suatu produk khusus, layanan dan hasil akhir, intinya proyek harus memiliki tujuan yang khusus..
2. Proyek mempunyai awalan dan akhir sehingga dapat dikatakan Proyek bersifat sementara.
3. Sumberdaya dapat berupa *hardware*, *software* dan sumberdaya lainnya yang dilakukan oleh pengguna sistem tersebut sehingga Proyek mendapatkan sumberdaya dari berbagai area.
4. Proyek harus memiliki pelanggan utama (*primary customer*) atau sponsor.

5. Proyek melibatkan ketidakpastian, karena setiap proyek bersifat unik sehingga sangat sulit untuk menentukan objektifitas proyek, mengestimasi waktu dan biaya proyek.

Setiap proyek memiliki batasan yang berbeda terhadap ruang lingkup, waktu dan biaya yang biasanya disebut sebagai *triple constraint* (tiga kendala) (Schwalbe, 2004). Oleh karena itu pula seorang *project manager* diharuskan memperhatikan hal-hal ini dalam manajemen proyek :

1. *Scope* : apa yang ingin dicapai dalam proyek, produk atau layanan apa yang pelanggan harapkan dari proyek tersebut
2. *Time* : berapa lama waktu yang dibutuhkan untuk menyelesaikan proyek dan bagaimana jadwal kegiatan proyek akan dilaksanakan
3. *Cost* : total biaya yang dibutuhkan untuk menyelesaikan proyek

Disini dapat disimpulkan bahwa, proyek adalah menghasilkan produk atau hasil yang baik dan jelas melalui usaha yang memiliki tempo sementara namun membutuhkan atribut yang dapat mengikuti jalannya proyek. Antara lain sumber daya, waktu, dan biaya.

2.3 Pengertian Manajemen Proyek Teknologi Informasi

Pengaplikasian pengetahuan, keahlian, peralatan dan teknik untuk kegiatan proyek yang sesuai dengan kebutuhan proyek dapat disebut Manajemen Proyek (Schwalbe, 2014).

Dari uraian di atas dapat disimpulkan bahwa, manajemen proyek teknologi informasi adalah merencanakan, menyusun, mengorganisasikan seluruh sumber daya, waktu, dan biaya untuk membuat suatu rekayasa manusia terhadap proses penyampaian informasi kepada penerima akan lebih cepat, lebih luas sebarannya dan lebih lama penyimpanannya.

2.4 Estimasi biaya perangkat lunak

Estimasi biaya perangkat lunak adalah proses dimana memprediksi biaya yang akan diperlukan saat melakukan pengembangan perangkat lunak. biaya perangkat lunak dapat diprediksi dari jumlah staf, jadwal dan biaya yang akan dikeluarkan dalam penyelesaian proyek pengembangan *software* (Maswinandar, 2016).

Metode yang dapat digunakan untuk menghitung estimasi biaya ada banyak namun dapat dikategorikan berupa dua pemodelan utama yaitu algoritmik dan metode non-algoritmik. Pada algoritmik model dalam penghitungan hasil estimasi biaya menggunakan model matematik sedangkan model non algoritmik adalah model yang menggunakan kemiripan proyek dengan proyek yang akan dianalisis menggunakan informasi estimasi biaya dari proyek sebelumnya (M.G. Bintiri, 2012).

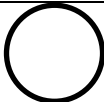

2.5 Metode COCOMO II



COCOMO (*Constructive Cost Model*) II adalah kategori pemodelan algoritmik yang digunakan sebagai alat bantu menghitung estimasi biaya, usaha dan juga waktu proyek seseorang ataupun kelompok saat melakukan aktivitas pengembangan perencanaan perangkat lunak. *COCOMO II* merupakan hasil pengembangan dari *COCOMO 81* yang dikembangkan oleh Boehm pada 1981 dan dipublikasikan pada tahun 1997. Model ini menggunakan rumus regresi dasar dalam penghitungannya dan menggunakan karakteristik proyek-proyek yang ada saat ini sebagai parameter. Menjadi salah satu model estimasi biaya perangkat lunak yang paling banyak digunakan dan dibahas dalam industri adalah Model *COCOMO* yang asli. Model ini menghitung fungsi ukuran program dan serangkaian pengendali biaya yang menyangkut penilaian subyektif terhadap produk, perangkat keras, personel dan atribut proyek sebagai usaha pengembangan perangkat lunak (Merlo, 2002). *COCOMO II* dinilai lebih realistis dikarenakan perhitungannya didasarkan pada nilai function point dan titik karakteristik obyek pada proyek (T.N.Sharma, 2011).

2.6 Data Flow Diagram

Data flow Diagram (DFD) merupakan gambaran yang merepresentasikan aliran data informasi dan juga perpindahan yang diterapkan sebagai data yang bergerak berupa input dan menghasilkan output. DFD dapat menggambarkan mekanisme yang digunakan untuk memodelkan fungsional sistem serta model aliran data informasi (Sommerville, 2010). *Data Flow* adalah representasi gambar grafik yang dapat menggambarkan aliran data informasi dan juga transformasi informasi yang digunakan sebagai data yang mengatur masukan (*input*) dan keluaran (*output*). *DFD* tidak sesuai untuk memodelkan sistem yang menggunakan pemrograman berorientasi objek. Oleh karena itu, DFD lebih sesuai untuk menggambarkan fungsi-fungsi sistem perangkat lunak yang menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi bagi-bagiannya ke dalam fungsi dari prosedur-prosedur dalam perangkat lunak (A.S. Rosa, 2014). Notasi-notasi pada *DFD* (Edward Yourdon dan Tom DeMarco, 1980) pada Tabel 2.1 berikut.

Tabel 2. 1 Data Flow Diagram

Keterangan	Simbol
Proses / <i>Process</i>	
Entitas Luar, Batas sistem / <i>External Entity, Boundary system</i>	

Keterangan	Simbol
Aliran data / <i>Data Flow</i>	
Penyimpan Data / <i>Data Store</i>	

Jadi disini DFD digunakan untuk menggambarkan fungsi fungsi dari sistem yang menggunakan pemrograman terstruktur dikarenakan pemrograman terstruktur membagi prosedur berdasarkan fungsi yang ada pada sistem.

2.7 Function Point

Dengan cara menganalisis kebutuhan fungsional sistem yang direpresentasikan ke dalam Data Flow Diagram (DFD) untuk nantinya mempermudah analisis ke dalam *function point* (FP) (Neelam Bawane nee' Singhal, 2008). *Function Point* digunakan untuk mengatasi kesulitan yang berhubungan dengan baris kode sebagai ukuran suatu perangkat lunak, serta untuk membantu dalam mengembangkan mekanisme prediksi usaha yang terkait dengan pengembangan perangkat lunak. Dalam melakukan analisis *Function Point*, sistem dibagi menjadi lima kelas besar sistem yang menilai fungsional umum dari sistem. Berikut ini merupakan lima komponen yang digunakan untuk mengukur jumlah kompleksitas dan fungsionalitas dalam sebuah perangkat lunak. Tabel 2.2 berikut ini merupakan komponen *Function Point*.

Tabel 2. 2 komponen function point

Komponen	Keterangan
<i>External Input (EI)</i>	Fungsi yang memindahkan data dari luar ke dalam aplikasi tanpa menyajikan manipulasi data.
<i>External Output (EO)</i>	Fungsi yang memindahkan data dari dalam ke pengguna dan menyajikan beberapa data yang telah dimanipulasi.
<i>External Inquiry (EQ)</i>	Fungsi yang memindahkan data dari dalam ke pengguna dan menyajikan beberapa data yang tanpa dimanipulasi.
<i>Internal Logical File (ILF)</i>	Logika dalam bentuk data tetap, yang dikelola oleh aplikasi melalui penggunaan masukan dari luar.
<i>External Interface File (EIF)</i>	Logika dalam bentuk data tetap, yang digunakan oleh aplikasi tetapi tidak berjalan dalam aplikasi tersebut.

Nantinya setiap komponen *Function Point* diklasifikasikan tingkat kompleksitasnya. Tingkat kompleksitas menentukan kuantitas *Unadjusted Function Point*. Bobot kompleksitas diklasifikasikan berdasarkan *DET*, *RET*, dan *FTR*. Berikut ini merupakan penjabaran *DET*, *RET*, dan *FTR* (Longstreet, 2004). *RET (Record Element Types)* *RET* merupakan subgrup data yang dikenali oleh pengguna dalam *Internal Logical File* atau *External Logical File*. Contohnya adalah file mahasiswa yang menyimpan nomor mahasiswa, nama, dan lain seterusnya. *DET (Data Elements Types)* *DET* dikenali oleh pengguna sebagai sesuatu yang unik dan tidak berulang apabila berulang maka dihitung 1 *DET*. Elemen data yang unik untuk membedakan komponen *Function Point* yang satu dengan komponen *Function Point* yang lainnya. Berupa data *input* untuk *EI*. *FTR (Files Type References)* dapat berupa *ILF* atau *EIF*. Setiap *ILF* yang berupa *EI* dihitung sebagai *FTR*. Setiap *ILF* atau *EIF* yang direferensikan oleh *EI* atau *EO* atau *EQ* untuk memelihara (*insert, update, delete*) dianggap sebagai *FTR*. Untuk dapat membedakan *EI* atau *EO* atau *EQ* dengan *EI* atau *EO* atau *EQ* lainnya.

Kemudian klasifikasikan setiap fungsi pada tabel diatas dihitung ke level kompleksitas *Low, Average, dan High* tergantung pada jumlah tipe elemen data yang ditentukan berdasarkan jumlah jenis file yang dijelaskan pada tabel 2.3 berikut ini (Longstreet, 2004).

Tabel 2. 3 Klasifikasi tipe data

<i>Internal logical File dan External Logical File</i>			
	<i>Data element</i>		
<i>Record Element Type</i>	<i>0-19</i>	<i>20-50</i>	<i>>50</i>
<i>1</i>	<i>Low</i>	<i>Low</i>	<i>Average</i>
<i>2-5</i>	<i>Low</i>	<i>Average</i>	<i>High</i>
<i>>5</i>	<i>Average</i>	<i>High</i>	<i>High</i>
<i>External Output dan External Inquiry</i>			
	<i>Data element</i>		
<i>File type</i>	<i>1-5</i>	<i>6-19</i>	<i>>19</i>
<i>0 or 1</i>	<i>Low</i>	<i>Low</i>	<i>Average</i>
<i>2-3</i>	<i>Low</i>	<i>Average</i>	<i>High</i>
<i>>4</i>	<i>Average</i>	<i>Average</i>	<i>High</i>
<i>External Input</i>			
	<i>Data element</i>		
<i>File type</i>	<i>1-4</i>	<i>5-15</i>	<i>>15</i>



0 or 1	Low	Low	Average
2-3	Low	Average	High
>3	Average	Average	High

Selanjutnya adalah menghitung bobot kompleksitas dari fungsi-fungsi tersebut. Bobot kompleksitas pada tiap fungsi dihitung menggunakan Tabel 2.4 berikut ini.

Tabel 2. 4 Klasifikasi Bobot kompleksitas

Tipe Fungsi	Bobot Kompleksitas		
	Low	Average	High
Internal Logical File	7	10	15
External Logical File	5	7	10
External input	3	4	6
External inquiry	3	4	6
External Output	4	5	7

Langkah selanjutnya adalah dengan menghitung *Unadjusted Function point* (UFP) dengan cara tambahkan semua fungsi-fungsi menjadi satu nilai *Unadjusted Function point* (UFP). Lalu nilai UFP yang telah didapatkan diubah ke dalam *Source lines of Code* (SLOC). Setelah itu nilai SLOC dibagi dengan 1000 untuk mendapatkan nilai *KSLOC* yang bisa dimasukkan dalam persamaan usaha. Selanjutnya kalikan dengan Bahasa pemrograman yang digunakan dalam sistem tersebut. Berikut ini adalah tabel 2.5 Bahasa pemrograman yang ada berdasarkan QSM (*Quantitative Software Management*) (*Quantitative software management*, -).

Tabel 2. 5 QSM language Table

bahasa	QSM SLOC/FP Data			
	Average	Median	Low	High
FoxPro	36	35	34	38
HTML *	34	40	14	48
J2EE *	46	49	15	67
Java *	53	53	14	134
JavaScript *	47	53	31	63
JCL *	62	48	25	221
LINC II	29	30	22	38
Lotus Notes *	23	21	19	40
Natural *	40	34	34	53
.NET *	57	60	53	60

Oracle *	37	40	17	60
PACBASE *	35	32	22	60
Perl *	24	15	15	60
PL/I *	64	80	16	80
PL/SQL *	37	35	13	60
Powerbuilder *	26	28	7	40
REXX *	77	80	50	80
Sabretalk *	70	66	45	109
SAS *	38	37	22	55
Siebel *	59	60	51	60
SLOGAN *	75	75	74	75
SQL *	21	21	13	37
VB.NET *	52	60	26	60
Visual Basic *	42	44	20	60

QSM SLOC/FP level data yang akan peneliti gunakan adalah median dikarenakan median indikator yang paling akurat dibandingkan dengan level yang lain.

2.8 Scale factor

Scale factor (SF) merupakan cara untuk menentukan usaha proyek dan juga karakteristik dari suatu proyek itu sendiri. *Scale factor* ini digunakan pada submodel *COCOMO II early design* dan juga *post-architecture model*. Berikut ini adalah persamaan yang digunakan untuk menghitung nilai *scale factor*.

$$E = B + 0.01 \times SF \quad \text{(Persamaan 1).}$$

Dimana:

E= faktor eksponen.

B = nilai koefisien 0.91(untuk COCOMO II.2000).

SF= total penjumlahan nilai *scale factor*.

Eksponen E adalah agregasi dari lima skala faktor (SF) yang memperhitungkan ekonomi atau disekonomi yang ditemui pada proyek perangkat lunak dengan ukuran berbeda. Jika $E < 1.0$, proyek ini menunjukkan skala ekonomis. Skala ekonomi sendiri merupakan nilai keuntungan biaya yang diperoleh dari suatu proyek. Beberapa skala ekonomi proyek dapat dicapai melalui alat ukur khusus untuk proyek (misalnya, simulasi, testbeds), tetapi secara umum ini sulit untuk dicapai. Jika $E = 1.0$, skala ekonomi dan disekonominya seimbang. Model linear ini sering digunakan untuk estimasi biaya proyek-proyek kecil. Jika $E > 1.0$, proyek ini menunjukkan skala disekonomi. Ini umumnya karena dua faktor utama: (1) meningkatnya overhead komunikasi interpersonal dan (2) meningkatnya integrasi sistem besar. Proyek yang lebih besar akan memiliki lebih banyak personel, dan dengan demikian jalur komunikasi interpersonal akan lebih banyak (Barry Boehm, 2000).

Untuk mendapatkan nilai *Scale factor*, terdapat 5 parameter pengukuran untuk submodel *Post-Architecture model*, yaitu: *Precedentedness (PREC)*, *Development Flexibility (FLEX)*, *Risk Resolution (RESL)*, *Team Cohesion (TEAM)*, dan *Process Maturity (PMAT)*. Berikut ini adalah define dari masing-masing *scale factor* tersebut:

a) *Precedentedness (PREC)*

Precedentedness (PREC) merupakan salah satu dari faktor skala yang menggambarkan pengalaman organisasi terhadap proyek-proyek sebelumnya yang sejenis. Makin besar ketidakpastian maka semakin tinggi nilai eksponennya.

b) *Development Flexibility (FLEX)*

Development Flexibility (FLEX) merupakan salah satu dari faktor skala yang menggambarkan kemampuan klien dalam menentukan tujuan dan menyampaikan serta mengkomunikasikan kebutuhan perangkat lunak kepada tim pengembang perangkat lunak

c) *Risk Resolution (RESL)*

Risk Resolution (RESL) merupakan salah satu dari faktor skala yang menggambarkan identifikasi resiko yang terkait dengan proyek. Jika semakin sedikit ataupun tidak ada dokumentasi risiko maka akan memperbesar nilai eksponennya.

d) *Team Cohesion (TEAM)*

Team Cohesion (TEAM) merupakan salah satu dari faktor skala yang menggambarkan kemampuan setiap anggota tim proyek dalam berkomunikasi dan bekerjasama

e) *Process Maturity (PMAT)*

Process Maturity (PMAT) merupakan salah satu dari faktor skala yang menggambarkan kematangan proses pengembangan perangkat lunak dalam organisasi. Hal ini didasarkan pada Model Kematangan Kemampuan Rekayasa Perangkat Lunak atau *Capability Maturity Model (CMM)*.

Nilai dari setiap atribut *scale factor* nantinya akan didapatkan berdasarkan kuesioner yang diisi oleh tim pengembang perangkat lunak. Selanjutnya bobot penilaian dari setiap faktor skala yang telah didapatkan nantinya akan dihitung dan digunakan dalam perhitungan *effort estimation*. Nilai dari setiap atribut akan dijelaskan pada Tabel 2.6 di bawah ini (Barry Boehm, 2000).

Tabel 2. 6 Bobot scale factor

No.	Scale factors	Very Low	Low	Nominal	High	Very High	Extra High
1.	<i>Precedentness (PREC)</i>	6.20	4.96	3.72	2.48	1.24	0.00
2.	<i>Development Flexibility (FLEX)</i>	5.07	4.05	3.04	2.03	1.01	0.00
3.	<i>Risk Resolution (RESL)</i>	7.07	5.65	4.24	2.83	1.41	0.00
4.	<i>Team Cohesion (TEAM)</i>	5.38	4.38	3.29	2.1 9	1.10	0.00
5.	<i>Process Maturity (PMAT)</i>	7.80	6.24	4.68	3.12	1.56	0.00

Setelah mendapat nilai dari setiap dari 5 atribut dari lembar kuesioner tambahkan setiap skala faktor sehingga menghasilkan total penilaian skala faktor yang nantinya digunakan untuk menghitung faktor eksponen menggunakan persamaan 1 sehingga nilai dari faktor eksponen nanti akan digunakan dalam persamaan *effort estimation*.

2.9 Effort Multiplier

Effort multiplier merupakan faktor penggerak biaya dalam menentukan usaha yang diperlukan dalam pengerjaan proyek perangkat lunak. *Effort multiplier* mempunyai 7 kategori dan 17 atribut pada submodel *post-architecture* model *COCOMO II*. Penjelasan kategori dan juga setiap atribut pada *effort multiplier* untuk submodel *post-architecture* model akan dijelaskan pada tabel 2.7 di bawah ini (Barry Boehm, 2000).

Tabel 2. 7 Deskripsi Effort Multiplier

NO	Kategori	Atribut
1.	<i>RCPX (Product Reliability and Complexity)</i>	Penilaian <i>cost driver</i> terkait beberapa faktor dari produk yaitu : 1. Sejauh mana perangkat lunak menjalankan aplikasi sesuai fungsinya selama periode waktu (<i>RELY</i>). 2. Ukuran <i>database</i> yang digunakan. Ukuran dapat dihitung menggunakan D/P (<i>DATA</i>).



NO	Kategori	Atribut
		<p>3. Perangkat lunak dan perangkat keras dalam melakukan tugasnya seperti platform (arsitektur, sistem operasi, bahasa pemrograman dan antar muka yang terkait), sistem manajemen <i>database</i>, <i>browser</i> yang sesuai digunakan dalam menjalankan aplikasi ini (<i>CPLX</i>).</p> <p>4. Kesesuaian dokumentasi proyek terhadap kebutuhan siklus hidup perangkat lunak (<i>DOCU</i>).</p>
2.	<i>PERS (Personnel Capability)</i>	<p>Penilaian <i>cost driver</i> terkait beberapa kemampuan personel yaitu :</p> <p>5. Kemampuan personel dalam analisis dan desain, efisiensi dan ketelitian, serta kemampuan untuk berkomunikasi dan bekerja sama. Dalam hal ini, dapat dinilai dari sertifikasi yang sudah didapatkan personel atau pengalaman kerja tim dalam suatu proyek (<i>ACAP</i>).</p> <p>6. Kemampuan <i>programmer</i> dalam efisiensi penulisan kode program, ketelitian dan kemampuan untuk berkomunikasi dan bekerja sama sebagai sebuah tim. Dengan kata lain, berapa banyak proyek dimana <i>programmer</i> tersebut terlibat (<i>PCAP</i>).</p> <p>7. Pergantian personel tiap tahun pada proyek. Semakin sedikit pergantian maka semakin tinggi skala (<i>PCON</i>).</p>
3.	<i>RUSE (Developed for Reusability)</i>	<p>8. <i>Ruse</i> merupakan <i>cost driver</i> terkait tingkat upaya yang diperlukan untuk mengembangkan komponen yang dimaksudkan untuk digunakan kembali pada proyek-proyek yang sedang berjalan atau proyek di masa mendatang.</p>
4.	<i>PDIF (Platform Difficulty)</i>	<p>Penilaian <i>cost driver</i> terkait beberapa kendala dalam pengembangan yaitu :</p> <p>9. Persentase kendala waktu eksekusi yang diharapkan dapat digunakan pada sistem perangkat lunak (<i>TIME</i>).</p> <p>10. Persentase tingkat kendala penyimpanan utama yang dikenakan pada sistem perangkat lunak (<i>STOR</i>).</p>

NO	Kategori	Atribut
		11. Perubahan yang terjadi pada <i>hardware</i> dan <i>software</i> dalam kurun waktu tertentu (<i>PVOL</i>).
5.	<i>PREX (Personnel Experience)</i>	Penilaian <i>cost driver</i> terkait beberapa pengalaman yaitu : 12. pengalaman kerja tim proyek pada suatu proyek pengembangan aplikasi sistem perangkat lunak atau subsistem (<i>APLEX</i>). 13. pemahaman tim proyek dalam menggunakan <i>platform, interface database, jaringan, middleware (PLEX)</i> . 14. pengalaman tim proyek dalam pemrograman dengan bahasa tertentu dan pemanfaatan <i>CASE tool</i> dalam mengembangkan perangkat lunak (<i>LTEX</i>).
6.	<i>FCIL (Facilities)</i>	15. <i>TOOL</i> merupakan penilaian <i>cost driver</i> terkait penggunaan <i>CASE tool</i> dalam pengembangan perangkat lunak pada proyek, seperti dari mengubah kode yang sederhana menjadi terintegrasi. 16 <i>SITE</i> adalah bagaimana cara komunikasi yang digunakan dalam pengembangan perangkat lunak pada proyek.
7.	<i>SCED (Required Development Schedule)</i>	17. <i>SCED</i> merupakan Penilaian <i>cost driver</i> terkait tingkat persentase dari percepatan atau kemunduran jadwal terhadap jadwal suatu proyek yang telah ditetapkan sebelumnya.

Nilai dari setiap atribut *effort multiplier* nantinya akan didapatkan berdasarkan kuesioner yang diisi oleh tim pengembang perangkat lunak. Lalu nilai yang didapatkan dari pengisian kuesioner nantinya akan dirata-rata untuk mendapatkan nilai rerata *effort multiplier* yang digunakan dalam penghitungan *effort estimation* nantinya. Nilai dari setiap atribut akan dijelaskan pada Tabel 2.8 di bawah ini (Barry Boehm, 2000).

Tabel 2. 8 Tabel nilai effort multipliers

No.	Effort Multiplier	Very Low	Low	Nominal	High	Very High	Extra High
1.	RELY	slight inconvenience	low, easily recoverable losses	moderate, easily recoverable losses	high financial loss	risk to human life	n/a
		0.82	0.92	1.00	1.10	1.26	
2.	DATA	n/a	Testing DB bytes/Pgm SLOC ≤ 10	10 ≤ D/P ≤ 100	100 ≤ D/P ≤ 1000	D/P ≥ 1000	
			0.90	1.00	1.14	1.28	n/a
3.	CPLX	0.73	0.87	1.00	1.17	1.34	1.74
4.	RUSE	n/a	none	across project	across program	across product line	across multiple product lines
			0.95	1.00	1.07	1.15	1.24
5.	DOCU	Many life-cycle needs uncovered	Some life-cycle needs uncovered.	Right-sized to life-cycle needs	Excessive for life-cycle needs	Very excessive for life-cycle needs	
		0.81	0.91	1.00	1.11	1.23	n/a
6.	TIME			50% use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
		n/a	n/a	1.00	1.11	1.29	1.63
7.	STOR			50% use of available storage	70% use of available storage	85% use of available storage	95% use of available storage
		n/a	n/a	1.00	1.05	1.17	1.46
8.	PVOL		Major change every 12 month, Minor change every 1 month	Major: 6 month; Minor: 2 week.	Major: 2 month, Minor: 1 week	Major: 2 Week, Minor: 2 days	
		n/a	0.87	1.00	1.15	1.30	n/a
9.	ACAP	15th	35th	55th	75th	90th	



No.	Effort Multiplier	Very Low	Low	Nominal	High	Very High	Extra High
		percentile 1.42	percentile 1.19	percentile 1.00	percentile 0.85	percentile 0.71	n/a
10.	PCAP	15th percentile 1.34	35th percentile 1.15	55th percentile 1.00	75th percentile 0.88	90th percentile 0.76	n/a
11.	PCON	48% / year 1.29	24% / year 1.12	12% / year 1.00	6% / year 0.90	3% / year 0.81	
12.	APEX	≤2 months 1.22	6 months 1.10	1 year 1.00	3 years 0.88	6 years 0.81	n/a
13.	PLEX	≤2 months 1.19	6 months 1.09	1 year 1.00	3 years 0.91	6 year 0.85	n/a
14.	LTEX	≤2 months 1.20	6 months 1.09	1 year 1.00	3 years 0.91	6 year 0.84	
15.	TOOL	edit, code, debug 1.17	simple, frontend, backend, CASE, little integration 1.09	basic life- cycle tools, moderately integrated 1.00	strong, mature life- cycle tools, moderately integrated 0.90	strong, mature, proactive life- cycle tools, well integrated with processes, methods, reuse 0.78	n/a
16.	SITE	Inter- national Some phone, mail 1.22	Multi-city and Multi- company Individual phone, FAX 1.09	Multi-city or Multi- company Narrow band email 1.00	Same city or metro. Area, Wideband electronic communication 0.93	Same building or complex Wideband electronic communication, occasional video conerence. 0.86	Fully collocated Interactive multimedia 0.80



No.	Effort Multiplier	Very Low	Low	Nominal	High	Very High	Extra High
17.	SCED	75% of nominal	85% of nominal	100% of nominal	130% Of nominal	160% of nominal	
		1.43	1.14	1.00	1.00	1.00	n/a

2.10 Effort Estimation

Disini *Effort estimation* dinyatakan dalam *Person-Month* (PM). Satu *Person-Month* merupakan jumlah waktu yang harus dihabiskan untuk bekerja pada proyek pengembangan sistem perangkat lunak selama satu bulan. Yang hasilnya nanti berupa nilai untuk menghitung estimasi waktu. Berikut adalah persamaanya.

$$PM = A \times (Size)^e \times \prod_{i=1}^{17} E_{Mi} \quad \text{(Persamaan 2).}$$

Dimana:

PM = *person-month*

A = nilai koefisien 2.94 (untuk COCOMO II, 2000).

Size = nilai KSLOC.

e = nilai faktor eksponen.

EM = effort multiplier.

Selanjutnya hasil dari perhitungan nantinya akan digunakan untuk menghitung estimasi biaya proyek.

2.11 Estimasi Biaya

COCOMO II mempunyai kemampuan estimasi waktu yang mirip dengan COCOMO'81. Persamaan ini digunakan untuk *Early-Design* dan juga *Post-architecture COCOMO II*. Berikut adalah persamaanya.

$$TDEV = C \times (PM)^{(D+0,2 \times (E-B))} \quad \text{(Persamaan 3)}$$

Dimana:

TDEV = Time to Development/jumlah waktu pengembangan

C= nilai koefisien 3,67 (untuk COCOMO II, 2000).

D = nilai koefisien 0,28 (untuk COCOMO II, 2000).

B = nilai koefisien 0,91 (untuk COCOMO II, 2000).

E = nilai faktor eksponen



PM= *person per month*.

TDEV (Time Development) adalah jumlah waktu yang nantinya akan dibutuhkan selama pengembangan produk perangkat lunak.

Selanjutnya menghitung jumlah staff menggunakan rumus berikut :

$$\text{Average Staff} = \text{PM}/\text{TDEV} \quad \text{(Persamaan 4)}$$

Dimana :

Average staff = jumlah pegawai dibutuhkan

PM = *person per month*.

TDEV = Time to Development/jumlah waktu pengembangan

Selanjutnya menghitung biaya perbulan menggunakan persamaan berikut

$$\text{Biaya per bulan} = \text{Average staff} \times \text{UMR} \quad \text{(Persamaan 5)}$$

Dimana:

Biaya perbulan = biaya yang dikeluarkan tiap bulan

Average staff = jumlah pegawai dibutuhkan

UMR = Upah minimum regional daerah

Setelah diketahui nilai biaya per bulan Selanjutnya menghitung biaya perbulan menggunakan persamaan berikut

$$\text{Biaya total} = \text{biaya per bulan} \times \text{TDEV} \quad \text{(Persamaan 6)}$$

Dimana :

Biaya total = biaya total proyek

Biaya perbulan = biaya yang dikeluarkan tiap bulan

TDEV = Time to Development/jumlah waktu pengembangan

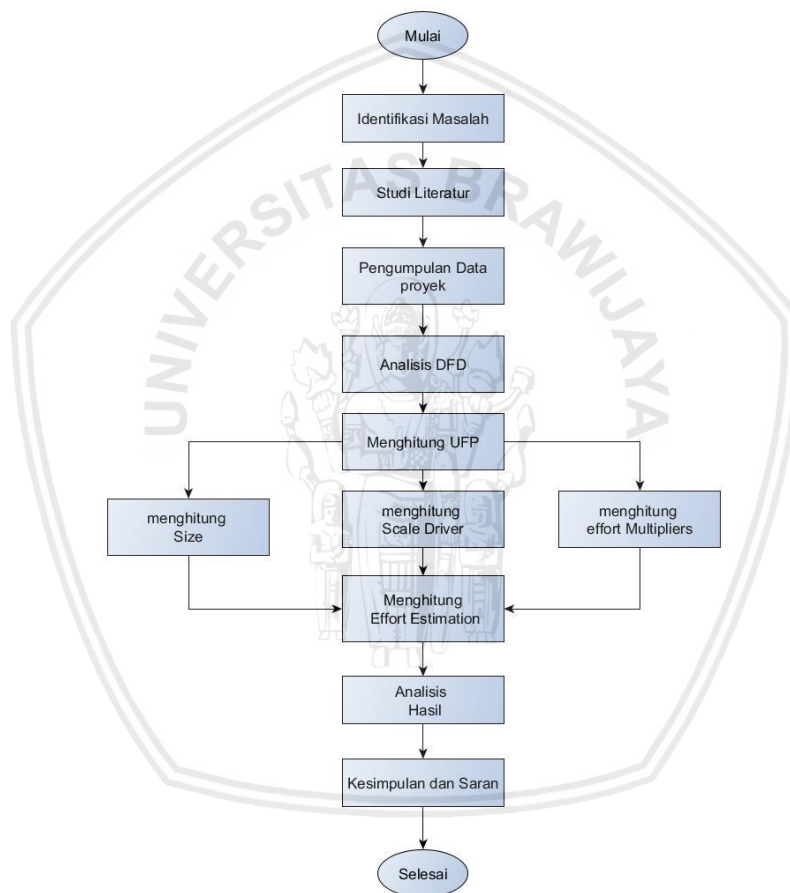
2.12 Agile Software Development

Dalam ukuran skala proyek *software development* di bagi atas 3 berdasarkan jumlah tim dalam proyek dengan jumlah SDM dalam satu tim berjumlah +/- 2 orang, yaitu: (a.) kategori skala kecil, jumlah 1 tim, (b.) kategori skala besar, jumlah 2-9 tim, (c.) kategori skala sangat besar, jumlah lebih dari 9 tim (Dingsøyr T., 2014).

BAB 3 METODOLOGI PENELITIAN

3.1 Sistematika Penelitian

Dalam sub bab ini peneliti ingin menggambarkan alur sistematika penelitian. Sistematika penelitian sendiri sangat diperlukan untuk suatu perencanaan dalam penelitian yang dilakukan agar dapat berjalan dengan baik dan juga sistematis. Sistematika penelitian yang akan peneliti lakukan dapat dilihat dari gambar 3.1.



Gambar 3. 1 Metodologi Penelitian

3.2 Identifikasi Masalah

Pada tahap identifikasi masalah ini dilakukan wawancara kepada perusahaan CV. Profile Image Studio terdapat beberapa masalah yang menyebabkan defisit pada beberapa proyek perangkat lunak dan juga pembengkakan waktu dikarenakan CV. Profile Image Studio menggunakan metode perkiraan dalam mengerjakan proyek perangkat lunak. Apabila hal ini

terus berlangsung dikhawatirkan akan mempengaruhi kepercayaan dan juga kepuasan pelanggan.

3.3 Studi Literatur

Pada tahap studi literatur, dilakukan pemahaman kepustakaan terhadap berbagai hal yang berhubungan dengan penelitian ini. Kepustakaan tersebut diantaranya tentang manajemen proyek perangkat lunak, dan metode *COCOMO II*. Studi literatur juga dilakukan dengan mempelajari penelitian yang dilakukan sebelumnya, buku, jurnal, dan juga informasi dari internet yang kredibel.

3.4 Pengumpulan Data

Pada tahap pengumpulan data ini, kami melakukan wawancara dengan CEO CV. Profile Image Studio untuk memperoleh informasi yang dibutuhkan sebagai bahan untuk analisis estimasi biaya perangkat lunak. Adapun beberapa langkah pengumpulan data yang dilakukan adalah:

3.4.1 Wawancara

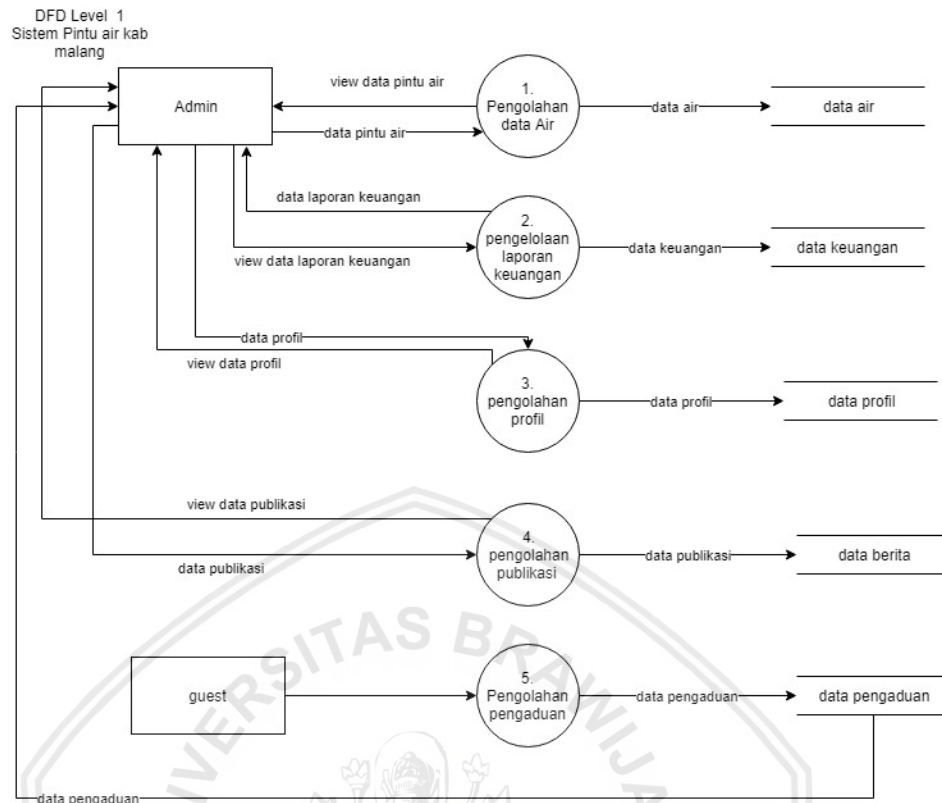
- Wawancara kepada pemilik CV. Profile Image Studio perihal manajemen proyek yang dilakukan di perusahaan CV. Profile Image Studio
- Wawancara dan permintaan data mengenai profil perusahaan, visi dan misi, struktur organisasi, dan pengalaman proyek yang pernah dikerjakan.
- Wawancara mengenai proyek yang akan diteliti

3.4.2 Observasi data

- Permintaan *Data flow Diagram* dari sistem yang akan diteliti.
- Pengisian kuesioner oleh tim proyek untuk mendapatkan penilaian *Scale factor* dan *Effort multipliers*

3.5 Analisis DFD (*Data Flow Diagram*)

Pada Tahap analisis DFD ini digunakan untuk menggambarkan kompleksitas dari sistem yang akan diteliti sehingga nantinya dapat mempermudah penentuan nilai UFP. Hal yang dilakukan adalah memetakan fungsi dari sistem ke dalam diagram DFD. Contoh DFD adalah seperti gambar di bawah ini.



Gambar 3. 2 contoh DFD level 1 sistem

3.6 Menghitung UFP (*Unadjusted Function Point*)

Pada tahap ini digunakan untuk mendapatkan nilai UFP dari fungsi DFD diatas berdasarkan 5 komponen *function point* yaitu: *External Input (EI)*, *External Output (EO)*, *External Inquiry (EQ)*, *Internal Logical File (ILF)*, *External Interface File (EIF)*. Setelah setiap fungsi di analisis sesuai 5 komponen *function point* maka dihitung nilai UFP berdasarkan jumlah *Data Element Type (DET)*, *Record Element Type (RET)*, dan *Files Type References (FTR)* lalu dikalikan nilai kompleksitas nya. Sehingga diperoleh nilai total UFP nya. selanjutnya ada 4 langkah dalam proses ini yaitu:

- Tentukan jumlah ILF, EIF, EI, EQ, dan EO dari setiap proses di dalam DFD.
- Tentukan jumlah DET, RET, dan FTR dari setiap ILF, EIF, EI, EQ, dan EO yang didapatkan dari PDM.
- Tentukan bobot kompleksitas dan nilai kompleksitas dari setiap ILF, EIF, EI, EQ, dan EO.
- Setelah itu menentukan nilai UFP dengan cara mengalikan nilai kompleksitas dengan jumlah dari setiap ILF, EIF, EI, EQ, dan EO dari setiap proses dalam DFD. Lalu tambahkan nilai UFP setiap proses sehingga menghasilkan nilai total UFP.



3.7 Menghitung Size

Disini kita akan menghitung nilai *size* perangkat lunak dalam satuan KSLOC yang akan digunakan dalam perhitungan *Effort Estimation* dengan cara menghitung SLOC (*Source Line Of Code*) yang dikonversikan ke dalam KSLOC (*Kilo Source Line Of Code*). Ada 4 langkah dalam proses ini yaitu:

1. masukkan nilai total UFP sistem perangkat lunak.
2. kalikan dengan standar konversi SLOC sesuai dengan bahasa pemrograman yang digunakan untuk pembuatan perangkat lunak yang hasilnya berupa SLOC
3. Lalu bagi nilai SLOC dengan 1000 agar menjadi nilai KSLOC.

3.8 Menghitung Scale factor

Tahap ini dilakukan untuk mendapatkan nilai *scale factor* dari parameter yang telah dikumpulkan dari kuesioner yang sebelumnya telah diisi oleh tim pengembang yang terdiri dari 5 atribut yaitu *Precedentness* (PREC), *Development Flexibility* (FLEX), *Risk Resolution* (RESL), *Team Cohesion* (TEAM), *Process Maturity* (PMAT).

Setelah mendapat nilai dari setiap dari 5 atribut dari lembar kuesioner tambahkan setiap skala faktor sehingga menghasilkan total penilaian skala faktor yang nantinya digunakan untuk menghitung faktor eksponen menggunakan persamaan 1 sehingga nilai dari faktor eksponen nanti akan digunakan dalam persamaan *effort estimation*.

3.9 Menghitung Effort multipliers

Pada tahap ini akan dilakukan untuk mendapatkan nilai *Effort multipliers* yang didapatkan dari 17 parameter dari *effort multipliers*.

Setelah mendapat nilai dari 17 parameter dari lembar kuesioner tambahkan setiap effort multiplier sehingga menghasilkan nilai rerata penilaian *effort multiplier* yang nantinya akan dimasukkan ke dalam perhitungan *effort estimation*.

3.10 Menghitung Effort Estimation

Tahap ini dilakukan untuk menghitung nilai estimasi proyek yang dapat diketahui dengan perhitungan rumus yang ada. Hasil yang ditampilkan merupakan hasil estimasi biaya proyek, waktu yang digunakan, dan jumlah pegawai untuk proyek yang diteliti. Ada 5 langkah untuk mendapatkan nilai *effort estimation* yaitu:

1. Menghitung nilai person-month menggunakan persamaan 2 .
2. Selanjutnya menghitung estimasi waktu menggunakan persamaan 3.

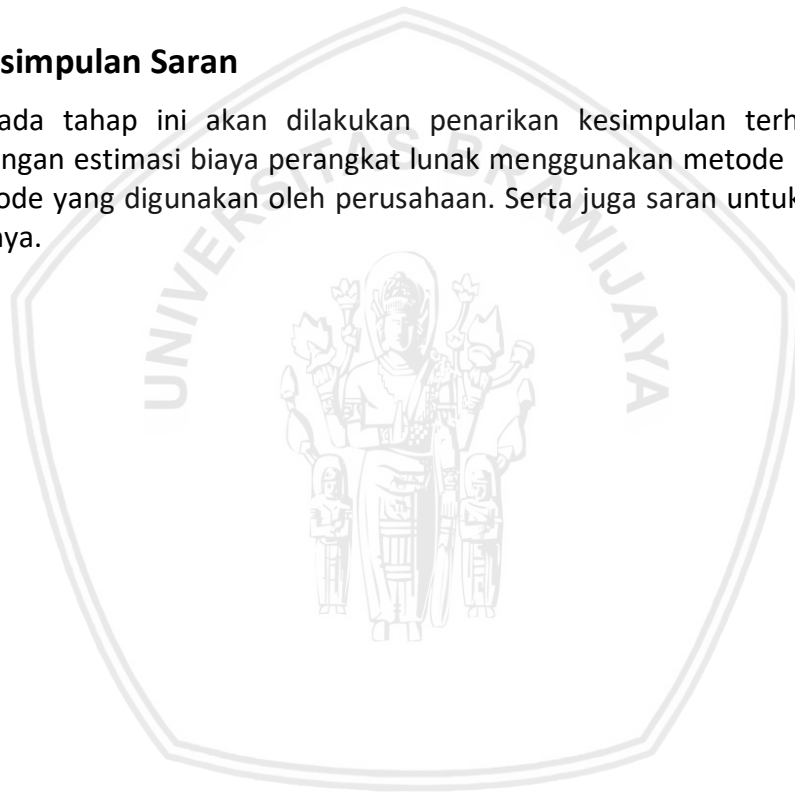
3. Selanjutnya menghitung jumlah personil menggunakan persamaan 4.
4. Selanjutnya menghitung total biaya per bulan menggunakan persamaan 5.
5. Selanjutnya menghitung total biaya proyek menggunakan persamaan 6.

3.11 Analisis Hasil

Setelah diketahui nilai estimasi biaya proyek perangkat lunak, waktu pengerjaan, serta jumlah pegawai yang bekerja menggunakan metode *COCOMO II*, kemudian akan dibandingkan dengan *actual cost* yang ada di CV. Profile Image Studio.

3.12 Kesimpulan Saran

Pada tahap ini akan dilakukan penarikan kesimpulan terhadap hasil perbandingan estimasi biaya perangkat lunak menggunakan metode *COCOMO II* dan metode yang digunakan oleh perusahaan. Serta juga saran untuk penelitian selanjutnya.



BAB 4 HASIL DATA

4.1 Hasil Wawancara

Wawancara dilakukan oleh peneliti terhadap tim proyek di CV. Profile Image Studio yang ikut terlibat dalam pengembangan proyek perangkat lunak menghasilkan informasi yang dapat digunakan untuk mendukung penelitian ini adalah profil CV. Profile Image Studio, deskripsi dari sistem *DBA Ticketing* dan Sistem informasi Pintu air, proses pengerjaan sistem *DBA Ticketing* dan Sistem informasi Pintu air, proses bisnis sistem *DBA Ticketing* dan Sistem informasi Pintu air, serta fitur fungsi juga fitur user pada sistem *DBA Ticketing* dan Sistem informasi Pintu air.

Perangkat lunak ini merupakan sistem *DBA Ticketing* untuk penjualan tiket bioskop. Sistem ini merupakan sistem yang tidak dapat diakses oleh public yang artinya sistem ini bersifat internal. Sistem ini merupakan sistem yang menangani pembelian ticket secara online melewati agen. Proses bisnis dari sistem E ticketing ini adalah dimulai dari pesanan pembeli melewati agen selanjutnya agen akan memesan ticket sesuai dengan pesanan lalu oleh sistem dievaluasi apakah pesanan bisa diteruskan apabila pesanan sudah dikonfirmasi oleh admin maka no seri ticket akan dikirimkan oleh sistem kepada agen untuk diberikan pada pembeli. Fitur yang ada pada sistem e ticketing ini adalah halaman pemesanan, registrasi agen, pembuatan ticket, pengelolaan agen dan saldo, dan juga fitur laporan keuangan.

Perangkat lunak lainnya merupakan sistem informasi pintu air milik jasa tirta 1. Sistem ini merupakan sistem yang dapat diakses oleh publik yang artinya sistem ini bersifat umum. Sistem ini merupakan *company profile* dari jasa tirta 1 yang isinya tentang produk dan layanannya sebagai sarana penyedia air bersih di wilayah tersebut. Isi konten utama dari web ini adalah menunjukkan informasi batas ketinggian air di beberapa pintu air di wilayah Jawa Timur sebagai antisipasi banjir sedangkan yang lainnya adalah profil produk dan layanan yang dapat diberikan pada klien di bidang penyediaan air bersih di wilayahnya, juga terdapat halaman pengaduan untuk klien mengadukan masalah yang berkaitan dengan pelayanan jasa tirta 1.

4.2 Hasil lembar penilaian faktor eksponen (*Scale factor*)

Scale factor merupakan nilai pada organisasi yang mempengaruhi nilai estimasi suatu proyek sehingga nantinya dapat menentukan usaha proyek dan juga karakteristik dari proyek itu sendiri. Pada tahap mendapatkan nilai *scale factor*, penelitian ini menggunakan penilaian yang berisi faktor-faktor yang diperlukan untuk pengembangan sebuah perangkat lunak terdapat 5 faktor yaitu: *Precedentness* (PREC), *Development Flexibility* (FLEX), *Risk Resolution* (RESL), *Team Cohesion* (TEAM), *Process Maturity* (PMAT).

Rentang Nilai factor ini tersebut adalah dari skala 0, 1, 2, 3, 4, dan 5 lembar ini diberikan pada responden untuk mendapatkan nilai *scale factor*.

Responden ini berasal dari tim proyek perangkat lunak *DBA Ticketing* dan Sistem informasi pintu air yaitu seorang manajer proyek dan juga tim pengembang proyek . Untuk proyek *DBA Ticketing* tim proyek yang terlibat hanya satu orang yang merangkap manajer proyek ,programmer,dan juga desainer UI/UX. Untuk proyek Sistem Informasi Pintu Air tim proyek yang terlibat ada 2 orang ,1 orang merangkap manajer proyek sekaligus programmer sedangkan 1 orang lagi adalah desain UI/UX.karena berjumlah 2 orang maka masing-masing factor scale nya akan dikalikan 0.5 untuk mendapatkan rata-rata perhitungan.

Pada table 4.1 menunjukkan hasil lembar penilaian *scale factor* pada proyek perangkat lunak *DBA Ticketing*.

Tabel 4. 1 Scale factors Perangkat Lunak *DBA Ticketing*

No.	<i>Scale factor</i>	Responden 1 (Fathin)	Hasil Penilaian
1.	<i>PREC</i>	<i>Very High</i>	1.24
2.	<i>FLEX</i>	<i>Very High</i>	1.01
3.	<i>RESL</i>	<i>High</i>	2.83
4.	<i>TEAM</i>	<i>Very high</i>	1.10
5.	<i>PMAT</i>	<i>High</i>	3.12
Total penilaian <i>factor scale</i>		=	9.3

Dari tabel 4.1 menunjukkan bahwa untuk *scale factor* *PREC* adalah *Very High* yang berarti bahwa organisasi mempunyai pengalaman di masa lalu dengan proyek yang serupa dengan perangkat lunak *DBA Ticketing* sehingga organisasi sebagian besar familiar dengan proyek yang serupa dengan proyek perangkat lunak *DBA Ticketing*, *scale factor* *FLEX* adalah *very high* artinya bahwa klien dapat secara lebih menyesuaikan diri dalam menentukan dan juga mengkomunikasikan permintaan kebutuhan perangkat lunak *DBA Ticketing* kepada tim pengembang, untuk *scale factor* *RESL* adalah *High* yang berarti bahwa manajemen resiko yang dimiliki oleh organisasi dalam menyelesaikan proyek perangkat lunak *DBA Ticketing* secara umum sudah sesuai dengan *risk management plan* adalah sebesar 75%(*generally*), untuk *scale factor* *TEAM* yaitu *very high* yang berarti bahwa tim pengembang bekerjasama dan berkomunikasi dalam pengembangan proyek perangkat lunak *DBA Ticketing* memiliki interaksi kooperatif yang tinggi dan *scale factor* *PMAT* adalah *High* yang artinya adalah kematangan proses organisasi pada pengembangan proyek perangkat lunak *DBA Ticketing* sudah sampai pada level 3 yaitu sampai pada *defined level*, organisasi sudah memiliki proses dan kegiatan rekayasa perangkat lunak *DBA Ticketing* (Analisis Kebutuhan perangkat lunak, Development, maintenance) sudah didokumentasikan dan diintegrasikan ke dalam proses pengembangan proyek perangkat lunak *DBA Ticketing* yang standar digunakan dalam organisasi.

Keseluruhan proyek dikembangkan dan dibuat menggunakan standar yang telah diakui di dalam organisasi.

Sedangkan pada tabel 4.2 menunjukkan hasil lembar penilaian *scale factor* pada proyek perangkat lunak Sistem Informasi Pintu Air

Tabel 4. 2 Scale factors Perangkat Lunak Sistem Informasi Pintu Air

<i>Scale factor</i>	Responden 1 (Yusak)	Responden 2 (Reno)	Hasil Penilaian
PREC	<i>Very High</i>	<i>Very high</i>	1.24
FLEX	<i>Nominal</i>	<i>Very high</i>	$(3.04 \times 0.5) + (1.01 \times 0.5) = 2.02$
RESL	<i>Very High</i>	<i>Very high</i>	1.41
TEAM	<i>Very High</i>	<i>Very high</i>	1.10
PMAT	<i>Very High</i>	<i>Nominal</i>	$(1.56 \times 0.5) + (3.12 \times 0.5) = 2.34$
Total penilaian <i>factor scale</i>	=		8.62

Dari tabel 4.1 menunjukkan bahwa untuk *scale factor* PREC adalah *Very High* yang berarti bahwa organisasi mempunyai pengalaman di masa lalu dengan proyek yang serupa dengan perangkat lunak Sistem Informasi Pintu Air sehingga organisasi sebagian besar familiar dengan proyek yang serupa dengan proyek perangkat lunak Sistem Informasi Pintu Air, *scale factor* FLEX adalah *high* artinya bahwa klien dapat secara umum dapat menyesuaikan diri dalam menentukan dan juga mengkomunikasikan permintaan kebutuhan perangkat lunak Sistem Informasi Pintu Air kepada tim pengembang, untuk *scale factor* RESL adalah *very High* yang berarti bahwa manajemen resiko yang dimiliki oleh organisasi dalam menyelesaikan proyek perangkat lunak Sistem Informasi Pintu Air sudah sesuai dengan *risk management plan* adalah sebesar 90% (*mostly*), untuk *scale factor* TEAM yaitu *very high* yang berarti bahwa tim pengembang bekerjasama dan berkomunikasi dalam pengembangan proyek perangkat lunak Sistem Informasi Pintu Air memiliki interaksi kooperatif yang tinggi dan *scale factor* PMAT adalah *High* yang artinya adalah kematangan proses organisasi pada pengembangan proyek perangkat lunak Sistem Informasi Pintu Air sudah sampai pada level 3 yaitu sampai pada *defined* level, organisasi sudah memiliki proses dan kegiatan rekayasa perangkat lunak Sistem Informasi Pintu Air (Analisis Kebutuhan perangkat lunak, Development, maintenance) sudah didokumentasikan dan diintegrasikan ke dalam proses pengembangan proyek perangkat lunak Sistem Informasi Pintu Air yang standar digunakan dalam organisasi. Keseluruhan

proyek dikembangkan dan dibuat menggunakan standar yang telah diakui di dalam organisasi.

4.3 Hasil lembar penilaian *Effort multipliers*

Sama dengan hasil penilaian pada *scale factors* untuk penilaian *Effort multipliers* pada penelitian ini menggunakan lembar penilaian yang berisi faktor-faktor penggerak biaya yang diperlukan untuk pengembangan sebuah perangkat lunak. Ada 17 faktor penggerak biaya untuk sub-model Post-architecture yaitu RELY(Required Software Reliability), DATA(Database Size), CPLX(Product Complexity), DOCU(Documentation Match To Life-Cycle Needs), ACAP (Analyst Capability), PCAP (Programmer Capability), PCON (Personel Continuity), RUSE (Developed for Reusability), TIME (Execution Time Constraint), STOR (Main Storage Constraint), PVOL (Platform Volatility), APEX(Applications Experience), PLEX(Platform Experience), LTEX(Language and Tool Experience), TOOL (Use of Software Tools), SITE (Multisite Development), dan SCED (required Development Schedule). Rentang dari nilai factor-faktor tersebut adalah *Very Low (VL)*, *Low (L)*, *Nominal (N)*, *High (H)*, *Very High (VH)*, *Extra high (EH)*.

Lembar penilaian ini akan diberikan pada responden untuk mendapatkan nilai *Effort Multiplier*. Responden ini berasal dari tim proyek perangkat lunak *DBA Ticketing* dan Sistem Informasi Pintu Air yaitu seorang manajer proyek dan juga tim pengembang proyek . Untuk proyek *DBA Ticketing* tim proyek yang terlibat hanya satu orang yang merangkap manajer proyek ,programmer,dan juga desainer UI/UX. Untuk proyek Sistem Informasi Pintu Air tim proyek yang terlibat ada 2 orang ,1 orang merangkap manajer proyek sekaligus programmer sedangkan 1 orang lagi adalah desain UI/UX.karena berjumlah 2 orang maka masing-masing factornya akan dikalikan 0.5 untuk mendapatkan rata-rata perhitungan. Pada tabel 4.3 menunjukkan hasil lembar penilaian *scale factor* pada proyek perangkat lunak *DBA Ticketing*.

Tabel 4. 3 *Effort multipliers* Proyek Perangkat lunak *DBA Ticketing*

<i>Effort multipliers</i>	Responden 1 (Fathin)	Hasil penilaian
RELY	<i>High</i>	1.10
DATA	<i>Very High</i>	1.28
CPLX	<i>Low</i>	0.87
DOCU	<i>Very low</i>	0.91
RUSE	<i>Nominal</i>	1.00
ACAP	<i>Very low</i>	1.19
PCON	<i>Very Low</i>	1.29
TIME	<i>Nominal</i>	1.00

<i>Effort multipliers</i>	Responden 1 (Fathin)	Hasil penilaian
STOR	<i>Nominal</i>	1.00
PVOL	<i>Very high</i>	1.30
PCAP	<i>Very low</i>	1.34
APEX/AEXP	<i>Nominal</i>	1.00
PLEX/PEXP	<i>Very Low</i>	1.19
LTEX	<i>Very Low</i>	1.20
TOOL	<i>High</i>	0.90
SITE	<i>High</i>	0.93
SCED	<i>Very High</i>	1.00
Total penilaian effort multipliers	=	18.5

Dari tabel diatas menunjukkan bahwa EM RELY adalah high berarti bahwa apabila terjadi kegagalan dalam sistem perangkat lunak *DBA Ticketing* dalam menjalankan fungsi selama periode tertentu maka akan mempengaruhi kerugian secara finansial, faktor DATA adalah Very high yakni berarti bahwa database yang digunakan dalam sistem adalah DB/P >1000, CPLX adalah low yang berarti ada beberapa error saat sistem berjalan, faktor DOCU adalah Very low yang berarti bahwa hanya beberapa siklus hidup perangkat lunak yang telah didokumentasikan, faktor RUSE adalah Nominal yang berarti bahwa dokumentasi dari proyek perangkat lunak *DBA Ticketing* akan digunakan dalam proyek lain, faktor ACAP dan PCON mendapatkan rating very low yang berarti bahwa 10 proyek terakhir analisis proyek ini pernah berpartisipasi/digunakan setidaknya sebanyak 2 kali dan pergantian personel tiap tahunnya pada proyek adalah 48%, factor TIME dan STOR mendapat rating nominal yang berarti bahwa waktu eksekusi yang diharapkan pada sistem perangkat lunak adalah sebanyak 50% dari waktu eksekusi yang tersedia dan penyimpanan utama yang digunakan adalah 50% dari total penyimpanan yang disediakan, Faktor PVOL memiliki rating very high yang berarti bahwa terjadi perubahan besar pada hardware dan software dalam 6 bulan dan perubahan kecil pada setiap minggunya, faktor PCAP didapatkan rating very low yakni bahwa dalam 10 proyek terakhir, programmer setidaknya pernah ikut berpartisipasi sebanyak 2 kali dalam proyek, factor APEX/AEXP mendapatkan rating nominal yang berarti bahwa pengalaman tim dalam mengembangkan sistem perangkat lunak sudah selama 1 tahun, faktor PEXP/PLEX dan LTEX mendapat rating very low yang berarti bahwa terkait pemahaman tim dalam menggunakan platform, interface, database, jaringan, dan middleware sudah selama 6 bulan dan pengalaman tim proyek dalam menggunakan Bahasa pemrograman yang digunakan dalam mengembangkan system perangkat lunak sudah selama 6 bulan, sedangkan untuk TOOL dan SITE

mendapatkan rating high yang berarti bahwa tim proyek memiliki kemampuan kuat dan matang dalam penggunaan tools pengembangan perangkat lunak dan juga dalam berkomunikasi sudah melalui *wide band electronic communication*, sedangkan factor SCED mendapatkan rating very high yang berarti jadwal yang terkompres sebesar 160% dari jadwal yang telah ditentukan.

Tabel 4. 4 Effort multipliers Proyek Perangkat Lunak Sistem Informasi Pintu Air

Effort multipliers	Responden 1 (Yusak)	Responden 2 (Reno)	Hasil penilaian
RELY	<i>Very Low</i>	<i>Very Low</i>	0,82
DATA	<i>High</i>	<i>Nominal</i>	$(1,14 \times 0,5) + (1,00 \times 0,5) = 1,07$
CPLX	<i>Nominal</i>	<i>Very high</i>	$(1,00 \times 0,5) + (1,34 \times 0,5) = 1,17$
DOCU	<i>Nominal</i>	<i>Nominal</i>	1,00
RUSE	<i>Low</i>	<i>High</i>	$(0,95 \times 0,5) + (1,15 \times 0,5) = 1,05$
ACAP	<i>Nominal</i>	<i>Very High</i>	$(1,00 \times 0,5) + (0,71 \times 0,5) = 0,85$
PCON	<i>Very low</i>	<i>Nominal</i>	$(1,29 \times 0,5) + (1,00 \times 0,5) = 1,14$
TIME	<i>Nominal</i>	<i>Nominal</i>	1,00
STOR	<i>Nominal</i>	<i>Nominal</i>	1,00
PVOL	<i>Low</i>	<i>Low</i>	0,87
PCAP	<i>Nominal</i>	<i>Nominal</i>	1,00
APEX/AEXP	<i>High</i>	<i>High</i>	0,88
PLEX/PEXP	<i>High</i>	<i>High</i>	0,91
LTEX	<i>High</i>	<i>Nominal</i>	$(0,91 \times 0,5) + (1,00 \times 0,5) = 0,95$

<i>Effort multipliers</i>	Responden 1 (Yusak)	Responden 2 (Reno)	Hasil penilaian
TOOL	<i>High</i>	<i>High</i>	0,90
SITE	<i>High</i>	<i>Extra High</i>	(0,93x0 ,5) +(0,86 x 0,5)=0,89
SCED	<i>Nominal</i>	<i>Nominal</i>	1,00
Total penilaian effort multipliers	=		16.5

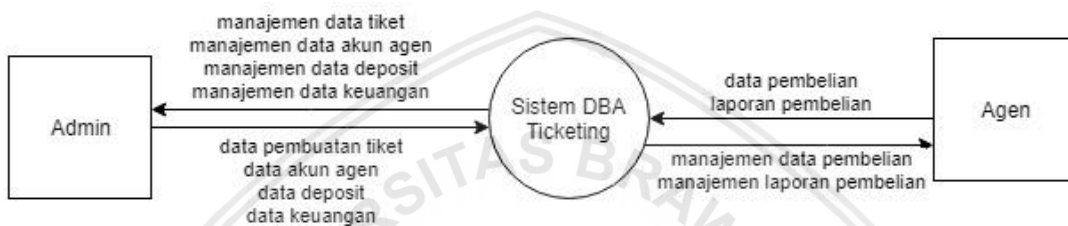
Dari tabel 4.4 diatas menunjukkan bahwa EM RELY adalah very low berarti bahwa apabila terjadi kegagalan dalam sistem perangkat lunak Sistem Informasi Pintu Air dalam menjalankan fungsi selama periode tertentu maka hanya akan berdampak pada ketidaknyamanan pengguna, faktor DATA adalah high yakni berate bahwa database yang digunakan dalam sistem adalah $100 \leq D/P < 1000$, CPLX adalah high yang berarti program berjalan sesuai dengan baik dan tidak ada error saat sistem berjalan, faktor DOCU adalah nominal yang berarti bahwa siklus hidup perangkat lunak yang telah didokumentasikan telah sesuai, faktor RUSE adalah high yang berarti bahwa dokumentasi dari proyek perangkat lunak Sistem Informasi Pintu Air akan digunakan dalam program lain, faktor ACAP mendapatkan rating high yang berarti bahwa 10 proyek terakhir analisis proyek ini pernah berpartisipasi/digunakan setidaknya sebanyak 8 kali, PCON mendapatkan rating low yang berarti bahwa pergantian personel tiap tahunnya pada proyek adalah 24%, factor TIME dan STOR mendapat rating nominal yang berate bahwa waktu eksekusi yang diharapkan pada system perangkat lunak adalah sebanyak 50% dari waktu eksekusi yang tersedia dan penyimpanan utama yang digunakan adalah 50% dari total penyimpanan yang disediakan, Faktor PVOL memiliki rating low yang berarti bahwa terjadi perubahan besar pada hardware dan software dalam 12 bulan dan perubahan kecil pada setiap 1 bulan, faktor PCAP didapatkan rating nominal yakni bahwa dalam 10 proyek terakhir, programmer setidaknya pernah ikut berpartisipasi sebanyak 6 kali dalam proyek, factor APEX/AEXP mendapatkan rating high yang berarti bahwa pengalaman tim dalam mengembangkan sistem perangkat lunak sudah selama 3 tahun , faktor PEXP/PLEX mendapat rating high yang berarti bahwa terkait pemahaman tim dalam menggunakan platform,interface, database, jaringan, dan middleware sudah selama 3 tahun, LTEX mendapat rating nominal yang berarti bahwa pengalaman tim proyek dalam menggunakan Bahasa pemrograman yang digunakan dalam mengembangkan system perangkat lunak sudah selama 1 tahun, sedangkan untuk TOOL dan SITE mendapatkan rating high yang berarti bahwa tim proyek memiliki kemampuan kuat dan matang dalam penggunaan tools pengembangan perangkat lunak dan juga dalam berkomunikasi sudah melalui *wide band electronic communication* , sedangkan factor SCED mendapatkan rating nominal yang berarti jadwal yang terkompres sebesar 100% dari jadwal yang telah ditentukan.

4.4 Data Flow Diagram (DFD)

Dalam perhitungan *COCOMO II* ada dua cara dalam menentukan nilai *size Kilo Source Lines Of Code (KSLOC)*, yang pertama adalah dengan cara menghitung *source line of code lines* dan yang kedua adalah dengan cara menganalisis kebutuhan fungsional sistem yang direpresentasikan ke dalam Data Flow Diagram (DFD) untuk nantinya mempermudah analisis ke dalam *function point (FP)* (Neelam Bawane' Singhal & C. Srikrishna,2008)

4.4.1 DFD level 0 Perangkat Lunak *DBA Ticketing*

DFD level 0 DBA ticketing



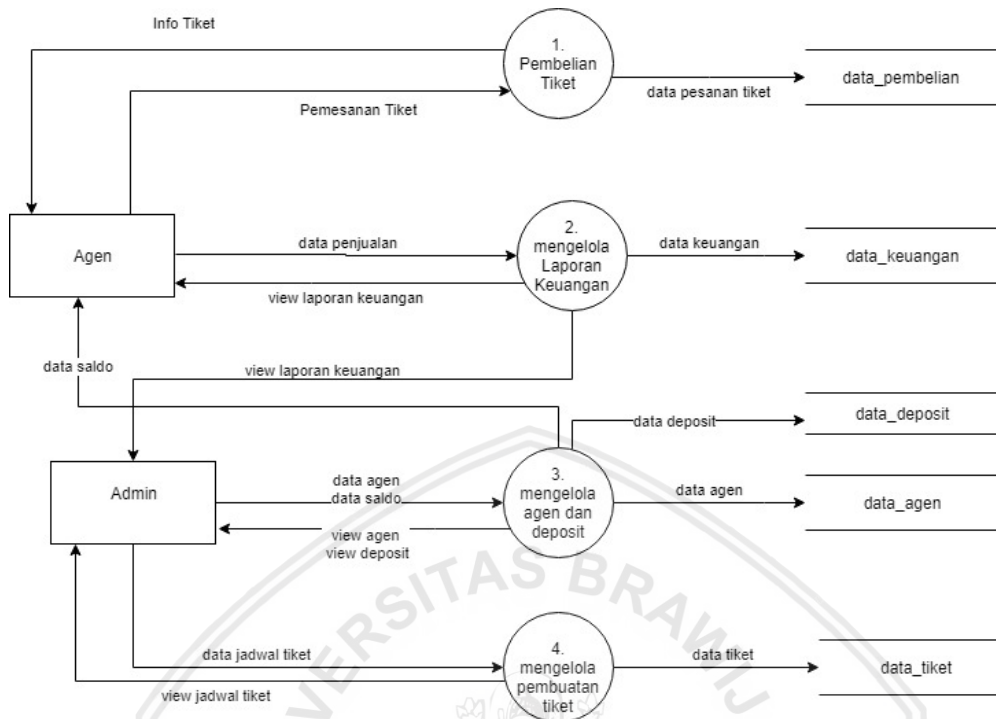
Gambar 4. 1 DFD level 0 *DBA Ticketing*

Gambar DFD level 0 perangkat lunak *DBA Ticketing* menggambarkan proses di dalam sistem sebagai lingkaran yang tunggal dengan mendapat inputan dan juga output data yang ditunjukkan dengan arah panah masuk untuk inputan dan arah panah keluar sebagai hasil output terhadap entitas luar yang digambarkan sebagai persegi (Sommerville, 2010). Entitas luar pada sistem ini ada 2 yaitu admin dan agen.

Admin system dapat melakukan log in, memasukkan, mengedit dan juga menghapus data tiket, data laporan keuangan, data deposit, dan juga data akun agen. Output yang dihasilkan dari admin adalah data tiket baru, data registrasi agen, dan data laporan (jumlah penjualan dan rekap laporan keuangan).

Agen dapat melakukan memasukkan, mengedit dan juga menghapus pesanan tiket, Melakukan deposit, dan Membuat laporan penjualan. Output yang dihasilkan oleh Agen adalah no seri tiket pesanan, data saldo, dan juga data rekap penjualan tiket.

4.4.2 DFD level 1 Perangkat Lunak DBA Ticketing



Gambar 4. 2 DFD level 1 DBA Ticketing

Gambar DFD level 1 perangkat lunak *DBA Ticketing* ini memiliki 4 proses, yakni : pembelian tiket, mengelola laporan keuangan, mengelola agen dan deposit, dan mengelola pembuatan tiket.

Proses untuk pembelian tiket mendapat input dari agen untuk proses pemesanan tiket. Kemudian data akan dicek dan juga diproses ke data store *data_pembelian*.

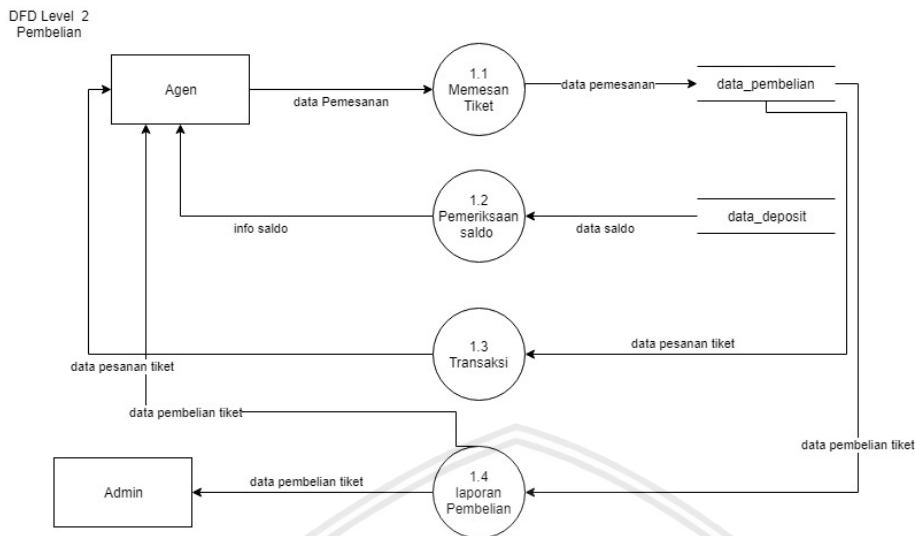
Proses untuk mengelola laporan keuangan mendapat input data penjualan dari agen untuk proses rekap data laporan keuangan. Kemudian data akan dicek dan juga diproses ke data store *data_keuangan*.

Proses untuk mengelola agen dan deposit mendapat input dari agen untuk proses data deposit dan mendapatkan input dari admin untuk data registrasi agen dan *data_saldo*. Kemudian data akan diproses ke data store *data_agen*

Proses untuk pembuatan tiket mendapat input dari admin untuk proses pembuatan tiket. Kemudian data akan dicek dan juga diproses ke data store *data_tiket*.



4.4.3 DFD level 2 Mengelola Pemesanan *DBA Ticketing*

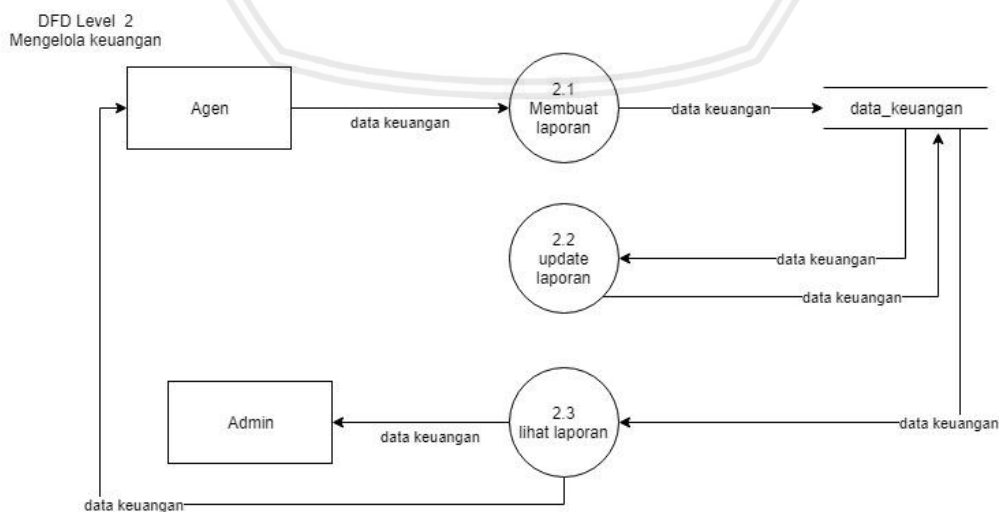


Gambar 4. 3 DFD level 2 pemesanan *DBA Ticketing*

Keterangan Gambar:

Pada proses memesan tiket terdapat input data pemesanan dan menghasilkan output data pemesanan tiket dan disimpan dalam data store pemesanan. Pada proses pemeriksaan saldo terdapat input data saldo dari data store deposit untuk mengecek saldo dari agen dan output yang dihasilkan berupa info saldo. Pada proses transaksi inputan dari data store pembelian dan menghasilkan output data tiket pesanan. Pada proses laporan pembelian mendapat inputan dari data store pembelian dan menghasilkan output data pembelian.

4.4.4 DFD level 2 mengelola laporan keuangan *DBA Ticketing*



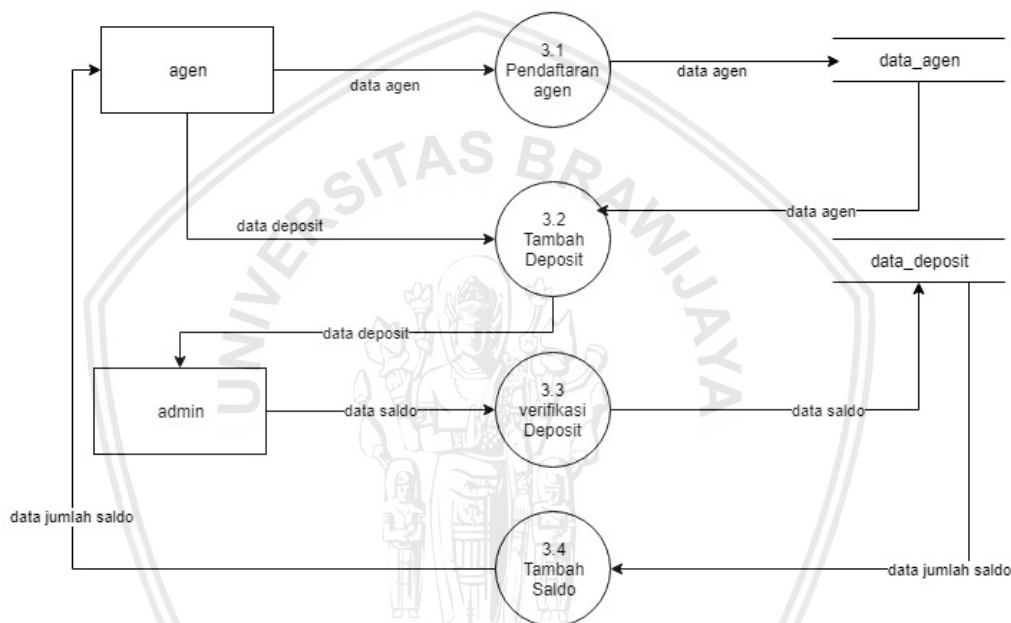
Gambar 4. 4 DFD level 2 mengelola laporan keuangan *DBA Ticketing*

Keterangan Gambar:

Pada proses membuat laporan mendapat input data transaksi lalu disimpan dalam data store keuangan. Pada proses rekap laporan mendapat inputan dari data store keuangan dan menghasilkan output data rekap laporan. Pada proses lihat laporan mendapat input dari data store keuangan lalu menghasilkan output data keuangan laporan.

4.4.5 DFD level 2 mengelola agen dan deposit *DBA Ticketing*

DFD Level 2
Pengelolaan agen



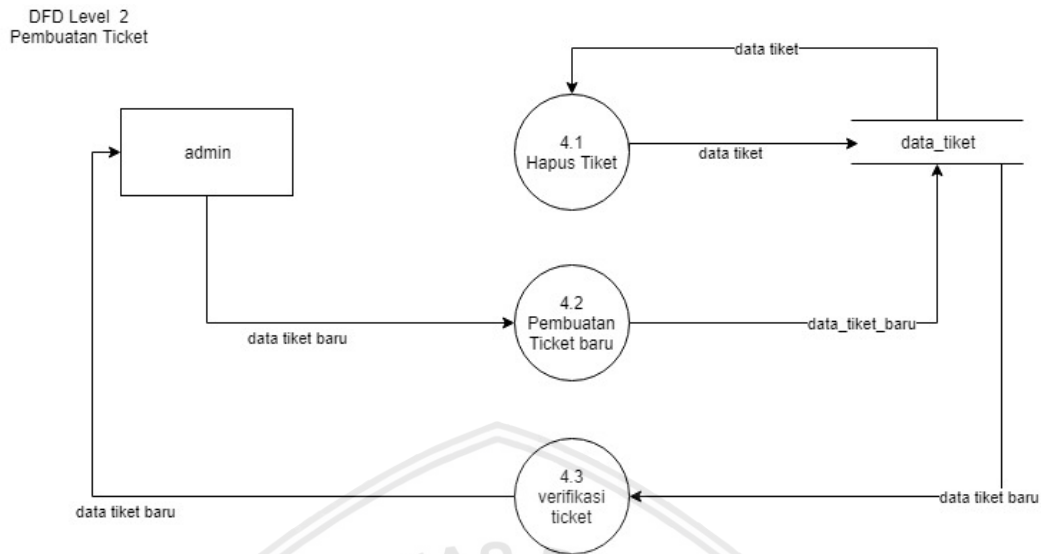
Gambar 4. 5 DFD level 2 mengelola agen dan deposit *DBA Ticketing*

Keterangan Gambar:

Pada proses pendaftaran agen mendapat inputan data agen baru lalu disimpan dalam data store agen. Pada proses tambah deposit mendapat inputan data agen dan jumlah deposit menghasilkan output data deposit . Pada proses verifikasi deposit mendapat inputan data saldo lalu disimpan dalam data store deposit. Pada proses tambah saldo mendapat inputan data jumlah saldo dari data store dan menghasilkan output data jumlah saldo.



4.4.6 DFD level 2 pembuatan tiket *DBA Ticketing*



Gambar 4. 6 DFD level 2 pembuatan tiket *DBA Ticketing*

Keterangan Gambar:

Pada proses hapus tiket mendapat input data tiket dari data store data_ticket dan menghasilkan data ticket. Pada proses pembuatan ticket mendapat input berupa data tiket baru dan disimpan dalam store data_ticket. Pada proses verifikasi tiket mendapat input data no ticket baru dan menghasilkan output data no tiket baru.

4.4.7 DFD level 0 Perangkat Lunak Sistem informasi Pintu Air

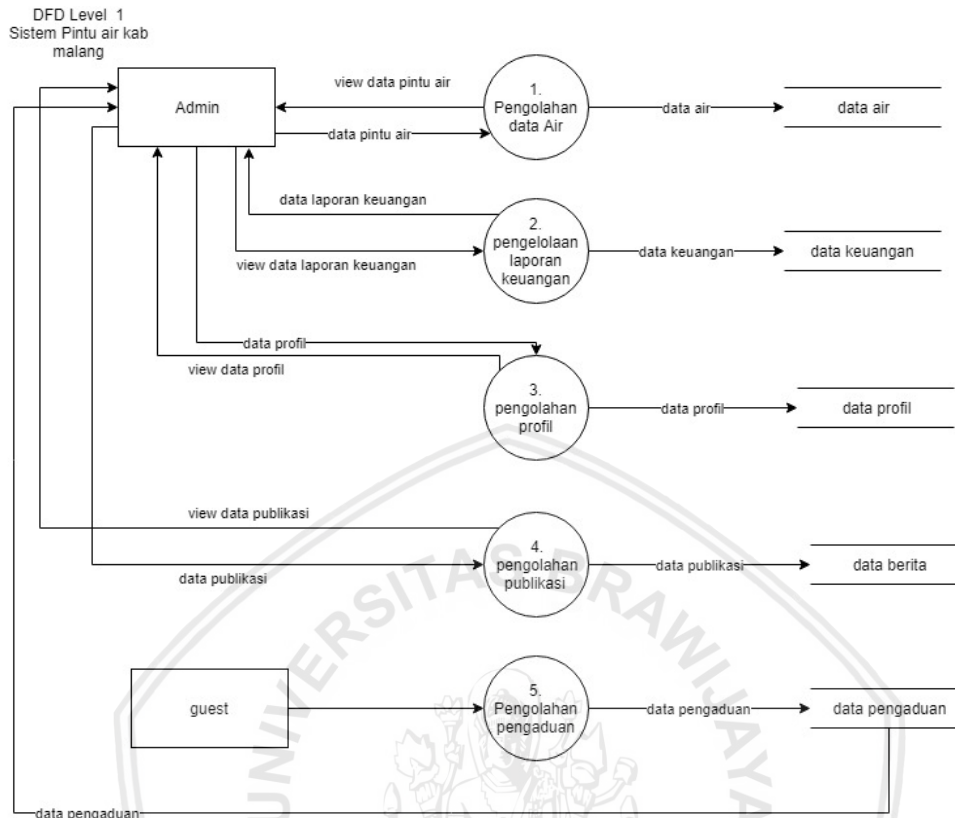


Gambar 4. 7 DFD level 0 Perangkat Lunak Sistem informasi Pintu Air

Admin sistem dapat melakukan memasukkan, mengedit dan juga menghapus data pintu air, data laporan keuangan, data profil, dan juga data publikasi. Output yang dihasilkan dari admin adalah data pintu air, data laporan keuangan, data profil, data pengaduan dan juga data publikasi.

pengaduan mendapat input data pengaduan dan menghasilkan data pengaduan. Pada proses lihat data pengaduan mendapat input dari store berupa data pengaduan dan menghasilkan output data pengaduan.

4.4.8 DFD level 1 Perangkat Lunak Sistem informasi Pintu Air



Gambar 4. 8 DFD level 1 Perangkat Lunak Sistem informasi Pintu Air

Gambar DFD level 1 perangkat lunak Sistem informasi Pintu Air ini memiliki 4 proses, yakni : Pengolahan data air, Pengelolaan laporan keuangan, Pengolahan profil, dan Pengolahan publikasi.

Proses untuk pengolahan data air mendapat input dari admin berupa data air. Kemudian data akan dicek dan juga diproses ke data store data air.

Proses untuk Pengelolaan laporan keuangan mendapat input data laporan keuangan dari admin berupa data laporan keuangan. Kemudian data akan dicek dan juga diproses ke data store data keuangan.

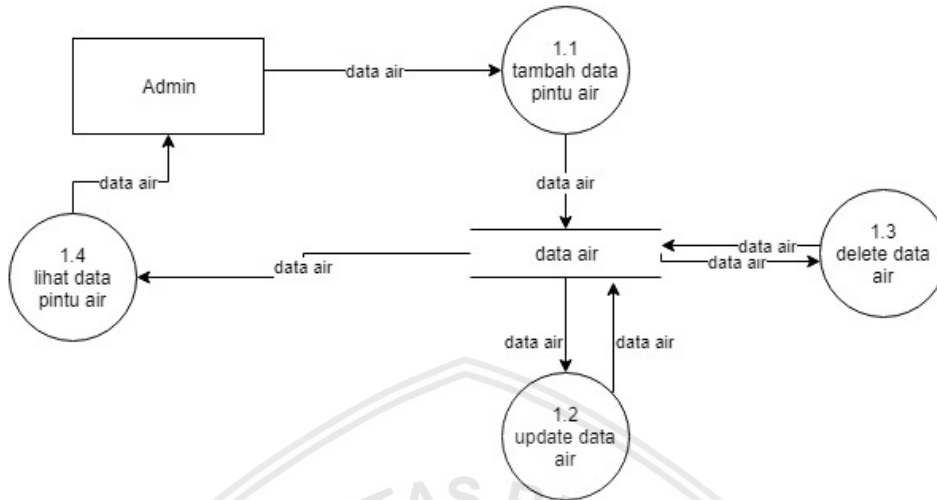
Proses untuk pengolahan data profil mendapat input dari admin berupa data profil. Kemudian data akan diproses ke data store data profil

Proses untuk pengolahan publikasi mendapat input dari admin berupa data publikasi. Kemudian data akan dicek dan juga diproses ke data store data berita.

Proses untuk pengolahan data pengaduan mendapat input dari admin berupa data pengaduan. Kemudian data akan diproses ke data store data pengaduan.

4.4.9 DFD level 2 Mengelola pintu air

DFD Level 2
mengelola pintu air



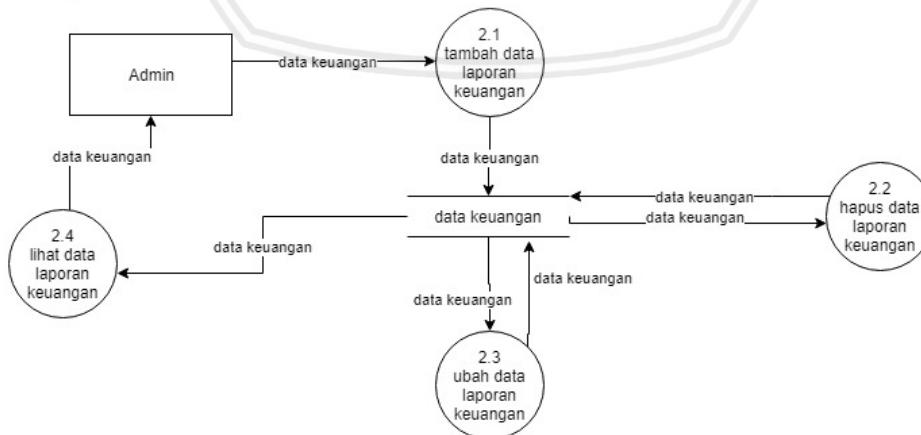
Gambar 4. 9 DFD level 2 mengelola pintu air

Keterangan Gambar :

Pada proses tambah data pintu air mendapat input data air dan menghasilkan data air. Pada proses ubah data pintu air mendapat input data air dan dapat memperbaharui data air kemudian disimpan dalam data store data air. Pada proses hapus data air mendapat input dari data store data air dan menghasilkan data air. Pada proses lihat data pintu air mendapat input dari store berupa data air dan menghasilkan output data air.

4.4.10 DFD level 2 mengelola laporan keuangan

DFD Level 2
mengelola laporan keuangan



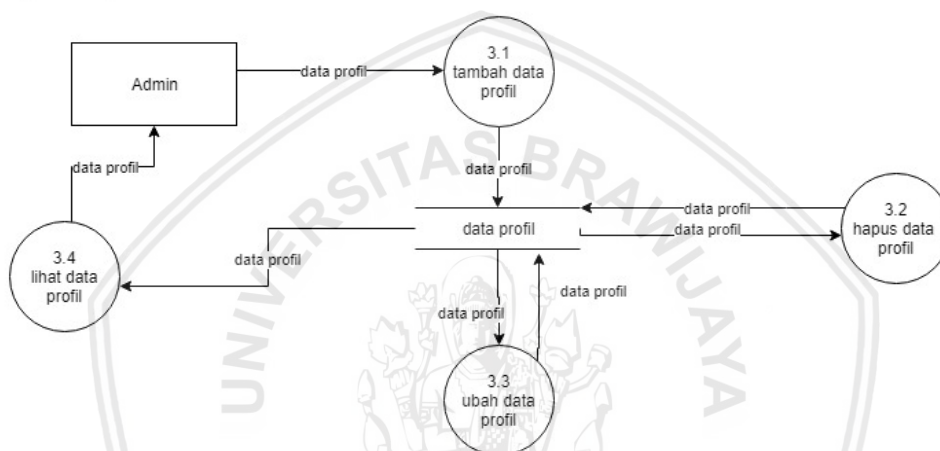
Gambar 4. 10 DFD level 2 mengelola laporan keuangan

Keterangan Gambar :

Pada proses tambah data laporan keuangan mendapat input data keuangan dan menghasilkan data keuangan. Pada proses ubah data laporan keuangan mendapat input data keuangan dan dapat memperbaharui data air kemudian disimpan dalam data store data air. Pada proses hapus data laporan keuangan mendapat input data keuangan dan menghasilkan data keuangan. Pada proses lihat data laporan keuangan mendapat input dari store berupa data keuangan dan menghasilkan output data keuangan.

4.4.11 DFD level 2 mengelola profil

DFD Level 2
mengelola data profil



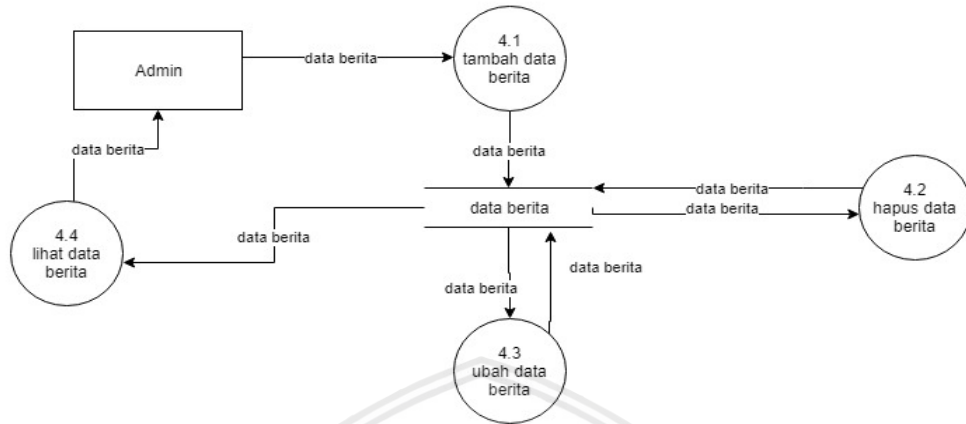
Gambar 4. 11 DFD level 2 mengelola profil

Keterangan Gambar :

Pada proses tambah data profil mendapat input data profil dan menghasilkan data profil. Pada proses ubah data profil mendapat input data profil dan dapat memperbaharui data profil kemudian disimpan dalam data store data profil. Pada proses hapus data laporan profil mendapat input data profil dan menghasilkan data profil. Pada proses lihat data profil mendapat input dari store berupa data profil dan menghasilkan output data profil.

4.4.12 DFD level 2 mengelola berita

DFD Level 2
mengelola berita



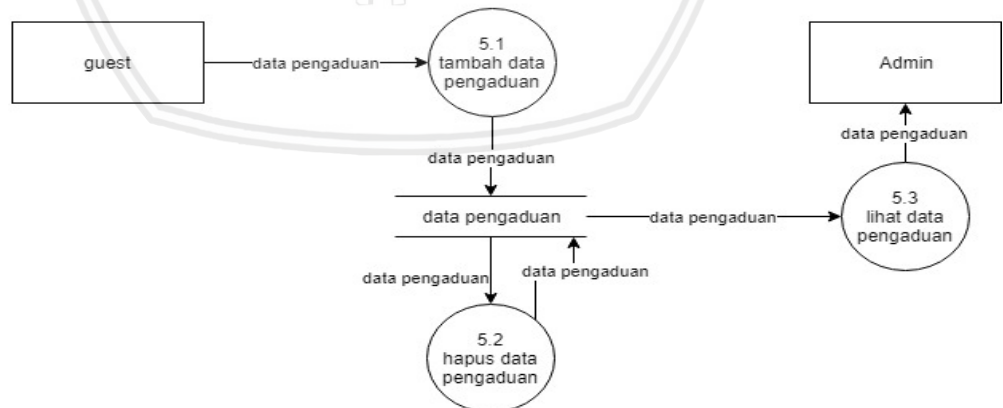
Gambar 4. 12 DFD level 2 mengelola berita

Keterangan Gambar :

Pada proses tambah data berita mendapat input data berita dan menghasilkan data berita. Pada proses ubah data berita mendapat input data berita dan dapat memperbaharui berita kemudian disimpan dalam data store data berita. Pada proses hapus data berita mendapat input data berita dan menghasilkan data berita. Pada proses lihat berita mendapat input dari store berupa data berita dan menghasilkan output data berita.

4.4.13 DFD level 2 mengelola pengaduan

DFD Level 2
mengelola pengaduan



Gambar 4. 13 DFD level 2 mengelola pengaduan

Keterangan Gambar :

Pada proses tambah data pengaduan mendapat input data pengaduan dan menghasilkan data pengaduan. Pada proses hapus data pengaduan



mendapat input data pengaduan dan menghasilkan data pengaduan. Pada proses lihat data pengaduan mendapat input dari store berupa data pengaduan dan menghasilkan output data pengaduan.



BAB 5 PEMBAHASAN

5.1 Skala Proyek

Skala proyek yang digunakan untuk mengetahui seberapa besar ukuran proyek yang akan dikerjakan sehingga nantinya akan mampu memperkirakan besarnya biaya, waktu maupun SDM yang dibutuhkan di dalam pengerjaan proyek. Untuk ini CV. Profile Image Studio menggunakan pedoman Agile development untuk mengukur skala proyek. Pada Agile Development ada 3 jenis skala proyek berdasarkan yang ada pada tim proyek. Dengan jumlah SDM (Sumber Daya Manusia) +/- 2 orang, yaitu: (a.) kategori skala kecil, jumlah 1 tim, (b.) kategori skala besar, jumlah 2-9 tim, (c.) kategori skala sangat besar, jumlah lebih dari 9 tim. Nantinya, masing-masing proyek perangkat lunak yang ada pada penelitian ini memiliki 1-2 orang pada 1 tim maka dapat dikategorikan sebagai proyek skala kecil.

5.2 Analisis Function Point

Analisis function point digunakan untuk mengetahui seberapa besar kompleksitas yang ada pada sistem. Function Point (FP) dapat mengukur proyek perangkat lunak dengan cara mengkuantifikasi fungsi-fungsi yang terdapat pada sistem berupa pemrosesan informasi yang berkaitan dengan masuk data, keluaran data, dan tipe berkas data atau kontrol eksternal berupa *Internal Logical File (ILF)*, *External Interface File (EIF)*, *External Output (EO)*, *External Input (EI)*, *External Inquiry (EQ)*. Syarat penentuan dari kelima factor tersebut adalah: (a). ILF berkaitan erat dengan kumpulan data yang disimpan melalui proses dasar dalam bentuk table yang tersimpan dalam batas perangkat lunak itu sendiri contohnya adalah data table yang ada dalam database perangkat lunak, (b). EIF berkaitan dengan kumpulan data yang disimpan dalam ILF perangkat lunak lain, (c). EO berkaitan erat dengan hasil keluaran data dari proses masukan data dapat berupa tampilan maupun hasil cetak, (d). EI merupakan inputan berupa insert, update, maupun delete data, (e). EQ merupakan inputan yang nantinya akan menghasilkan respon output secara seketika contoh fitur *Search By*. yang akan dijelaskan pada subbab berikut

5.2.1 Analisis Function Point Perangkat Lunak *DBA Ticketing*

Dari hasil dekomposisi DFD level 0 perangkat lunak *DBA Ticketing* menghasilkan DFD level 1 proses mengelola pembelian. Untuk hasilnya akan dijelaskan oleh tabel 5.1 berikut.

Tabel 5. 1 Analisis Function Point Perangkat Lunak *DBA Ticketing* proses mengelola pembelian

<i>Internal logical File (ILF)</i>	<i>External Interface File (EIF)</i>	<i>External Input (EI)</i>	<i>External Inquiry (EQ)</i>	<i>External Output (EO)</i>
Data Pemesanan	-	Data pemesanan	-	View data pemesanan
Data laporan transaksi		Update pemesanan Hapus pemesanan Data transaksi Update transaksi Hapus transaksi		View Data laporan transaksi

Hasil proses analisis *function point* perangkat lunak *DBA Ticketing* proses mengelola pembelian menunjukkan bahwa terdapat 2 ILF, 6 EI, dan 2 EO. Untuk EIF tidak ditemukan dikarenakan tidak ada data yang diambil dari database perangkat lunak lain dan EQ tidak ditemukan karena tidak ditemukan hasil keluaran yang seketika dalam proses mengelola pembelian.

Selanjutnya dari hasil dekomposisi DFD level 0 perangkat lunak *DBA Ticketing* menghasilkan DFD level 1 untuk proses mengelola laporan keuangan. Untuk hasilnya akan dijelaskan oleh tabel 5.2 berikut.

Tabel 5. 2 Analisis Function Point Perangkat Lunak *DBA Ticketing* proses mengelola laporan keuangan

<i>Internal logical File (ILF)</i>	<i>External Interface File (EIF)</i>	<i>External Input (EI)</i>	<i>External Inquiry (EQ)</i>	<i>External Output (EO)</i>
Data Keuangan	-	Data keuangan	-	View laporan keuangan
Data pembelian		Update laporan keuangan		View data pembelian

<i>Internal logical File (ILF)</i>	<i>External Interface File (EIF)</i>	<i>External Input (EI)</i>	<i>External Inquiry (EQ)</i>	<i>External Output (EO)</i>
		Delete laporan keuangan Data pembelian Update Data pembelian Delete Data pembelian		

Hasil proses analisis *function point* perangkat lunak *DBA Ticketing* proses mengelola laporan keuangan menunjukkan bahwa terdapat 2 ILF, 6 EI, dan 2 EO. Untuk EIF tidak ditemukan dikarenakan tidak ada data yang diambil dari database perangkat lunak lain dan EQ tidak ditemukan karena tidak ditemukan hasil keluaran yang seketika dalam proses mengelola laporan keuangan.

Selanjutnya dari hasil dekomposisi DFD level 0 perangkat lunak *DBA Ticketing* menghasilkan DFD level 1 proses Mengelola Agen dan Deposit. Untuk hasilnya akan dijelaskan oleh tabel 5.3 berikut.

Tabel 5. 3 Analisis Function Point Perangkat Lunak *DBA Ticketing* proses mengelola agen dan deposit

<i>Internal logical File (ILF)</i>	<i>External Interface File (EIF)</i>	<i>External Input (EI)</i>	<i>External Inquiry (EQ)</i>	<i>External Output (EO)</i>
Data agen Data saldo	-	Data agen Update agen Delete agen Data saldo Update data saldo	-	View data agen View data saldo

Hasil proses analisis *function point* perangkat lunak *DBA Ticketing* proses mengelola agen dan deposit menunjukkan bahwa terdapat 2 ILF, 5 EI, dan 2 EO. Untuk EIF tidak ditemukan dikarenakan tidak ada data yang diambil dari database perangkat lunak lain dan EQ tidak ditemukan karena tidak ditemukan hasil keluaran yang seketika dalam proses mengelola laporan keuangan.

Selanjutnya dari hasil dekomposisi DFD level 0 perangkat lunak *DBA Ticketing* menghasilkan DFD level 1 proses mengelola pembuatan tiket. Untuk hasilnya akan dijelaskan oleh tabel 5.4 berikut.



Tabel 5. 4 Analisis Function Point Perangkat Lunak DBA Ticketing proses mengelola pembuatan tiket

<i>Internal logical File (ILF)</i>	<i>External Interface File (EIF)</i>	<i>External Input (EI)</i>	<i>External Inquiry (EQ)</i>	<i>External Output (EO)</i>
Data tiket	-	Data tiket Update tiket Delete tiket	-	View data tiket

Hasil proses analisis *function point* perangkat lunak DBA Ticketing proses mengelola pembuatan tiket menunjukkan bahwa terdapat 1 ILF, 3 EI, dan 1 EO. Untuk EIF tidak ditemukan dikarenakan tidak ada data yang diambil dari database perangkat lunak lain dan EQ tidak ditemukan karena tidak ditemukan hasil keluaran yang seketika dalam proses mengelola pembuatan tiket.

5.2.2 Analisis Function Point Perangkat Lunak Sistem Pintu Air

Selanjutnya dari hasil dekomposisi DFD level 0 perangkat lunak sistem pintu air menghasilkan DFD level 1 proses pengelolaan data air. Untuk hasilnya akan dijelaskan oleh tabel 5.5 berikut.

Tabel 5. 5 Analisis Function Point Perangkat Lunak Sistem Pintu Air proses pengelolaan data air

<i>Internal logical File (ILF)</i>	<i>External Interface File (EIF)</i>	<i>External Input (EI)</i>	<i>External Inquiry (EQ)</i>	<i>External Output (EO)</i>
Data pintu air Data indikator	-	Data pintu air Update data pintu air Hapus data pintu air Data indikator Update data Indikator	Search by	View data pintu air Data indikator

Hasil proses analisis *function point* perangkat lunak Sistem Pintu Air proses pengelolaan data air menunjukkan bahwa terdapat 2 ILF, 5 EI, 1 EQ, dan 2 EO. Untuk EIF tidak ditemukan dikarenakan tidak ada data yang diambil dari database perangkat lunak lain.

Selanjutnya dari hasil dekomposisi DFD level 0 perangkat lunak sistem pintu air menghasilkan DFD level 1 proses pengolahan laporan keuangan. Untuk hasilnya akan dijelaskan oleh tabel 5.6 berikut.

Tabel 5. 6 Analisis Function Point Perangkat Lunak Sistem Pintu Air proses pengolahan laporan keuangan

<i>Internal logical File (ILF)</i>	<i>External Interface File (EIF)</i>	<i>External Input (EI)</i>	<i>External Inquiry (EQ)</i>	<i>External Output (EO)</i>
Data laporan keuangan	-	Data laporan keuangan Update data laporan keuangan Delete laporan keuangan	-	View data laporan keuangan

Hasil proses analisis *function point* perangkat lunak Sistem Pintu Air proses pengolahan laporan keuangan menunjukkan bahwa terdapat 1 ILF, 3 EI, dan 1 EO. Untuk EIF tidak ditemukan dikarenakan tidak ada data yang diambil dari database perangkat lunak lain dan EQ tidak ditemukan karena tidak ditemukan hasil keluaran yang seketika dalam proses pengolahan laporan keuangan.

Selanjutnya dari hasil dekomposisi DFD level 0 perangkat lunak sistem pintu air menghasilkan DFD level 1 proses pengolahan profil. Untuk hasilnya akan dijelaskan oleh tabel 5.7 berikut.

Tabel 5. 7 Analisis Function Point Perangkat Lunak Sistem Pintu Air proses pengolahan profil

<i>Internal logical File (ILF)</i>	<i>External Interface File (EIF)</i>	<i>External Input (EI)</i>	<i>External Inquiry (EQ)</i>	<i>External Output (EO)</i>
Data profil	-	Data profil Update profil Delete profil	-	View data profil

Hasil proses analisis *function point* perangkat lunak Sistem Pintu Air proses pengolahan profil menunjukkan bahwa terdapat 1 ILF, 3 EI, dan 1 EO. Untuk EIF tidak ditemukan dikarenakan tidak ada data yang diambil dari database perangkat lunak lain dan EQ tidak ditemukan karena tidak ditemukan hasil keluaran yang seketika dalam proses pengolahan laporan keuangan.



Selanjutnya dari hasil dekomposisi DFD level 0 perangkat lunak sistem pintu air menghasilkan DFD level 1 proses pengolahan publikasi. Untuk hasilnya akan dijelaskan oleh tabel 5.8 berikut.

Tabel 5. 8 Analisis Function Point Perangkat Lunak Sistem Pintu Air proses pengolahan publikasi

<i>Internal logical File (ILF)</i>	<i>External Interface File (EIF)</i>	<i>External Input (EI)</i>	<i>External Inquiry (EQ)</i>	<i>External Output (EO)</i>
Data berita Data artikel	-	Data berita Update berita Delete berita Data artikel Update artikel Delete artikel	-	View data berita View data artikel

Hasil proses analisis *function point* perangkat lunak Sistem Pintu Air proses pengolahan publikasi menunjukkan bahwa terdapat 2 ILF, 6 EI, dan 2 EO. Untuk EIF tidak ditemukan dikarenakan tidak ada data yang diambil dari database perangkat lunak lain dan EQ tidak ditemukan karena tidak ditemukan hasil keluaran yang seketika dalam proses pengolahan publikasi.

Selanjutnya dari hasil dekomposisi DFD level 0 perangkat lunak sistem pintu air menghasilkan DFD level 1 proses pengolahan pengaduan. Untuk hasilnya akan dijelaskan oleh tabel 5.9 berikut.

Tabel 5. 9 Analisis Function Point Perangkat Lunak Sistem Pintu Air proses pengolahan pengaduan

<i>Internal logical File (ILF)</i>	<i>External Interface File (EIF)</i>	<i>External Input (EI)</i>	<i>External Inquiry (EQ)</i>	<i>External Output (EO)</i>
Data pengaduan	-	Data pengaduan Delete pengaduan	-	View data pengaduan

Hasil proses analisis *function point* perangkat lunak Sistem Pintu Air proses pengolahan pengaduan menunjukkan bahwa terdapat 1 ILF, 2 EI, dan 1 EO. Untuk EIF tidak ditemukan dikarenakan tidak ada data yang diambil dari database perangkat lunak lain dan EQ tidak ditemukan karena tidak ditemukan hasil keluaran yang seketika dalam proses pengolahan pengaduan.



5.3 Analisis Perhitungan UFP

Selanjutnya diubbab ini akan menghitung nilai UFP setelah menganalisis *function point* masing-masing fungsi. Untuk penghitungan nilai bobot kompleksitas masing-masing *function point* pada setiap proses ddihitung sesuai aturan DET (*Data Element type*),RET (*Record Element Type*), dan FTR (*Field Record Type*). Untuk penjelasan DET, RET, dan FTR adalah sebagai berikut. (a) DET adalah dapat dikenali oleh pengguna sebagai sesuatu yang unik dan tidak berulang,dikenali serta disimpan atau diambil dari ILF dan EIF , dapat berupa notifikasi,tombol dan juga pesan gagal. (b) RET adalah merupakan subgrup data yang dikenali oleh pengguna dalam *Internal Logical File* atau *External Logical File*. (c) FTR dapat berupa *ILF (Internal Logical File)* atau *EIF (External Interface File)*.

5.3.1 Analisis Perhitungan UFP Perangkat Lunak DBA Ticketing

Disini masing-masing *function point* pada setiap proses akan dinilai berdasarkan aturan DET, RET, dan FTR.

Berikut ini adalah hasil analisis DET, RET, dan FTR untuk proses mengelola pembelian yang peneliti dapatkan dari PDM. Seperti dijelaskan pada tabel 5.10 di bawah ini.

Tabel 5. 10 Analisis DET,RET, Dan FTR Perangkat Lunak DBA Ticketing proses mengelola pembelian

Fungsi	DET	RET	FTR
ILF	Id_pembelian Id_ticket Nama_pembeli No_ticket Tanggal_pesan Tanggal_beli harga judul id_kategori kategori keterangan id_laporan laporan id_keterangan	Data Pemesanan Data laporan transaksi	-
	14 DET	2 RET	0 FTR

Fungsi	DET	RET	FTR
EIF	-	-	-
	0 DET	0 RET	0 FTR
EI	Nama_pembeli Tanggal_pesanan Tanggal_beli Jumlah Judul Kategori laporan tombol error message	-	Data Pemesanan Data laporan transaksi
	9 DET	-	2 FTR
EQ	-	-	-
	0 DET	0 RET	0 FTR
EO	Nama_pembeli No_ticket Tanggal_pesanan Tanggal_beli harga judul kategori laporan	-	Data Pemesanan Data laporan transaksi
	8 DET	0 RET	2 FTR

Langkah selanjutnya adalah mendapatkan nilai UFP dari analisis DET, RET, dan FTR pada proses mengelola pembelian yang nantinya hasilnya akan dikonversikan ke dalam perhitungan nilai KSLOC. Seperti digambarkan pada tabel 5.11 dibawah ini.

Tabel 5. 11 Perhitungan UFP Perangkat Lunak DBA Ticketing pada proses mengelola pembelian

Fungsi	Jumlah	Perhitungan bobot kompleksitas			Nilai kompleksitas	Bobot Kompleksitas	UFP
		DET	RET	FTR			
ILF	2	14	2	-	Low	7	14
EIF	0	-	-	-	-	-	0
EI	6	9	-	2	Average	4	24
EQ	0	-	-	-	-	-	0
EO	2	8	-	2	Average	5	10
Total UFP							48

Selanjutnya Berikut ini adalah hasil analisis DET, RET, dan FTR untuk proses mengelola laporan keuangan yang peneliti dapatkan dari PDM. Seperti digambarkan pada tabel 5.12 dibawah ini.

Tabel 5. 12 Analisis DET,RET, Dan FTR Perangkat Lunak DBA Ticketing proses mengelola laporan keuangan

Fungsi	DET	RET	FTR
ILF	Id_laporan Id_agen Nama_pembeli Nama_agen Jumlah_pesan Jumlah_beli Total Tanggal Id_pembelian Id_keterangan_laporan	Data laporan keuangan Data pembelian	-
	11 DET	2 RET	0 FTR
EIF	-	-	-
	0 DET	0 RET	0 FTR

Fungsi	DET	RET	FTR
EI	Nama_pembeli Jumlah_pesanan Jumlah_beli Tanggal laporan Tombol Error message	-	Data laporan keuangan Data pembelian
	6 DET	-	2 FTR
EQ	-	-	-
	0 DET	0 RET	0 FTR
EO	Nama_agen Nama_pembeli Jumlah_pesanan Jumlah_beli total tanggal laporan	-	Data laporan keuangan Data pembelian
	7 DET	0 RET	2 FTR

Langkah selanjutnya adalah mendapatkan nilai UFP dari analisis DET, RET, dan FTR pada proses mengelola laporan keuangan yang nantinya hasilnya akan dikonversikan ke dalam perhitungan nilai KSLOC. Seperti dijelaskan pada tabel 5.13 di bawah ini.

Tabel 5. 13 Perhitungan UFP Perangkat Lunak DBA Ticketing pada proses mengelola laporan keuangan

Fungsi	Jumlah	Perhitungan bobot kompleksitas			Nilai kompleksitas	Bobot Kompleksitas	UFP
		DET	RET	FTR			
ILF	2	11	2	-	Low	7	14
EIF	0	-	-	-	-	-	0
EI	6	6	-	2	Average	4	24
EQ	0	-	-	-	-	-	0
EO	2	7	-	2	Average	5	10
Total UFP							48

Selanjutnya Berikut ini adalah hasil analisis DET, RET, dan FTR untuk proses mengelola laporan keuangan yang peneliti dapatkan dari PDM. Seperti dijelaskan pada tabel 5.14 di bawah ini.

Tabel 5. 14 Analisis DET,RET, Dan FTR Perangkat Lunak DBA Ticketing proses mengelola mengelola agen dan saldo

Fungsi	DET	RET	FTR
ILF	Id_agen Nama_agen No_identitas Alamat Id_jenis_kelamin Jenis_kelamin No_hp No_telp Id_saldo Jumlah_saldo keterangan	Data agen Data saldo	-
	11 DET	2 RET	0 FTR
EIF	-	-	-



Fungsi	DET	RET	FTR
	0 DET	0 RET	0 FTR
EI	Nama_agen No_identitas Jenis_kelamin alamat No_HP No_telp Jumlah_saldo Tombol Error message	-	Data agen Data saldo
	9 DET	-	2 FTR
EQ	-	-	-
	DET	0 RET	0 FTR
EO	Nama_agen No_identitas Jenis_kelamin alamat No_HP No_telp Jumlah_saldo keterangan	-	Data agen Data saldo
	8 DET	0 RET	2 FTR

Langkah selanjutnya adalah mendapatkan nilai UFP dari analisis DET, RET, dan FTR pada proses mengelola agen dan saldo yang nantinya hasilnya akan dikonversikan ke dalam perhitungan nilai KSLOC. Seperti dijelaskan pada tabel di bawah 5.15 ini.

Tabel 5. 15 Perhitungan UFP Perangkat Lunak DBA Ticketing pada proses mengelola laporan agen dan saldo

Fungsi	Jumlah	Perhitungan bobot kompleksitas			Nilai kompleksitas	Bobot Kompleksitas	UFP
		DET	RET	FTR			
ILF	2	10	2	-	Low	7	14
EIF	0	-	-	-	-	-	0
EI	5	9	-	2	Average	4	20
EQ	0	-	-	-	-	-	0
EO	2	7	-	2	Average	5	10
Total UFP							44

Selanjutnya Berikut ini adalah hasil analisis DET, RET, dan FTR untuk proses mengelola tiket yang peneliti dapatkan dari PDM. Seperti dijelaskan pada tabel di bawah 5.16 ini.

Tabel 5. 16 Analisis DET,RET, Dan FTR Perangkat Lunak DBA Ticketing proses mengelola tiket

Fungsi	DET	RET	FTR
ILF	Id_tiket Judul Jumlah_tiket Harga No_tiket Id_kategori Kategori keterangan	Data tiket Data kategori	-
	8 DET	2 RET	0 FTR
EIF	-	-	-
	0 DET	0 RET	0 FTR
EI	Judul Jumlah_tiket	-	Data tiket Data kategori

Fungsi	DET	RET	FTR
	No_tiket Harga Kategori keterangan Tombol Error message		
	8 DET	-	2 FTR
EQ	-	-	-
	DET	0 RET	0 FTR
EO	Judul Jumlah_tiket No_tiket Harga Kategori keterangan	-	Data tiket Data kategori
	6 DET	0 RET	2 FTR

Langkah selanjutnya adalah mendapatkan nilai UFP dari analisis DET, RET, dan FTR pada proses mengelola tiket yang nantinya hasilnya akan dikonversikan ke dalam perhitungan nilai KSLOC. Seperti dijelaskan pada tabel 5.17 di bawah ini.

Tabel 5. 17 Perhitungan UFP Perangkat Lunak *DBA Ticketing* pada proses mengelola tiket

Fungsi	Jumlah	Perhitungan bobot kompleksitas			Nilai kompleksitas	Bobot Kompleksitas	UFP
		DET	RET	FTR			
ILF	1	7	1	-	Low	7	7
EIF	0	-	-	-	-	-	0
EI	3	7	-	1	Low	3	9
EQ	0	-	-	-	-	-	0
EO	1	5	-	1	Low	4	4
Total UFP							20

Selanjutnya adalah menghitung total UFP yang ada pada sistem *DBA Ticketing* untuk dikonversi menjadi KSLOC. Seperti dijelaskan pada tabel 5.18 di bawah ini.

Tabel 5. 18 Proses Perhitungan KSLOC Perangkat Lunak *DBA Ticketing*

Proses	ILF	EIF	EI	EQ	EO	UFP	SLOC	KSLOC
Mengolah pembelian	2	-	6	-	2	48	1920	1.92
Laporan keuangan	2	-	6	-	2	48	1920	1.92
Mengelola agen dan saldo	2	-	5	-	2	44	1760	1.76
Mengelola tiket	1	-	3	-	1	20	800	0.8
Total	7		20		7	160	6430	6.43

Dari tabel 5.18 di atas kita mendapatkan hasil nilai UFP dari setiap proses dari DFD level 1 sistem informasi Pintu Air yang diakumulasikan sehingga didapatkan total UFP sebesar **160**, selanjutnya setiap UFP akan dikalikan dengan 40 yang merupakan nilai standar dari bahasa pemrograman HTML, dikarenakan bahasa pemrograman yang digunakan untuk membuat perangkat lunak sistem informasi Pintu Air. Sehingga nilai konversi dari sistem informasi pintu air sebesar

6430 SLOC, untuk mendapatkan nilai *size* dalam satuan KSLOC maka nilai SLOC dibagi dengan 1000. Hasil akhir nilai *size* dari sistem informasi Pintu Air adalah 6.43 yang nantinya akan dimasukkan ke dalam perhitungan *effort estimation*.

5.3.2 Analisis Perhitungan UFP Perangkat Lunak Sistem Informasi Pintu Air

Disini masing-masing function point pada setiap proses akan dinilai berdasarkan aturan DET, RET, dan FTR.

Berikut ini adalah hasil analisis DET, RET, dan FTR untuk proses mengelola pintu air yang peneliti dapatkan dari PDM. Seperti dijelaskan pada tabel 5.19 di bawah ini.

Tabel 5. 19 Analisis DET,RET, dan FTR perangkat lunak Sistem Informasi Pintu Air proses mengelola pintu air

Fungsi	DET	RET	FTR
ILF	Id_pintuair Nama_waduk Batas_atas Batas_bawah Id_Latitude Id_Longitude Alamat_waduk tanggal Id_indikator keterangan Indikator	Data pintu air Data indikator	-
	11 DET	2 RET	0 FTR
EIF	-	-	-
	0 DET	0 RET	0 FTR
EI	Nama_waduk Batas_atas Batas_bawah Tanggal Alamat_waduk Tombol Error message	-	Data pintu air Data indikator



Fungsi	DET	RET	FTR
	7 DET	-	2 FTR
EQ	Search by	-	Data pintu air Data indikator
	1 DET	0 RET	2 FTR
EO	Nama_waduk Batas_atas Batas_bawah Latitude Longitude Alamat_waduk tanggal Indikator keterangan	-	Data pintu air Data indikator
	9 DET	0 RET	2 FTR

Langkah selanjutnya adalah mendapatkan nilai UFP dari analisis DET, RET, dan FTR pada proses mengelola pintu air yang nantinya hasilnya akan dikonversikan ke dalam perhitungan nilai KSLOC. Seperti dijelaskan pada tabel 5.20 di bawah ini.

Tabel 5. 20 Perhitungan UFP perangkat lunak Sistem Informasi Pintu Air pada proses mengelola pintu air

Fungsi	Jumlah	Perhitungan bobot kompleksitas			Nilai kompleksitas	Bobot Kompleksitas	UFP
		DET	RET	FTR			
ILF	2	11	2	-	Low	7	14
EIF	0	-	-	-	-	-	0
EI	5	7	-	2	Average	4	20
EQ	1	1	-	2	Low	3	3
EO	2	9	-	2	Average	5	10
Total UFP							47

Berikut ini adalah hasil analisis function point untuk proses mengelola mengelola laporan keuangan yang peneliti dapatkan dari PDM. Seperti dijelaskan pada tabel 5.21 di bawah ini.

Tabel 5. 21 Analisis DET,RET, dan FTR perangkat lunak Sistem Informasi Pintu Air proses mengelola laporan keuangan

Fungsi	DET	RET	FTR
ILF	Id_laporan Judul_laporan Isi_laporan Id_jenis_laporan Jenis_laporan Keterangan_laporan Tanggal	Data laporan keuangan	-
	7 DET	1 RET	0 FTR
EIF	-	-	-
	0 DET	0 RET	0 FTR
EI	Judul_laporan Isi_laporan Jenis_laporan Keterangan_laporan Tanggal Tombol Error message	-	Data laporan keuangan
	7 DET	-	1 FTR
EQ	-	-	-
	DET	0 RET	0 FTR
EO	Judul_laporan Isi_laporan Jenis_laporan Keterangan_laporan Tanggal	-	Data laporan keuangan
	5 DET	0 RET	1 FTR

Langkah selanjutnya adalah mendapatkan nilai UFP dari analisis DET, RET, dan FTR pada proses mengelola laporan keuangan yang nantinya hasilnya akan dikonversikan ke dalam perhitungan nilai KSLOC. Seperti dijelaskan pada tabel 5.22 di bawah ini.

Tabel 5. 22 Perhitungan UFP perangkat lunak Sistem Informasi Pintu Air pada proses mengelola laporan keuangan

Fungsi	Jumlah	Perhitungan bobot kompleksitas			Nilai kompleksitas	Bobot Kompleksitas	UFP
		DET	RET	FTR			
ILF	1	7	1	-	Low	7	7
EIF	0	-	-	-	-	-	0
EI	3	7	-	1	Low	3	9
EQ	0	-	-	-	-	-	0
EO	1	5	-	1	Low	4	4
Total UFP							20

Berikut ini adalah hasil analisis function point untuk proses mengelola profil yang peneliti dapatkan dari PDM. Seperti dijelaskan pada tabel 5.23 di bawah ini.

Tabel 5. 23 Analisis DET,RET, dan FTR perangkat lunak Sistem Informasi Pintu Air proses mengelola profil

Fungsi	DET	RET	FTR
ILF	Id_profil Judul_latar_belakang deskripsi Judul_visi_misi Deskripsi Judul_struktur Deskripsi Judul_wilayah Deskripsi Judul_contact Deskripsi	Data profil	-



Fungsi	DET	RET	FTR
	11 DET	1 RET	0 FTR
EIF	-	-	-
	0 DET	0 RET	0 FTR
EI	Judul_latar_belakang deskripsi Judul_visi_misi Deskripsi Judul_struktur Deskripsi Judul_wilayah Deskripsi Judul_contact Deskripsi	-	Data profil
	10 DET	-	1 FTR
EQ	-	-	-
	DET	0 RET	0 FTR
EO	Judul_latar_belakang deskripsi Judul_visi_misi Deskripsi Judul_struktur Deskripsi Judul_wilayah Deskripsi Judul_contact Deskripsi	-	Data profil
	10 DET	0 RET	1 FTR

Langkah selanjutnya adalah mendapatkan nilai UFP dari analisis DET, RET, dan FTR pada proses mengelola profil yang nantinya hasilnya akan

dikonversikan ke dalam perhitungan nilai KSLOC. Seperti dijelaskan pada tabel 5.24 di bawah ini.

Tabel 5. 24 Perhitungan UFP perangkat lunak Sistem Informasi Pintu Air pada proses mengelola profil

Fungsi	Jumlah	Perhitungan bobot kompleksitas			Nilai kompleksitas	Bobot Kompleksitas	UFP
		DET	RET	FTR			
ILF	1	11	1	-	Low	7	7
EIF	0	-	-	-	-	-	0
EI	3	10	-	1	Low	3	9
EQ	0	-	-	-	-	-	0
EO	1	10	-	1	Low	4	4
Total UFP							20

Berikut ini adalah hasil analisis function point untuk proses mengelola publikasi yang peneliti dapatkan dari PDM. Seperti dijelaskan pada tabel 5.25 di bawah ini.

Tabel 5. 25 Analisis DET,RET, dan FTR perangkat lunak Sistem Informasi Pintu Air proses mengelola publikasi

Fungsi	DET	RET	FTR
ILF	Id_berita Judul_berita Deskripsi Id_artikel Judul_artikel deskripsi tanggal	Data berita Data artikel	-
	7 DET	1 RET	0 FTR
EIF	-	-	-
	0 DET	0 RET	0 FTR
EI	Judul_berita Deskripsi	-	Data berita Data artikel

Fungsi	DET	RET	FTR
	Judul_artikel deskripsi tanggal		
	5 DET	-	1 FTR
EQ	-	-	-
	DET	0 RET	0 FTR
EO	Judul_berita Deskripsi Judul_artikel deskripsi tanggal	-	Data berita Data artikel
	5 DET	0 RET	1 FTR

Langkah selanjutnya adalah mendapatkan nilai UFP dari analisis DET, RET, dan FTR pada proses mengelola publikasi yang nantinya hasilnya akan dikonversikan ke dalam perhitungan nilai KSLOC. Seperti dijelaskan pada tabel 5.26 di bawah ini.

Tabel 5. 26 Perhitungan UFP perangkat lunak Sistem Informasi Pintu Air pada proses mengelola publikasi

Fungsi	Jumlah	Perhitungan bobot kompleksitas			Nilai kompleksitas	Bobot Kompleksitas	UFP
		DET	RET	FTR			
ILF	2	7	2	-	Low	7	14
EIF	0	-	-	-	-	-	0
EI	6	5	-	2	Average	4	24
EQ	0	-	-	-	-	-	0
EO	2	5	-	2	Low	4	8
Total UFP							46

Berikut ini adalah hasil analisis function point untuk proses mengelola pengaduan yang peneliti dapatkan dari PDM. Seperti dijelaskan pada tabel 5.27 di bawah ini.

Tabel 5. 27 Analisis DET,RET, Dan FTR Perangkat Lunak Sistem Informasi Pintu Air proses mengelola pengaduan

Fungsi	DET	RET	FTR
ILF	Id_pengaduan Nama Email Profesi Unit_kerja Alamat Deskripsi_pengaduan	Data pengaduan	-
	7 DET	1 RET	0 FTR
EIF	-	-	-
	0 DET	0 RET	0 FTR
EI	Nama Email Profesi Unit_kerja Alamat Deskripsi_pengaduan Tombol	-	Data pengaduan
	7 DET	-	1 FTR
EQ	-	-	-
	DET	0 RET	0 FTR
EO	Nama Email Profesi Unit_kerja Alamat Deskripsi_pengaduan	-	Data pengaduan
	6 DET	0 RET	1 FTR



Langkah selanjutnya adalah mendapatkan nilai UFP dari analisis DET, RET, dan FTR pada proses mengelola pengaduan yang nantinya hasilnya akan dikonversikan ke dalam perhitungan nilai KSLOC. Seperti dijelaskan pada tabel 5.28 di bawah ini.

Tabel 5. 28 Perhitungan UFP Perangkat Lunak Sistem Informasi Pintu Air Pada proses mengelola pengaduan

Fungsi	Jumlah	Perhitungan bobot kompleksitas			Nilai kompleksitas	Bobot Kompleksitas	UFP
		DET	RET	FTR			
ILF	1	7	1	-	low	7	7
EIF	0	-	-	-	-	-	0
EI	2	7	-	1	Low	3	6
EQ	0	-	-	-	-	-	0
EO	1	6	-	1	Low	4	4
Total UFP							17

Selanjutnya adalah menghitung total UFP pada Sistem Informasi Pintu Air untuk dikonversi menjadi KSLOC. Seperti dijelaskan pada tabel 5.29 di bawah ini.

Tabel 5. 29 Proses Perhitungan KSLOC Perangkat Lunak Sistem Informasi Pintu Air

Proses	ILF	EIF	EI	EQ	EO	UFP	SLOC	KSLOC
Pengelolaan air	2	-	5	1	2	47	1880	1.88
Mengelola Laporan keuangan	1	-	3	-	1	20	800	0.8
Mengelola profil	1	-	3	-	1	20	800	0.8
Mengelola publikasi	2	-	6	-	2	46	1840	1.84
Mengelola Pengaduan	1	-	2	-	1	17	680	0.68



Proses	ILF	EIF	EI	EQ	EO	UFP	SLOC	KSLOC
Total	7	-	19	1	7	150	6000	6.00

Dari tabel 5.29 diatas kita mendapatkan hasil nilai UFP dari setiap proses dari DFD level 1 sistem informasi Pintu Air yang diakumulasikan sehingga didapatkan total UFP sebesar **150**, selanjutnya setiap UFP akan dikalikan dengan 40 yang merupakan nilai standar dari bahasa pemrograman HTML, dikarenakan bahasa pemrograman yang digunakan untuk membuat perangkat lunak sistem informasi Pintu Air. Sehingga nilai konversi dari sistem informasi pintu air sebesar **6000 SLOC**, untuk mendapatkan nilai *size* dalam satuan KSLOC maka nilai SLOC dibagi dengan 1000. Hasil akhir nilai *size* dari sistem informasi Pintu Air adalah **6.00** yang nantinya akan dimasukkan ke dalam perhitungan *effort estimation*.

5.4 Analisis perhitungan Faktor Skala

Pada tahap mendapatkan nilai faktor skala, penelitian ini menggunakan penilaian yang berisi faktor-faktor yang diperlukan untuk pengembangan sebuah proyek perangkat lunak. nilai faktor skala yang dihitung guna mendapatkan nilai faktor eksponen yang selanjutnya digunakan untuk menghitung nilai effort estimation.

5.4.1 Perhitungan Faktor Skala Perangkat Lunak *DBA Ticketing*

Metode yang peneliti gunakan dalam mendapatkan data nilai faktor skala menggunakan lembar kuesioner penilaian faktor skala yang telah berisi parameter dan skala yang diberikan pada anggota tim proyek yang terlibat dalam pengembangan perangkat lunak *DBA Ticketing*. Berikut ini nilai dari faktor skala perangkat lunak *DBA Ticketing* yang dijelaskan pada tabel 5.30 di bawah ini.

Tabel 5. 30 Perhitungan Nilai Faktor Skala Perangkat Lunak *DBA Ticketing*

No.	Scale factor	Responden 1 (Fathin)	Hasil Penilaian
1.	PREC	<i>Very High</i>	1.24
2.	FLEX	<i>Very High</i>	1.01
3.	RESL	<i>High</i>	2.83
4.	TEAM	<i>Very high</i>	1.10
5.	PMAT	<i>High</i>	3.12
Total penilaian factor scale		=	9.3

Nilai dari tabel 5.30 di atas dimasukkan dalam persamaan yang digunakan untuk menghitung nilai *effort estimation*.

$$E = 0.91 + 0.01 * 9.3 = 1.01$$

Dari persamaan diatas, nilai 0.91 dan 0.01 didapatkan dari persamaan factor eksponen metode COCOMO II, nilai 9.3 adalah nilai total faktor skala. Sehingga nilai faktor eksponen yang didapatkan untuk perangkat lunak *DBA Ticketing* adalah **1.01** yang nantinya akan dimasukkan dalam persamaan *effort estimation*.

5.4.2 Perhitungan Faktor Skala Perangkat Lunak Sistem Pintu Air

Metode yang peneliti gunakan dalam mendapatkan data nilai faktor skala menggunakan lembar kuesioner penilaian faktor skala yang telah berisi parameter dan skala yang diberikan pada anggota tim proyek yang terlibat dalam pengembangan perangkat lunak Sistem Pintu Air. Berikut ini nilai dari faktor skala perangkat lunak Sistem Pintu Air yang dijelaskan pada tabel 5.31 berikut.

Tabel 5. 31 Perhitungan Nilai Faktor Skala Perangkat Lunak Sistem Pintu Air

<i>Scale factor</i>	Responden 1 (Yusak)	Responden 2 (Reno)	Hasil Penilaian
PREC	<i>Very High</i>	<i>Very high</i>	1.24
FLEX	<i>Nominal</i>	<i>Very high</i>	$(3.04 \times 0.5) + (1.01 \times 0.5) = 2.02$
RESL	<i>Very High</i>	<i>Very high</i>	1.41
TEAM	<i>Very High</i>	<i>Very high</i>	1.10
PMAT	<i>Very High</i>	<i>Nominal</i>	$(1.56 \times 0.5) + (3.12 \times 0.5) = 2.34$
Total penilaian factor scale	=		8.62

Nilai dari tabel 5.31 diatas dimasukkan dalam persamaan yang digunakan untuk menghitung nilai *effort estimation*.

$$E = 0.91 + 0.01 * 8.62 = 0.99$$

Dari persamaan diatas, nilai 0.91 dan 0.01 didapatkan dari persamaan faktor eksponen metode COCOMO II, nilai 8.62 adalah nilai total faktor skala. Sehingga nilai faktor eksponen yang didapatkan untuk perangkat lunak Sistem Pintu Air adalah **0.99** yang nantinya akan dimasukkan dalam persamaan *effort estimation*.

5.5 Analisis Perhitungan *Effort estimation*

Pada tahap mendapatkan nilai *effort multiplier*, penelitian ini menggunakan penilaian yang berisi faktor-faktor yang diperlukan untuk pengembangan sebuah proyek perangkat lunak. nilai *Effort multiplier* ini selanjutnya digunakan untuk menghitung nilai *effort estimation*.

5.5.1 Analisis Perhitungan *Effort Estimation* Perangkat Lunak *DBA Ticketing*

Metode yang peneliti gunakan dalam mendapatkan data nilai *Effort multiplier* menggunakan lembar kuesioner penilaian *Effort multiplier* yang telah berisi parameter dan skala yang diberikan pada anggota tim proyek yang terlibat dalam pengembangan perangkat lunak *DBA Ticketing*. Berikut ini nilai dari *Effort multiplier* perangkat lunak *DBA Ticketing* yang dijelaskan pada tabel 5.32 di bawah.

Tabel 5. 32 Perhitungan Nilai Faktor Skala Perangkat Lunak *DBA Ticketing*

<i>Effort multipliers</i>	Responden 1 (Fathin)	Hasil penilaian
RELY	High	1.10
DATA	Very High	1.28
CPLX	Low	0.87
DOCU	Very low	0.91
RUSE	Nominal	1.00
ACAP	Very low	1.19
PCON	Very Low	1.29
TIME	Nominal	1.00
STOR	Nominal	1.00
PVOL	Very high	1.30
PCAP	Very low	1.34
APEX/AEXP	Nominal	1.00
PLEX/PEXP	Very Low	1.19
LTEX	Very Low	1.20
TOOL	High	0.90
SITE	High	0.93
SCED	Very High	1.00
Total penilaian <i>effort multipliers</i>	=	18.5/17=1.08

Rerata nilai effort multiplier dari tabel 5.32 diatas dimasukkan dalam persamaan yang digunakan untuk menghitung nilai *effort estimation*.

$$PM = 2.94 \times (6.43)^{1.01} \times 1.08 = \mathbf{20.80}$$

Dari persamaan diatas, nilai 2.94 didapatkan nilai metode COCOMO II, nilai **6.43** adalah nilai total UFP perangkat lunak *DBA Ticketing*, nilai **1.01** merupakan nilai faktor eksponen yang didapatkan untuk perangkat lunak *DBA Ticketing*, dan nilai **1.08** merupakan nilai rerata *effort multipliers* perangkat lunak *DBA Ticketing*. Sehingga person month perangkat lunak *DBA Ticketing* adalah **20.80** yang nantinya dimasukkan ke dalam TDEV.

5.5.2 Analisis perhitungan Effort Estimation Perangkat lunak Sistem Pintu Air

Metode yang peneliti gunakan dalam mendapatkan data nilai *Effort multiplier* menggunakan lembar kuesioner penilaian *Effort multiplier* yang telah berisi parameter dan skala yang diberikan pada anggota tim proyek yang terlibat dalam pengembangan perangkat lunak Sistem Pintu Air. Berikut ini nilai dari *Effort multiplier* perangkat lunak Sistem Pintu Air yang dijelaskan pada tabel 5.33 berikut.

Tabel 5. 33 perhitungan nilai faktor skala perangkat lunak Sistem Pintu Air

<i>Effort multipliers</i>	Responden 1 (Yusak)	Responden 2 (Reno)	Hasil penilaian
RELY	<i>Very Low</i>	<i>Very Low</i>	0.82
DATA	<i>High</i>	<i>Nominal</i>	$(1,14 \times 0,5) + (1,00 \times 0,5) = 1,07$
CPLX	<i>Nominal</i>	<i>Very high</i>	$(1,00 \times 0,5) + (1,34 \times 0,5) = 1,17$
DOCU	<i>Nominal</i>	<i>Nominal</i>	1,00
RUSE	<i>Low</i>	<i>High</i>	$(0,95 \times 0,5) + (1,15 \times 0,5) = 1,05$ high
ACAP	<i>Nominal</i>	<i>Very High</i>	$(1,00 \times 0,5) + (0,71 \times 0,5) = 0,85$
PCON	<i>Very low</i>	<i>Nominal</i>	$(1,29 \times 0,5) + (1,00 \times 0,5) = 1,14$
TIME	<i>Nominal</i>	<i>Nominal</i>	1,00
STOR	<i>Nominal</i>	<i>Nominal</i>	1,00
PVOL	<i>Low</i>	<i>Low</i>	0,87

<i>Effort multipliers</i>	Responden 1 (Yusak)	Responden 2 (Reno)	Hasil penilaian
PCAP	<i>Nominal</i>	<i>Nominal</i>	1,00
APEX/AEXP	<i>High</i>	<i>High</i>	0,88
PLEX/PEXP	<i>High</i>	<i>High</i>	0,91
LTEX	<i>High</i>	<i>Nominal</i>	(0,91x0 ,5) +(1,00 x 0,5)= 0,95
TOOL	<i>High</i>	<i>High</i>	0,90
SITE	<i>High</i>	<i>Extra High</i>	(0,93x0 ,5) +(0,86 x 0,5)=0,89
SCED	<i>Nominal</i>	<i>Nominal</i>	1,00
Total penilaian effort multipliers	=		16.5/17=0.97

Rerata nilai effort multiplier dari tabel 5.33 diatas dimasukkan dalam persamaan yang digunakan untuk menghitung nilai *effort estimation*.

$$PM = 2.94 \times (6.00)^{0.99} \times 0.97 = \mathbf{16.08}$$

Dari persamaan diatas, nilai 2.94 didapatkan nilai metode COCOMO II, nilai **6.00** adalah nilai total UFP perangkat lunak Sistem Pintu Air, nilai **0.99** merupakan nilai faktor eksponen yang didapatkan untuk perangkat lunak Sistem Pintu Air, dan nilai **0.97** merupakan nilai rerata *effort multipliers* perangkat lunak Sistem Pintu Air. Sehingga person month perangkat lunak Sistem Pintu Air adalah **16.08** yang nantinya dimasukkan ke dalam TDEV.

5.6 Menghitung Estimasi Biaya Proyek

Selanjutnya setelah mendapatkan nilai person-month (PM) maka dilanjutkan dengan menghitung persamaan TDEV untuk mendapatkan perkiraan waktu pengerjaan proyek dalam satuan bulan. Setelahnya mendapat estimasi waktu pengerjaan proyek maka dilanjutkan dengan menghitung jumlah pegawai yang dibutuhkan menggunakan persamaan *Average Staffing*. Setelah mendapatkan nilai dari estimasi waktu dan SDM maka selanjutnya adalah menghitung estimasi biaya proyek dengan cara mengalikan jumlah pegawai yang dibutuhkan dengan gaji UMR kota malang kemudian hasilnya nanti dikalikan lagi dengan estimasi waktu proyek. Berikut ini adalah persamaan 3 untuk menghitung estimasi waktu. Kemudian estimasi jumlah SDM yang dibutuhkan dapat dihitung menggunakan persamaan 4.

5.6.1 Analisis perhitungan Estimasi Biaya Perangkat lunak DBA Ticketing

Setelah menganalisis *effort estimation* perangkat lunak DBA Ticketing. Selanjutnya kita menghitung estimasi waktu dan juga biaya yang dikeluarkan untuk mengerjakan proyek perangkat lunak DBA Ticketing. Untuk menghitung estimasi waktu maka digunakan persamaan 3 dengan nilai PM sebesar 20,80 dan E = 1,01 lalu dihitung seperti persamaan di bawah ini.

$$TDEV = 3.67 \times (20.80)^{(0.28+0,2 \times (1.01-0.91))} = 9.12 \text{ atau } 9 \text{ bulan}$$

Setelah mendapat nilai estimasi waktu/TDEV, Kemudian langkah selanjutnya adalah mendapatkan nilai estimasi staff, untuk menghitung estimasi pegawai maka digunakan persamaan 4 dengan nilai PM sebesar 20,80 dan TDEV = 9 lalu dihitung seperti persamaan di bawah ini.

$$\text{Average staff} = \frac{20.80}{9} = 2,3 \text{ atau } 3 \text{ orang pegawai.}$$

Langkah selanjutnya adalah mendapatkan nilai estimasi biaya perbulan, untuk menghitung estimasi biaya perbulan maka digunakan persamaan 5 dengan nilai *Average staff* = 3 dan UMR = Rp.2.272.000 lalu dihitung seperti persamaan di bawah ini.

$$\text{Biaya per bulan} = \text{Average staff} \times \text{UMR} = 3 \times \text{Rp.2.272.000} = \text{Rp.6.816.000}$$

Langkah selanjutnya adalah mendapatkan nilai estimasi biaya total, untuk menghitung estimasi biaya total maka digunakan persamaan 6 dengan nilai biaya perbulan = Rp.6.816.000 dan TDEV = 9 lalu dihitung seperti persamaan di bawah ini.

$$\begin{aligned} \text{Biaya Total yang didapatkan} &= \text{Biaya per bulan} \times \text{TDEV} \\ &= 9 \times \text{Rp.6.816.000} = \text{Rp.61.344.000,-} \end{aligned}$$

5.6.2 Analisis Perhitungan Estimasi Biaya Perangkat lunak Sistem Pintu Air

Setelah menganalisis *effort estimation* perangkat lunak Sistem Informasi Pintu Air. Selanjutnya kita menghitung estimasi waktu dan juga biaya yang dikeluarkan untuk mengerjakan proyek perangkat lunak Sistem Informasi Pintu Air. Untuk menghitung estimasi waktu maka digunakan persamaan 3 dengan nilai PM = 16,08 dan E = 0,99 lalu dihitung seperti persamaan di bawah ini.

$$TDEV = 3.67 \times (16.08)^{(0.28+0,2 \times (0.99-0.91))} = 8.35 \text{ atau } 8 \text{ bulan}$$

Setelah mendapat nilai estimasi waktu/TDEV, Kemudian langkah selanjutnya adalah mendapatkan nilai estimasi staff, untuk menghitung estimasi pegawai maka digunakan persamaan 4 dengan nilai PM = 16,08 dan TDEV = 8 lalu dihitung seperti persamaan di bawah ini.

$$\text{Average staff} = \frac{16.08}{8} = 2.01 \text{ atau } 2 \text{ orang pegawai.}$$

Langkah selanjutnya adalah mendapatkan nilai estimasi biaya perbulan, untuk menghitung estimasi biaya perbulan maka digunakan persamaan 5 dengan nilai *Average staff* = 2 dan UMR = Rp.2.272.000 lalu dihitung seperti persamaan di bawah ini.

$$\text{Biaya per bulan} = \text{Average staff} \times \text{UMR} = 2 \times 2.272.000 = \text{Rp.4.544.000,-}$$

Langkah selanjutnya adalah mendapatkan nilai estimasi biaya total, untuk menghitung estimasi biaya total maka digunakan persamaan 6 dengan nilai biaya perbulan = Rp.4.544.000,- dan TDEV = 8 lalu dihitung seperti persamaan di bawah ini.

$$\begin{aligned} \text{Biaya Total yang didapatkan} &= \text{Biaya per bulan} \times \text{TDEV} \\ &= 8 \times \text{Rp.4.544.000} = \text{Rp.36.352.000,-} \end{aligned}$$

5.7 Pembahasan Estimasi waktu, SDM, dan Biaya

Selanjutnya adalah pembahasan hasil estimasi waktu, SDM, dan Biaya antara metode yang digunakan oleh perusahaan dengan metode COCOMO II.

5.7.1 Estimasi Sumberdaya Manusia (SDM)

Dalam estimasi sumberdaya manusia merupakan hasil perkiraan jumlah sumberdaya manusia yang dibutuhkan untuk mengerjakan suatu proyek agar nantinya proyek yang dikerjakan berhasil seperti yang diharapkan. Dengan mengetahui skala proyek maka dapat diperkirakan jumlah sumberdaya yang dibutuhkan sehingga nantinya dapat menurunkan biaya proyek agar tidak terlalu banyak dalam membiayai sumberdaya manusia, juga nantinya dapat membuat proyek selesai sesuai waktu yang ditentukan dan tidak melebihi batas dikarenakan SDM yang kurang.

Jumlah masing-masing SDM yang diberikan oleh CV. Profile Image Studio untuk perangkat lunak *DBA Ticketing* adalah 1 orang dan untuk perangkat lunak Sistem Informasi Pintu Air adalah sebanyak 2 orang. Sedangkan dari hasil analisis COCOMO II didapatkan untuk perangkat lunak *DBA Ticketing* adalah 3 orang, dan perangkat lunak Y adalah 2 orang. Selisih yang didapatkan antara CV. Profile Image Studio dengan COCOMO II adalah sebesar 2 orang.

Dalam menentukan jumlah SDM pihak perusahaan memang lebih sedikit dalam menggunakan SDM dibandingkan dengan metode COCOMO II akan tetapi dalam pengerjaan proyek perangkat lunak akan terjadi SDM banyak melakukan pekerjaan ganda sedangkan dalam metode COCOMO II lebih mementingkan jumlah pembagian SDM berdasarkan besarnya nilai estimasi usaha yang ada pada setiap proyek. Karena pada dasarnya semakin cepat pengerjaan proyek maka semakin banyak membutuhkan SDM begitu juga sebaliknya.

5.7.2 Estimasi Waktu

Estimasi waktu adalah perkiraan pembuatan jadwal yang digunakan untuk menyelesaikan suatu proyek. Penentuan waktu proyek harusnya dilakukan dengan cara analisis yang tepat sehingga nantinya dapat selesai sesuai waktu yang telah dijadwalkan. Hal ini dibuat karena dapat menekan biaya dan juga waktu dalam mengerjakan proyek agar tidak bertambah lama.

Waktu yang diberikan oleh CV. Profile Image Studio untuk perangkat lunak *DBA Ticketing* dan Sistem Informasi Pintu Air diberi waktu yang sama yaitu 4 bulan. Sedangkan hasil dari analisis menggunakan metode *COCOMO II*, waktu yang dibutuhkan dalam mengerjakan proyek perangkat lunak *DBA Ticketing* adalah 9 bulan sedangkan perangkat lunak Y adalah 8 bulan. Jadi selisih yang diberikan oleh CV. Profile Image Studio dengan metode *COCOMO II* adalah 9 bulan.

Meskipun perkiraan dari metode *COCOMO II* lebih lama tetapi hal ini dapat meminimalisir risiko terjadinya *overtime* pengerjaan pada proyek yang dikerjakan sehingga tidak mengurangi kepercayaan klien terhadap perusahaan.

5.7.3 Estimasi Biaya

Biaya adalah semua pengorbanan yang perlu dikeluarkan dalam suatu proses produksi baik barang maupun jasa yang biasanya dinyatakan dalam satuan nominal uang menurut harga pasar yang berlaku baik yang sudah maupun sedang terjadi. Estimasi biaya perangkat lunak adalah suatu proses dalam memperkirakan biaya yang akan muncul dalam membangun atau memelihara suatu perangkat lunak.

Estimasi biaya proyek yang diberikan oleh CV. Profile Image Studio untuk perangkat lunak *DBA Ticketing* adalah Rp.25.000.000, sedangkan menurut menurut hasil analisis menggunakan metode *COCOMO II* adalah sebesar Rp.61.344.000,-. Untuk perangkat lunak Sistem Informasi Pintu Air CV. Profile Image Studio memberikan harga sebesar Rp.35.000.000, sedangkan menurut hasil analisis metode *COCOMO II* adalah sebesar Rp. Rp.36.352.000,-.

Selisih harga untuk perangkat lunak *DBA Ticketing* dan Sistem Informasi Pintu Air adalah Rp.37.686.000, selisih tersebut dikarenakan metode *COCOMO II* menggunakan setiap fungsi dari DFD untuk menilai kompleksitas suatu sistem yang diukur berdasarkan besarnya jumlah nilai UFP pada sistem. Karena semakin besar nilai UFP sistem maka semakin besar juga nilai estimasi usaha dari proyek tersebut sehingga nantinya akan mempengaruhi besarnya estimasi biaya yang akan dikeluarkan begitu juga sebaliknya dan juga metode *COCOMO II* menggunakan lembar penilaian *scale factor* dan *effort multipliers* dalam menentukan jenis karakteristik dari proyek yang dikerjakan. Sehingga metode *COCOMO II* akan lebih tepat dalam memperkirakan estimasi biaya pada suatu proyek yang pada akhirnya dapat meminimalisir kerugian yang diterima oleh perusahaan.

BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dalam melakukan penghitungan estimasi biaya pengembangan perangkat lunak *DBA Ticketing* dan Sistem Informasi Pintu Air pada CV. Profile Image Studio maka peneliti dapat menyimpulkan sebagai berikut:

1. Dalam menentukan Skala proyek, CV. Profile Image Studio menggunakan pedoman Agile development untuk mengukur skala proyek. Masing-masing proyek perangkat lunak yang ada pada penelitian ini memiliki 1-2 orang pada 1 tim maka dapat dikategorikan sebagai proyek skala kecil.
2. Dari waktu asli yang diberikan oleh CV. Profile Image Studio untuk perangkat lunak *DBA Ticketing* dan Sistem Informasi Pintu Air diberi waktu yang sama yaitu 3 bulan. Sedangkan hasil dari analisis menggunakan metode *COCOMO II*, waktu yang dibutuhkan dalam mengerjakan proyek perangkat lunak *DBA Ticketing* adalah 9 bulan sedangkan perangkat lunak Sistem Informasi Pintu Air adalah 8 bulan.
3. Dari jumlah SDM yang diberikan oleh CV. Profile Image Studio untuk perangkat lunak *DBA Ticketing* adalah 1 orang dan Sistem Informasi Pintu Air diberi 2 orang. Sedangkan dari hasil analisis *COCOMO II* didapatkan untuk perangkat lunak *DBA Ticketing* adalah 3 orang, dan perangkat lunak Sistem Informasi Pintu Air adalah 2 orang.
4. Estimasi biaya proyek yang diberikan oleh CV. Profile Image Studio untuk perangkat lunak *DBA Ticketing* adalah Rp.25.000.000,-. Untuk perangkat lunak Sistem Informasi Pintu Air sebesar Rp.35.000.000, sedangkan menurut menurut hasil analisis menggunakan metode *COCOMO II* untuk perangkat lunak *DBA Ticketing* adalah sebesar Rp.61.344.000,- dan perangkat lunak Sistem Informasi Pintu Air adalah sebesar Rp.36.352.000,-.
5. Perbandingan estimasi biaya, waktu, dan sumberdaya manusia seluruh perangkat lunak dengan skala proyek yang sama antara metode *COCOMO II* dengan actual cost memiliki selisih sebesar Rp.37.686.000, dengan selisih waktu yang dimiliki sebesar 9 bulan dan selisih sumberdaya manusia yang digunakan sebesar 2 orang. Selisih ini dikarenakan metode *COCOMO II* menggunakan setiap fungsi dari DFD untuk menilai kompleksitas suatu sistem yang diukur berdasarkan besarnya jumlah nilai UFP pada sistem. Karena semakin besar nilai UFP sistem maka semakin besar juga nilai estimasi usaha dari proyek tersebut begitu juga sebaliknya dan juga metode *COCOMO II* menggunakan lembar penilaian *scale factor* dan *effort multipliers* dalam menentukan jenis karakteristik dari proyek yang dikerjakan. Sehingga metode *COCOMO II* akan lebih tepat dalam memperkirakan estimasi biaya pada suatu proyek yang pada akhirnya dapat meminimalisir kerugian yang diterima oleh perusahaan.

6.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan dalam melakukan penghitungan estimasi biaya pengembangan perangkat lunak *DBA Ticketing* dan Sistem Informasi Pintu Air pada CV. Profile Image Studio maka peneliti dapat memberikan saran sebagai berikut:

1. Penelitian selanjutnya dapat mengembangkan penelitian ini dengan cara menambah kan metode estimasi lainnya untuk sebagai pembanding dengan penelitian ini.
2. Penelitian selanjutnya diharapkan dapat menggunakan pendekatan metode sub-model COCOMO II yang lain untuk dapat lebih memahami metode COCOMO II.
3. Penelitian selanjutnya dapat mengembangkan penelitian ini dengan menambahkan jenis proyek perangkat lunak yang berbeda.



DAFTAR REFERENSI

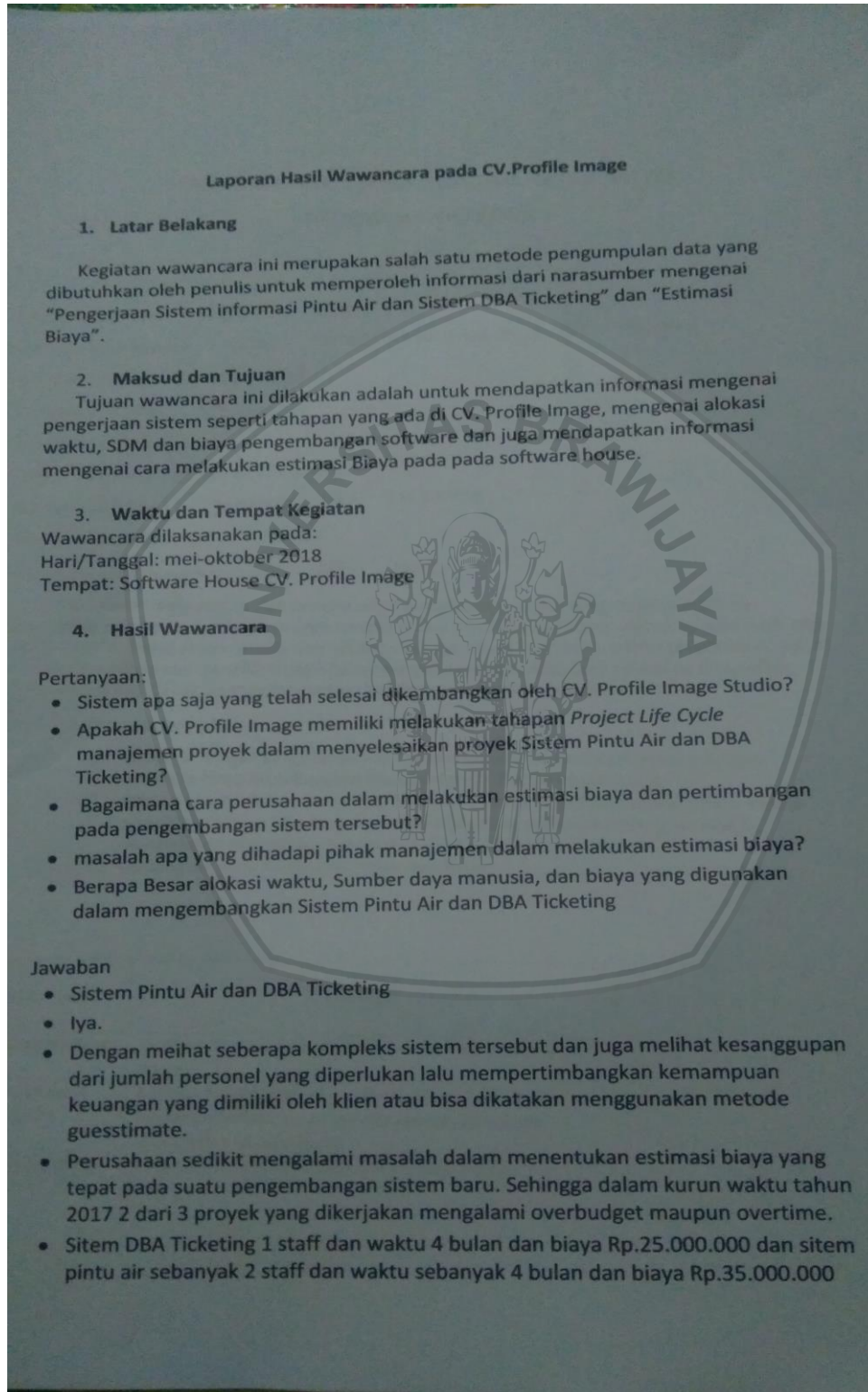
- A.S. Rosa, M. s., 2014. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- B. Prasetyo, S., 2006. Penggunaan Model Function Point Dalam Estimasi Usaha Proyek Pengembangan Sistem Informasi Bisnis. *Risalah Lokakarya Komputasi dalam Sains dan Teknologi Nuklir XVII*, p. 338.
- Barry Boehm, C. A. A. W. B. B. C. D.-C., 2000. *COCOMO II Model Definition Manual Version 2.1*. s.l.:University of Southern California.
- Cotterel, B. H. d. M., 2006. *Software Project management*. 4th Edition ed. s.l.:s.n.
- Dingsøyr T., F. T. , I. J., 2014. *What Is Large in Large-Scale? A Taxonomy of Scale for Agile Software Development*. 15th International Conference, PROFES 2014.
- Longstreet, D., 2004. *Fundamentals of Function Point Analysis*, s.l.: Software Development Magazine.
- M.G. Bintiri, A. S. R. D., 2012. *Perbandingan model algoritmik dan Non Algoritmik Untuk Estimasi Biaya Perangkat Lunak*. Yogyakarta, Seminar Nasional Aplikasi teknologi Informasi 2012.
- Marchewka, J., 2003. In: *Information Technology Project Management*. Hoboken: NJ Wiley.
- Maswinandar, 2016. *Perhitungan Biaya Proyek Sistem informasi Rekam Medis dengan Menggunakan Metode COCOMO II (Studi Kasus: 7Treesdigital)*. - ed. Malang: Universitas Brawijaya.
- Merlo, N., 2002. COCOMO (Constructive Cost Model). In: *Seminar on Cost Estimation WS 02/03*. Switzerland: Requirement Engineering Research Group Department of Computer Science University of Zurich, p. 8.
- Neelam Bawane nee' Singhal, C. V. S., 2008. A Case Study to Assess the Validity of Function. *World Academy of Science, Engineering and Technology*, 2(6), pp. 1860-1862.
- Quantitative software management, -. *function point languages table*. [Online] Available at: <http://www.qsm.com/resources/function-point-languages-table> [Accessed 15 08 2018].
- Schwalbe, K., 2014. *Information technology project management*. Seventh Edition ed. Boston: Cengage Learning.
- Sommerville, I., 2010. *Software Engineering*. 9th ed. Boston: Pearson Education inc..
- T.N.Sharma, 2011. Analysis of Software Cost Estimation using COCOMO II. *International Journal of Scientific & Engineering Research*, 2(6).

T.N.Sharma, A. B. A. S., 2011. A Comparative study of COCOMO II and Putnam models of Software Cost Estimation. *International Journal of Scientific & Engineering Research*, 2(11).

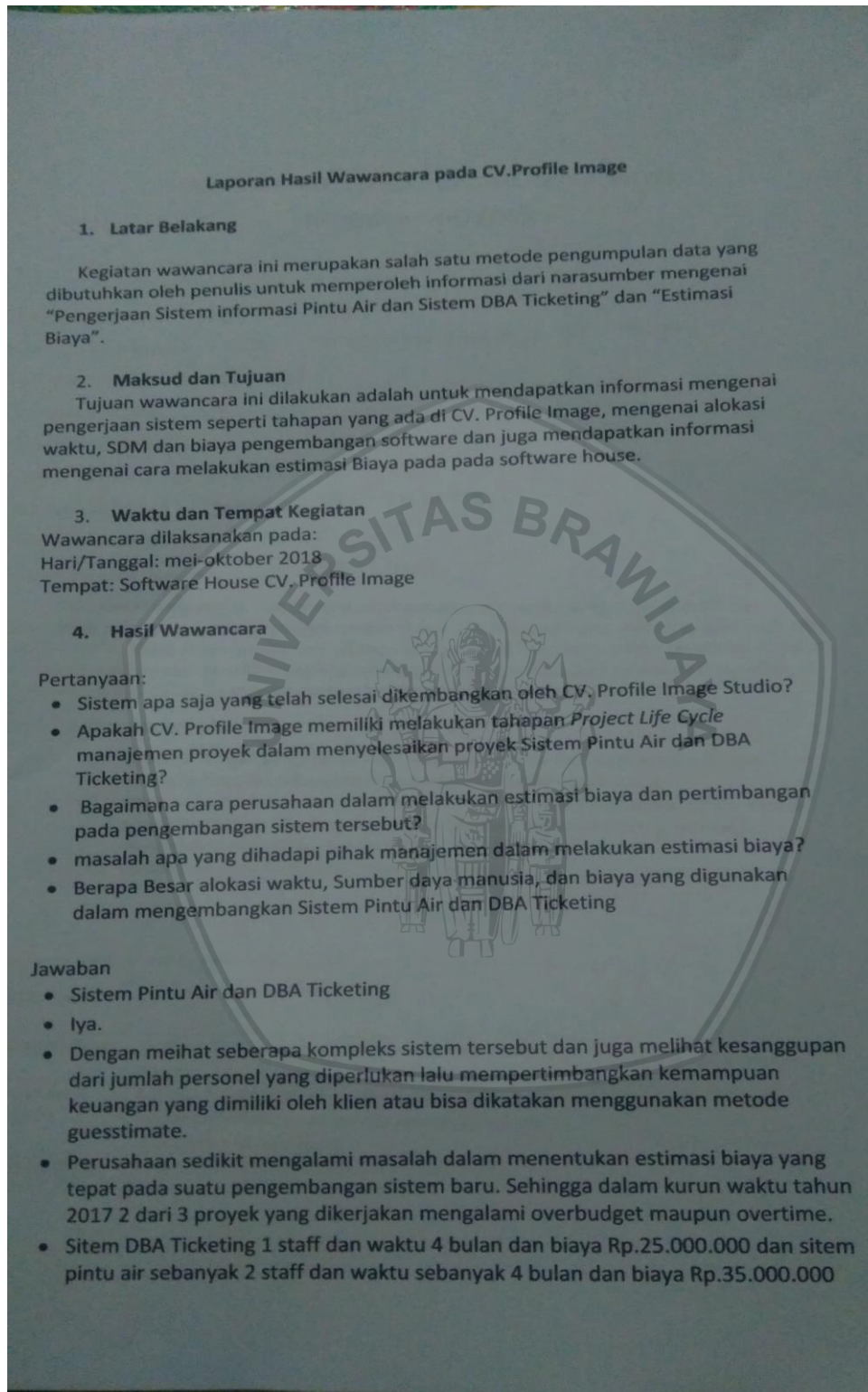


LAMPIRAN A Hasil Wawancara

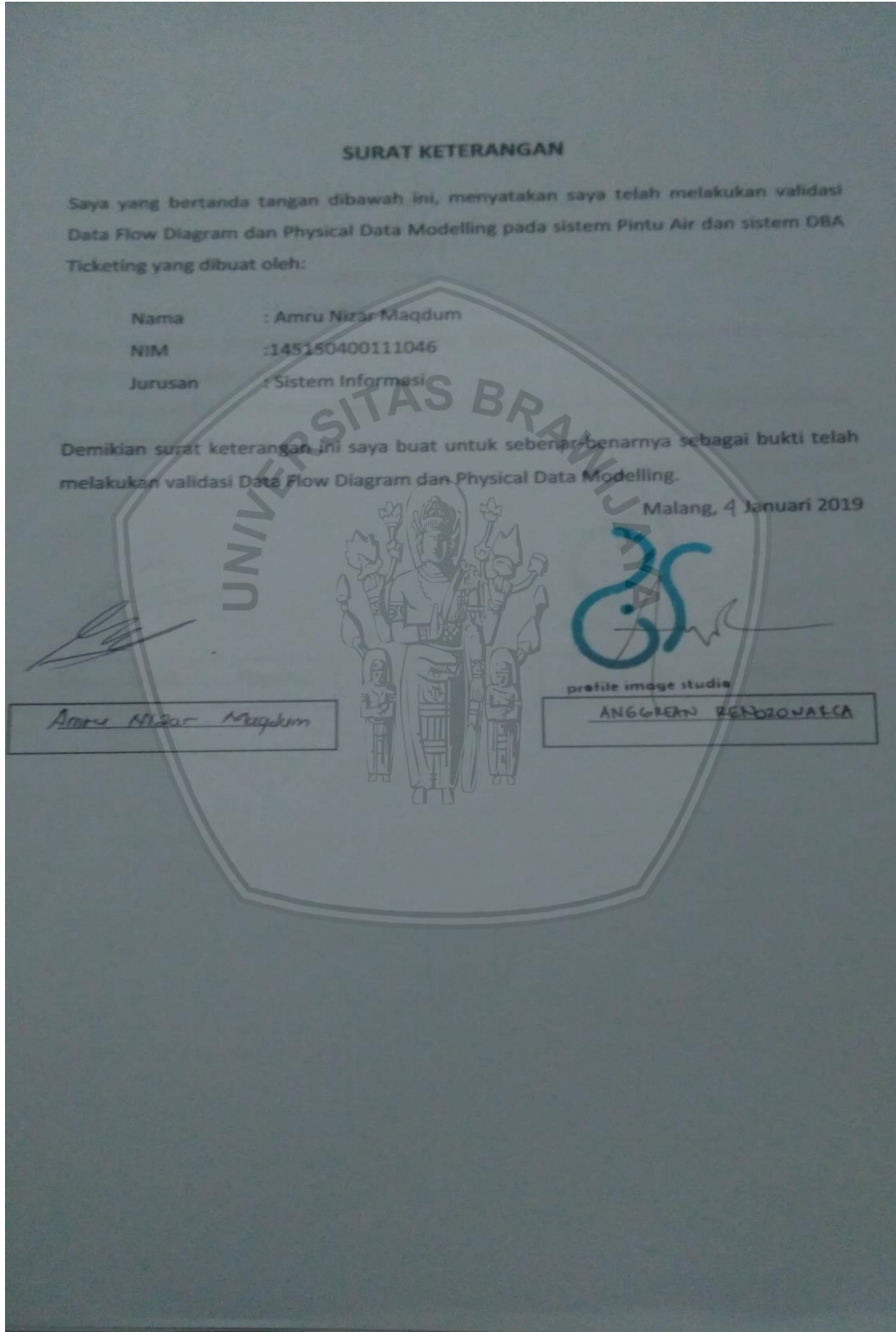
A.1 Hasil Wawancara



A.2 Validasi Wawancara



LAMPIRAN B LEMBAR VALIDASI PDM DAN DATA FLOW DIAGRAM



LAMPIRAN C LEMBAR PENILAIAN SCALE FACTOR

**PENILAIAN FAKTOR SKALA PENGEMBANGAN SISTEM
INFORMASI PINTU AIR Kabupaten MALANG**

Nama : Yusak Wijaya Santosa
 Alamat : Lodon 30 Malang
 No. Telp : 0877 0326 0209
 Pendidikan Terakhir : S1
 Jenis Kelamin : Laki-laki
 Email : yusakwijayas@gmail.com
 Jabatan : Front-end Developer
 Pengalaman di bidang IT : 6 tahun
 Tanggal Pengisian Kuisisioner : 20 April 2018

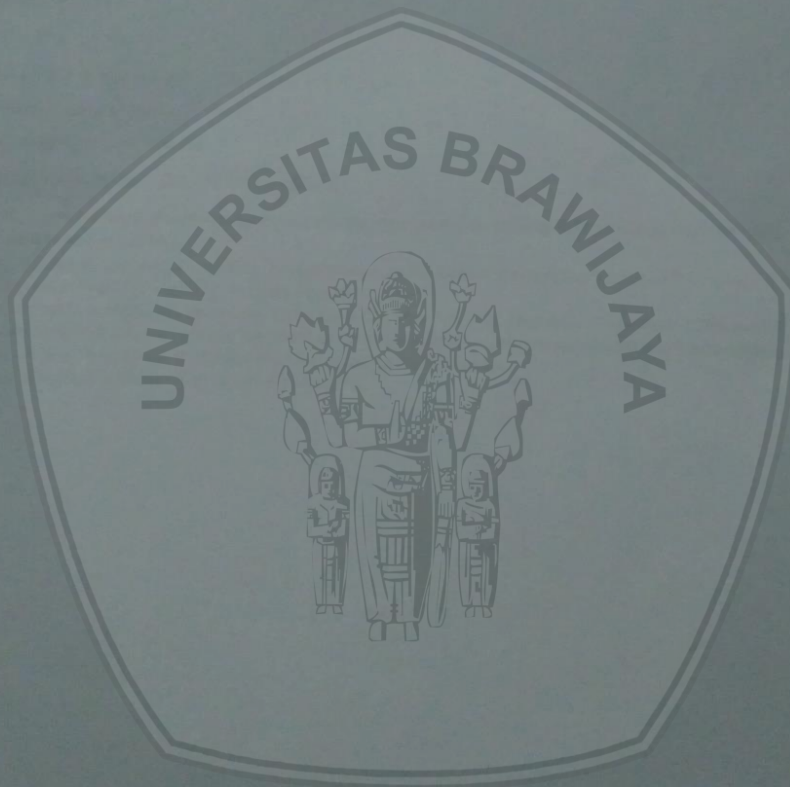
Kuisisioner ini digunakan oleh peneliti untuk mendapatkan faktor yang digunakan dalam perhitungan estimasi biaya dalam pengembangan sistem informasi pintu air yang merupakan salah satu proyek dari CV. Profile Image Studio, dan dijadikan sebagai obyek studi kasus dalam penelitian ini. Hasil penilaian digunakan oleh peneliti sebagai bahan pengerjaan tugas akhir, oleh karena itu diharapkan kerjasamanya bagi Bapak/Ibu/Saudara untuk mengisi kuisisioner ini dengan jujur.

Pentunjuk pengisian kuisisioner :

1. Isi sesuai dengan pengetahuan dan pengalaman anda dalam pengerjaan system informasi Pintu Air kabupaten Malang oleh CV. Profile Image Studio
2. Pilih salah satu skala, yaitu 0, 1, 2, 3, 4, atau 5 untuk setiap pernyataan dalam lembar penelitian. Dimana nilai 0 menandakan nilai paling minimum sedangkan nilai 5 menandakan nilai paling maksimal
3. Centang lah pada kolom yang telah disediakan

Faktor Skala	Deskripsi	0	1	2	3	4	5
<i>Precedentedness</i>	Kualitas pengalaman organisasi dalam menyelesaikan proyek					✓	
<i>Development Flexibility</i>	Kualitas Fleksibilitas klien dalam menentukan dan mengkomunikasikan kebutuhan perangkat lunak kepada tim pengembang				✓		
<i>Risk resolution</i>	Kualitas manajemen resiko yang dimiliki oleh						

	organisasi dalam menyelesaikan proyek perangkat lunak						✓	
<i>Team Cohesion</i>	Kualitas tim pengembang dalam berkomunikasi dan bekerja sama mengembangkan perangkat lunak						✓	
<i>Process Maturity</i>	Kualitas kematangan proses organisasi dalam pengembangan perangkat lunak						✓	



LAMPIRAN D EFFORT MULTIPLIER

**PENILAIAN KUESIONER EFFORT MULTIPLIER PENGEMBANGAN SISTEM
INFORMASI PINTU AIR Kabupaten Malang**

Nama : ANGGREAN RENOSAMARA
Alamat : JL. AKORDION, PERUM SUMI TUGGULWULUNG INDAH G/8
No. Telp : 081357177085
Pendidikan Terakhir : S1
Jenis Kelamin : LAFI LAFI
Email : reno@profileimage.studio
Jabatan : MANAJE OPERASIONAL
Pengalaman di bidang IT : 4 TAHUN
Tanggal Pengisian Kuisisioner : 20 April 2018

1. Silahkan memberikan penilaian pada setiap pernyataan dan setiap faktor eksponen dan juga faktor penyesuaian usaha.
2. Isi sesuai dengan pengetahuan dan pengalaman anda dalam pengerjaan sistem informasi Pintu Air kabupaten Malang oleh CV. Profile Image Studio.
3. Pilih salah satu skala, yaitu Very Low (VL), Low (L), Normal (N), High (H), Very High (VH) untuk setiap pernyataan dalam lembar penelitian. Dimana setiap nilai menandakan perbedaan arti yang akan dideskripsikan di dalam masing-masing pernyataan.
4. Centang lah pada kolom yang telah disediakan.

Nama	Deskripsi	VL	L	N	H	VH	EH
RELY	<p>Terkait sejauh mana perangkat lunak menjalankan aplikasi sesuai fungsinya selama periode waktu.</p> <p>Kriteria Penilaian: Very Low: Jika keagalannya hanya berdampak pada ketidaknyamanan pengguna Low: Jika keagalannya rendah, kerugian mudah diperoleh kembali Normal: Jika keagalannya sedang, kerugian mudah diperoleh kembali High: Jika keagalannya merugikan dalam hal finansial Very High: Jika keagalannya berisiko bagi kehidupan manusia</p>	✓					
DATA	<p>Ukuran <i>database</i> yang digunakan. Ukuran dapat dihitung menggunakan D/P</p> <p>Kriteria Penilaian: Low: jika DB bytes/pgm SLOC < 10 Normal: jika 10 D/P < 100 High: jika 100 D/P < 1000 Very High: jika DB bytes/pgm SLOC > 1000</p>						
CPLX	<p>Perangkat lunak dan perangkat keras dalam melakukan tugasnya seperti platform (arsitektur, sistem operasi, bahasa pemrograman dan antarmuka yang terkait), sistem manajemen <i>database</i>, <i>browser</i> yang sesuai digunakan dalam menjalankan aplikasi ini</p> <p>Kriteria Penilaian: Very Low: perangkat keras dan perangkat lunak tidak berjalan dengan baik Low: terdapat beberapa eror saat menjalankan perangkat lunak Normal: perangkat lunak berjalan dengan sebagaimana mestinya High: perangkat lunak berjalan dengan baik dan tidak ada eror sama sekali Very High: perangkat lunak dijalankan dan saat pengujian tidak terdapat eror</p>					✓	



<p>DOCU</p>	<p>Kesesuaian dokumentasi proyek terhadap kebutuhan siklus hidup perangkat lunak</p> <p>Kriteria Penilaian: Very Low: hanya beberapa siklus hidup yang didokumentasikan Low: hanya sebagian kebutuhan siklus hidup yang didokumentasikan Normal: dokumentasi sesuai dengan siklus hidup perangkat lunak High: berlebihan dalam dokumentasi siklus hidup Very High: sangat berlebihan dalam dokumentasi siklus hidup</p>					
<p>RUSE</p>	<p>Dokumentasi yang diperlukan untuk mengembangkan komponen yang dimaksudkan untuk digunakan kembali pada proyek-proyek yang sedang berjalan atau proyek di masa mendatang.</p> <p>Kriteria Penilaian: Low: dokumentasi tidak digunakan dalam proyek, program, dan produk lain. Nominal: dokumentasi akan digunakan dalam proyek lain. High: dokumentasi akan digunakan dalam program lain. Very High: dokumentasi akan digunakan dalam produk lain. Extra High: dokumentasi akan digunakan dalam produk lini lain.</p>					
<p>ACAP</p>	<p>Kemampuan personel dalam analisis dan desain, efisiensi dan ketelitian, serta kemampuan untuk berkomunikasi dan bekerja sama. Dalam hal ini, dapat dinilai dari sertifikasi yang sudah didapatkan personel atau pengalaman kerja tim dalam suatu proyek</p> <p>Kriteria Penilaian: Very Low: 10 proyek terakhir, analisis proyek ini pernah berpartisipasi setidaknya 2 kali Low: 10 proyek terakhir, analisis proyek ini pernah berpartisipasi setidaknya 4 kali Normal: 10 proyek terakhir, analisis proyek ini pernah berpartisipasi setidaknya 6 kali High: 10 proyek terakhir, analisis proyek ini pernah berpartisipasi setidaknya 8 kali</p>					

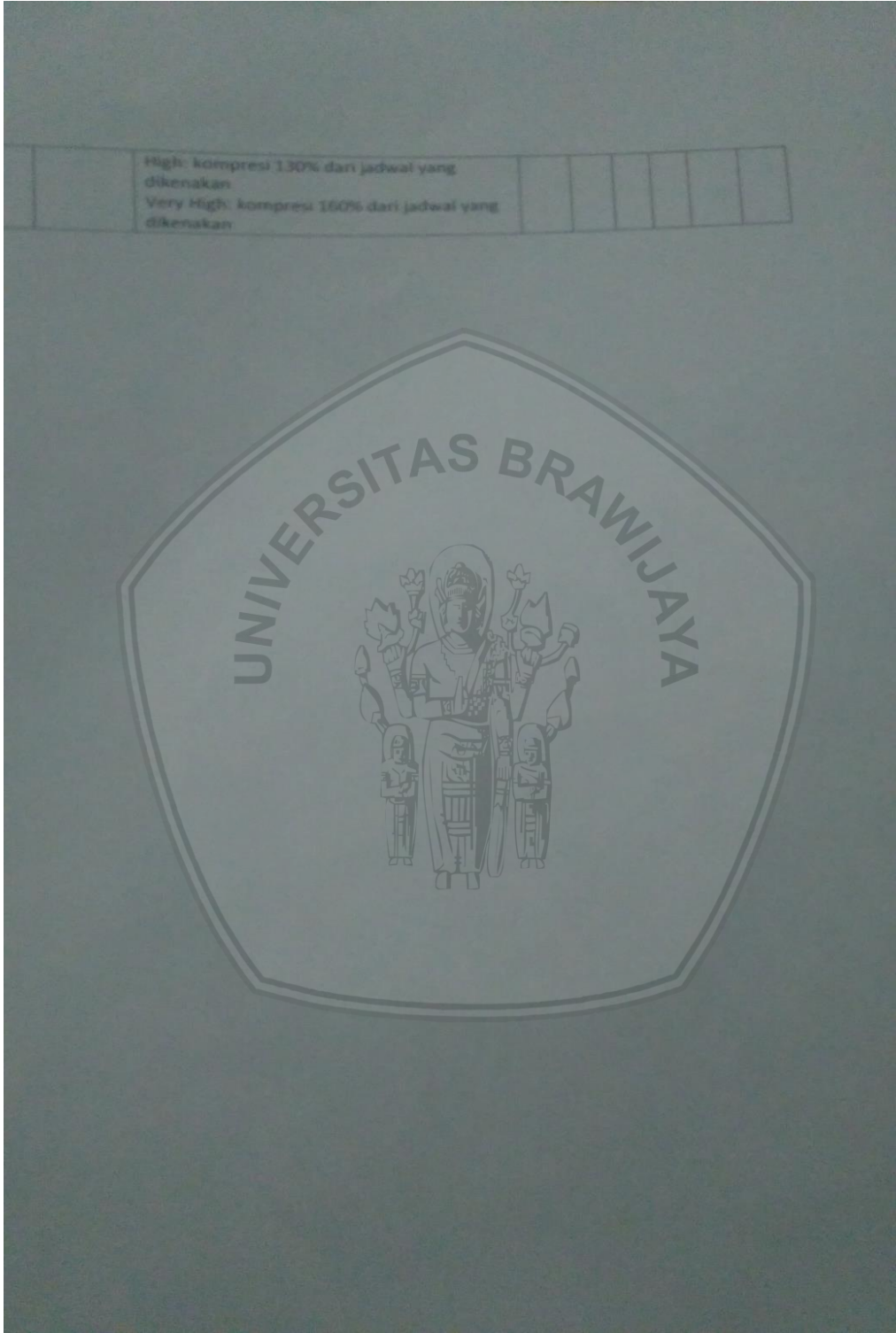


	Very High: 10 proyek terakhir, analisis proyek ini pernah berpartisipasi setidaknya 9 kali								✓
PCON	Terkait dengan pergantian personel tiap tahun pada proyek. Semakin sedikit pergantian maka semakin tinggi skala Kriteria Penilaian: Very Low: 48% per tahun Low: 24% per tahun Normal: 12% per tahun High: 8% per tahun Very High: 3% per tahun							✓	
TIME	Terkait dengan persentase kendala waktu eksekusi yang diharapkan dapat digunakan pada sistem perangkat lunak Kriteria Penilaian: Nominal: 50% dari waktu yang tersedia High: 75% dari waktu yang tersedia Very High: 85% dari waktu yang tersedia Extra High: 95% dari waktu yang tersedia							✓	
STOR	Terkait dengan persentase tingkat kendala penyimpanan utama yang dikenakan pada sistem perangkat lunak. Kriteria Penilaian: Nominal: 50% dari waktu yang tersedia High: 75% dari waktu yang tersedia Very High: 85% dari waktu yang tersedia Extra High: 95% dari waktu yang tersedia							✓	
PVOL	Terkait dengan perubahan yang terjadi pada <i>hardware</i> dan <i>software</i> (OS,DBMS) dalam kurun waktu tertentu. Kriteria Penilaian: Low: jika terjadi perubahan besar dalam 12 bulan, kecil 1 bulan. Normal: jika terjadi perubahan besar dalam 6 bulan, kecil 2 minggu. High: jika terjadi perubahan besar dalam 6 bulan, kecil 1 minggu. Very High: jika terjadi perubahan besar dalam 2 minggu, kecil dalam hari.							✓	
PCAP	Terkait dengan kemampuan <i>programmer</i> dalam efisiensi penulisan kode program, ketelitian dan kemampuan untuk								



TOOL	<p>Terkait dengan kemampuan penggunaan alat pengembangan perangkat lunak pada proyek, seperti mengubah kode yang sederhana menjadi terintegrasi.</p> <p>Kriteria Penilaian: Very Low: edit, kode, debug dan komunikasi melalui telpon, email Low: sederhana, backend case, integrase dalam hal kecil dan komunikasi melalui telpon, fax. Nominal: basic lifecycle tools, integrasi cukup, dan komunikasi melalui email pribadi. High: kuat, mature lifecycle tools, integrasi cukup, dan komunikasi melalui <i>wide band communication</i> (smartphone, dll) Very High: kuat, matang, integrasi cukup, dan komunikasi melalui <i>wide band communication</i> (smartphone, dll), video conference.</p>					✓
SITE	<p>Terkait dengan cara distribusi informasi dalam pengembangan dan bagaimana cara komunikasi yang digunakan dalam pengembangan perangkat lunak pada proyek.</p> <p>Kriteria Penilaian: Very Low: komunikasi melalui telpon, email Low: komunikasi melalui telpon, fax Nominal komunikasi melalui email pribadi. High: komunikasi melalui <i>wide band communication</i> (smartphone, dll) Very High: komunikasi melalui <i>wide band communication</i> (smartphone, dll), video conference. Extra High: interactive multimedia</p>					✓
SCED	<p>Terkait dengan kendala jadwal yang dikenakan pada tim proyek pengembangan perangkat lunak.</p> <p>Kriteria Penilaian: Very Low: kompresi 75% dari jadwal yang dikenakan Low: kompresi 85% dari jadwal yang dikenakan Normal: kompresi 100% dari jadwal yang dikenakan</p>				✓	





High: kompresi 130% dari jadwal yang dikenakan							
Very High: kompresi 160% dari jadwal yang dikenakan							

UNIVERSITAS BRAWIJAYA

