



**PERBANDINGAN EVALUASI BIAYA PENGEMBANGAN
SISTEM ANTRIAN RSUD DR SOETRASNO REMBANG
MENGUNAKAN METODE *USE CASE POINT* DAN
*FUNCTION POINT***

(Studi Kasus: CV Pabrik Teknologi)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Dhya Fairuzu Zahiroh

NIM: 135150401111027



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2017



PENGESAHAN

Perbandingan Evaluasi Biaya Pengembangan Sistem Antrian RSUD Dr. Soetrasno
Rembang Menggunakan Metode *Use Case Point* dan *Function Point*
(Studi Kasus: CV Pabrik Teknologi)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Dhya Fairuzu Zahiroh
NIM: 135150401111027

Skripsi ini telah diuji dan dinyatakan lulus pada
03 Agustus 2017
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

M. Chandra Saputra, S.Kom., M.Eng
NIK: 201609860106 1001

Admaja Dwi Herlambang, S.Pd., M.Pd
NIK: 201609890802 1001



Mengetahui
Rektua Jurusan Sistem Informasi

Herman Tolle, Dr. Eng., S.T, M.T
NIP: 19740823 200012 1 00



IDENTITAS TIM PENGUJI



Nama : Aditya Rachmadi, S.ST., M.Ti

Tempat, Tanggal Lahir: Malang, 21 April 1986

Jenis Kelamin : Laki-laki

Agama : Islam

Nomor Telepon : +62 823 31154613

Email : rachmadi.aditya@ub.ac.id

BIOGRAFI

Lahir di Malang pada tanggal 21 April 1986. Berhasil menyelesaikan studi D4 PENS ITS dan melanjutkan Magister Teknologi Informasi di Universitas Indonesia.

Saat ini sedang aktif sebagai dosen Teknik Informatika UB dengan bidang keminatan RPL, Perencanaan Strategis Sistem Informasi.



Nama : Faizatul Amalia, S.Pd., M.Pd

Tempat, Tanggal Lahir: -

Jenis Kelamin : Perempuan

Agama : Islam

Nomor Telepon : +62 8563071448

Email : faiz_amalia@ub.ac.id

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 pustaka.

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, 03 Agustus 2017



Dhya Fairuzu Zahiroh

NIM: 135150401111027



DAFTAR RIWAYAT HIDUP

Data Pribadi



Nama : Dhya Fairuzu Zahiroh
 Tempat, Tanggal Lahir: Blitar, 22 Juni 1995
 Alamat : Jalan Mertojoyo Barat Perum. Dinoyo
 Royal Park, Malang 65144
 Jenis Kelamin : Perempuan
 Agama : Islam
 Nomor Telepon : +62 812 292 6696
 Email : dhyafairuzu@gmail.com

Latar Belakang Pendidikan

A. Pendidikan Formal

TK Indria Sungai Gerong (1999-2001)
 SD Patra Mandiri 3 Sungai Gerong (2001-2007)
 SMPIT Raudhatul Ulum Indralaya (2007-2008)
 SMP Patra Mandiri 2 Sungai Gerong (2008-2010)
 SMA Negeri 04 Palembang (2010-2013)
 Fakultas Ilmu Komputer Universitas Brawijaya (2013-sekarang)

B. Pendidikan Non Formal

Belum Ada.

Skill dan Keterampilan

1. Keahlian Komputer (MS Word, MS Excel, MS Powerpoint)
2. Membuat *website* atau sistem informasi

Pengalaman Kerja

Belum ada.

Demikian daftar riwayat hidup saya buat dengan sebenar-benarnya.

Malang, 08 Agustus 2017

Dhya Fairuzu Zahiroh



Sembah sujud serta syukur kepada Allah SWT. Taburan cinta dan kasih sayang-Mu telah memberikan kekuatan, membekaliku dengan ilmu serta memperkenalkanku dengan cinta. Atas karunia serta kemudahan yang Engkau berikan akhirnya skripsi ini dapat terselesaikan.

Sholawat dan salam selalu terlimpahkan kehariban Rasulullah Muhammad SAW.

Kupersembahkan karya ini kepada orang yang sangat kukasihi dan kusayangi

M. Chandra Saputra, S.Kom., M.Eng selaku Dosen Pembimbing I dan Admaja Dwi Herlambang, S.Pd., M.Pd selaku Dosen Pembimbing II skripsi yang telah bersedia meluangkan banyak waktu untuk memberikan bimbingan, nasihat, saran dan motivasi sehingga penulis dapat menyelesaikan penulisan skripsi ini dengan baik dan tepat waktu.

Seluruh Civitas Akademika Fakultas Ilmu Komputer, Universitas Brawijaya yang telah bersedia menyalurkan semua ilmu pengetahuan saat penulis mengikuti perkuliahan sehingga dengan bermodalkan ilmu pengetahuan tersebut penulis dapat mengerjakan skripsi ini dan siap turun ke masyarakat.

Orangtua penulis, T. Kholifah Huda dan Siti Ulfah, adik penulis Qotrunnada, Za'faron, Zarabiyu, Wilda dan Fathi yang telah memberikan dukungan baik secara moril maupun materil, yang selalu memanjatkan do'a untuk penulis disetiap sujudnya dan menjadi inspirasi bagi penulis untuk terus semangat dan gigih dalam mengejar cita-cita.

M. Rizqi Zamzami Alnobeta, selaku sahaba dan keluarga yang selalu memberikan dukungan, do'a dan motivasi agar penulis semangat dan pantang menyerah menyelesaikan penulisan skripsi serta teman satu topik yang menjadi rekan diskusi selama proses penulisan skripsi.

ABSTRAK

Manajemen proyek harus mempertimbangkan tiga aspek yaitu ruang lingkup (*scope*), waktu (*time*) dan biaya (*cost*) agar proyek dapat dikatakan sukses. Masalah yang dihadapi *softwarehouse* dalam pengerjaan suatu proyek yaitu dalam perhitungan estimasi biaya, *softwarehouse* belum memiliki standar perhitungan biaya yang baik, maka perlu adanya metode estimasi biaya yang digunakan sebagai standar untuk perhitungan biaya total yang dibutuhkan selama pengembangan perangkat lunak dilakukan. Selain itu, akan lebih baik dalam setiap pengerjaan proyek mengacu pengerjaan yang terstruktur berdasarkan *Work Breakdown Structure* (WBS).

Berdasarkan permasalahan proyek, penelitian ini mengangkat judul Perbandingan Evaluasi Biaya Sistem Antrian RSUD Dr. Soetrasno Rembang Menggunakan Metode *Use Case Point* dan *Function Point* (Studi Kasus: CV Pabrik Teknologi). Penelitian ini menjabarkan tahapan menggunakan pendekatan WBS dan penjadwalan menggunakan *Gantt Chart* karena sebuah WBS merupakan landasan dari perencanaan proyek agar proyek dapat dilaksanakan, dikendalikan dengan lebih efektif serta menggunakan dua metode estimasi biaya yaitu metode *Use Case Point* dan metode *Function Point*.

Perhitungan *Use Case Point* membutuhkan *Use Case Diagram* dan *Use Case Scenario* sehingga menghasilkan *hours of effort* sebesar 2.850 jam kerja dan estimasi biaya total sebesar Rp 157.843.750,00. Sedangkan perhitungan *Function Point* membutuhkan *Data Flow Diagram* sehingga menghasilkan estimasi *effort* (usaha) yaitu 24 orang selama 9 bulan dan estimasi biaya total sebesar Rp 200.650.000,00. Penelitian ini akan memberikan bahan pertimbangan kepada CV Pabrik Teknologi dalam menyelesaikan masalah yaitu dengan menggunakan metode *Use Case Point* dan *Function Point* dan analisis perbandingan antara metode *Use Case Point*, *Function Point* dan *Guesstimate*.

Kata kunci: Estimasi biaya, *Use Case Point*, *Function Point*, *Guesstimate*

ABSTRACT

Project Management consist of scope, time and cost. The problem in a project is calculation of the estimated costs doesn't have standards for calculating the cost, then there is need for the method of estimated costs of which is used as standards for the calculation of the total expenses required for the development of software. In addition, it would be better in every work projects refers to the execution of a structured on the Work Breakdown Structure (WBS).

*Based on the project, research was lifting the title *Perbandingan Evaluasi Biaya Sistem Antrian RSUD Dr. Soetrasno Rembang Menggunakan Metode Use Case Point dan Function Point (Studi Kasus: CV Pabrik Teknologi)*. This research lays out the phase using WBS and scheduling using Gantt Chart because of a WBS is the basis of planning project for the project can be held, controlled more effectively and also using two methods of the estimated costs that is Use Case Point and Function Point.*

The calculation of the Use Case Point needs Use Case Diagrams and Use Case Scenario so as to produce hours of effort is 2.850 hours and the estimated total cost is Rp 157.843.750, 00. While Function Point needs Data Flow diagram so as to produce the effort 24 people during the nine months and the estimated total cost is Rp 200.650.000, 00. The result can helps CV Pabrik Teknologi in solving the problem in the form of analysis of the comparison between the Use Case Point, Function Point and Guesstimate.

Keywords: Cost estimation, Use Case Point, Function Point, Guesstimate



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena berkat rahmat dan hidayah-Nya serta atas junjungan Nabi Muhammad SAW, sehingga penulis diberi kemudahan, kekuatan, kesabaran dan hikmah dalam menyelesaikan skripsi dengan judul “PERBANDINGAN EVALUASI BIAYA PENGEMBANGAN SISTEM ANTRIAN RSUD DR SOETRASNO REMBANG MENGGUNAKAN METODE USE CASE POINT DAN FUNCTION POINT (Studi Kasus: CV Pabrik Teknologi)” dengan baik dan benar yang disusun untuk memenuhi syarat tugas akhir pada program studi Sistem Informasi dan mencapai gelar Sarjana Komputer di Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

Skripsi ini disusun setelah melakukan *survey* ke beberapa *softwarehouse* yang tersebar di kota Malang. Penulis dapat menyelesaikan skripsi ini juga tidak terlepas dari dukungan banyak pihak yang memberikan bantuan. Oleh karena itu, pada kesempatan kali ini perkenankanlah penulis mengucapkan terimakasih kepada:

1. M. Chandra Saputra, S.Kom., M.Eng selaku Dosen Pembimbing I dan Admaja Dwi Herlambang, S.Pd., M.Pd selaku Dosen Pembimbing II skripsi yang telah bersedia meluangkan banyak waktu untuk memberikan bimbingan, nasihat, saran dan motivasi sehingga penulis dapat menyelesaikan penulisan skripsi ini dengan baik dan tepat waktu.
2. Wayan Firdaus Mahmudy, S.Si, M.T., Ph.D., Heru Nurwarsito, Ir., M.Kom., Marji, Drs., M.T., dan Edy Santosa, S.Si, M.Kom. selaku Dekan, Wakil Dekan I, Wakil Dekan II dan Wakil Dekan III Fakultas Ilmu Komputer, Universitas Brawijaya.
3. Suprpto, S.T, M.T dan Ismiarta Aknuranda, S.T, M.Sc., Ph.D selaku Ketua dan Sekretaris Program Studi Sistem Informasi Fakultas Ilmu Komputer, Universitas Brawijaya.
4. Herman Tolle, Dr. Eng., S.T, M.T selaku Pembimbing Akademik.
5. Seluruh Civitas Akademika Fakultas Ilmu Komputer, Universitas Brawijaya yang telah bersedia menyalurkan semua ilmu pengetahuan saat penulis mengikut perkuliahan sehingga dengan bermodalkan ilmu pengetahuan tersebut penulis dapat mengerjakan skripsi ini dan siap turun ke masyarakat.
6. Farid Angga Pribadi, S.Kom, Firman Jati Pamungkas, S.Kom dan Masrur Anwar selaku *Project Manager* dan *Programmer* dari *softwarehouse* CV Pabrik Teknologi yang telah meluangkan waktu dan bersedia membantu penulis dalam proses pengumpulan data skripsi.
7. Orangtua penulis, T. Kholifah Huda dan Siti Ulfah, adik penulis Qotrunnada, Za’faron, Zarabiyu, Wilda dan Fathi yang telah memberikan dukungan baik secara moril maupun materil, yang selalu memanjatkan do’a untuk penulis disetiap sujudnya dan menjadi inspirasi bagi penulis untuk terus semangat dan gigih dalam mengejar cita-cita.
8. M. Rizqi Zamzami Alnobeta, selaku sahaba dan keluarga yang selalu memberikan dukungan, do’a dan motivasi agar penulis semangat dan pantang menyerah menyelesaikan penulisan skripsi serta teman satu topik yang menjadi rekan diskusi selama proses penulisan skripsi.



9. Novelasari, Zainal Arifin, Ivan Agustinus, Intan Rumaysha, Dinda Agnes, Faisyal, Irma, Alqis, Bunga Boru, Agung, Akbar M, Fatikh, Sukma, Eric yang telah memberikan banyak dukungan serta telah menjadi sahabat dan keluarga di kota perantauan dan memberikan banyak pengalaman dalam berorganisasi.
10. Teman-teman Fakultas Ilmu Komputer dan Universitas Brawiaya yang telah memberikan banyak pengalaman dan berbagi ilmu kepada penulis.
11. Serta semua pihak yang membantu secara langsung maupun tidak langsung yang tidak dapat disebutkan satu per satu yang telah berjasa dalam membantu penulis dalam menyelesaikan penulisan skripsi.

Malang, 03 Agustus 2017

Dhya Fairuzu Zahiroh

dhyafairuzu@gmail.com



DAFTAR ISI

LEMBAR PERSETUJUAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xvi
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN.....	18
1.1 Latar belakang.....	17
1.2 Rumusan masalah.....	19
1.3 Tujuan	19
1.4 Manfaat.....	19
1.5 Batasan masalah	19
1.6 Sistematika pembahasan.....	19
BAB 2 LANDASAN KEPUSTAKAAN	21
2.1 Kajian Pustaka.....	21
2.2 Profil CV Pabrik Teknologi.....	22
2.2.1 Visi CV Pabrik Teknologi.....	23
2.2.2 Misi CV Pabrik Teknologi.....	23
2.3 Manajemen Proyek.....	23
2.4 Project Cost Management	24
2.5 Software Cost Management	24
2.6 <i>Project Life Cycle</i> (PLC).....	25
2.7 <i>Software Development Life Cycle</i> (SDLC).....	26
2.8 Work Breakdown Structure	26
2.9 Gantt Chart	27
2.10 Use Case.....	27
2.11 Metode Use Case Point	28



2.11.1 Unadjusted Actor Weight (UAW).....	28
2.11.2 Unadjusted Use Case Weight (UUCW).....	29
2.11.3 Unadjusted Use Case Point (UUCP)	29
2.11.4 Technical Complexity Adjustment Factor (TCF).....	29
2.11.5 Environment Complexity Factor (ECF).....	30
2.11.6 Estimasi Effort	30
2.11.7 Estimasi Biaya.....	30
2.12 Metode Function Point.....	31
2.12.1 Menghitung Nilai Function Point	32
2.12.2 Estimasi Effort (Usaha).....	37
2.12.3 Estimasi Biaya.....	38
2.13 Data Flow Diagram.....	39
2.14 Peran dan Tanggung Jawab Profesi TI	40
2.15 Metode Guesstimate	40
BAB 3 METODOLOGI	41
3.1 Studi Pustaka.....	42
3.2 Pengumpulan Data	42
3.2.1 Wawancara	42
3.2.2 Lembar penilaian.....	42
3.3 Analisis Hasil.....	43
3.4 Kesimpulan.....	47
BAB 4 HASIL.....	48
4.1 Mekanisme Wawancara	48
4.2 Hasil Wawancara	48
4.3 Mekanisme Lembar Penilaian.....	50
4.4 Hasil Lembar Penilaian.....	50
4.5 <i>User Story</i>	51
4.6 Pembuatan Use Case Diagram.....	52
4.6.1 Use Case Diagram	53
4.6.2 Use Case Scenario	54
4.7 Pembuatan Data Flow Diagram.....	66
4.7.1 Context Diagram	66



4.7.2 Data Flow Diagram Level 1.....	67
4.7.3 Data Flow Diagram Level 2.....	68
4.7.3.1 <i>Data Flow Diagram</i> Level 2 Melakukan Login	68
4.7.3.2 <i>Data Flow Diagram</i> Level 2 Mengelola SMS Gateway....	69
4.7.3.3 <i>Data Flow Diagram</i> Level 2 Mengelola Data User.....	70
4.7.3.4 <i>Data Flow Diagram</i> Level 2 Mengelola Data Antrian	70
4.7.3.5 <i>Data Flow Diagram</i> Level 2 Mengelola Data Admin.....	71
4.7.3.6 <i>Data Flow Diagram</i> Level 2 Mengelola Monitor.....	72
BAB 5 PEMBAHASAN.....	73
5.1 <i>Work Breakdown Structure</i> (WBS).....	73
5.2 Perhitungan Biaya dengan Metode <i>Use Case Point</i>	76
5.2.1 Perhitungan <i>Unadjusted Use Case Point</i> (UUCP).....	76
5.2.1.1 Perhitungan <i>Unadjusted Actor Weight</i> (UAW).....	77
5.2.1.2 Perhitungan <i>Unadjusted Use Case Weight</i> (UUCW).....	77
5.2.2 Perhitungan <i>Technical and Environment Complexity Factor</i>	85
5.2.2.1 <i>Technical Complexity Factor</i> (TCF)	85
5.2.2.2 <i>Environment Complexity Factor</i> (ECF).....	85
5.2.3 Perhitungan Nilai <i>Use Case Point</i> (UCP)	86
5.2.4 Perhitungan <i>Hours of Effort</i>	86
5.2.5 Perhitungan Biaya Total.....	87
5.3 Perhitungan Biaya dengan Metode <i>Function Point</i>	89
5.3.1 Menentukan Tipe Fungsi Pengguna.....	89
5.3.2 Menentukan Bobot Kompleksitas <i>Function Point</i>	90
5.3.3 Perhitungan <i>Unadjusted Function Point</i> (UFP).....	91
5.3.4 Perhitungan <i>Technical Complexity Factor</i> (TCF).....	91
5.3.5 Perhitungan Nilai <i>Function Point</i> (FP).....	92
5.3.6 Perhitungan <i>Effort</i>	93
5.3.7 Perhitungan Biaya Total.....	93
5.4 Analisis Perhitungan dengan <i>Guesstimate</i>	95
5.5 Analisis Hasil <i>Use Case Point</i> , <i>Function Point</i> dan <i>Guesstimate</i>	96
BAB 6 Penutup	99
6.1 Kesimpulan.....	99



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	21
Tabel 2.2 Komponen <i>Use Case Diagram</i>	27
Tabel 2.3 Tipe dan Bobot Aktor pada <i>Use Case Point</i>	29
Tabel 2.4 Tipe dan Bobot <i>Use Case</i> pada <i>Use Case Point</i>	29
Tabel 2.5 <i>Technical Factors</i>	29
Tabel 2.6 <i>Environment Factors</i>	30
Tabel 2.7 Bobot Kompleksitas <i>External Input</i>	32
Tabel 2.8 Bobot Kompleksitas <i>External Output</i> dan <i>External Inquiries</i>	33
Tabel 2.9 Bobot Kompleksitas <i>Logical Internal File</i> dan <i>External Interface File</i> ...	33
Tabel 2.10 Perhitungan <i>Unadjusted Function Point</i>	33
Tabel 2.11 Aturan Perhitungan Bobot Kompleksitas.....	34
Tabel 2.12 Bobot <i>Technical Complexity Factor</i>	37
Tabel 2.13 <i>Technical Factors</i>	37
Tabel 2.14 <i>Jone's First Order Estimate Exponent</i>	38
Tabel 2.15 Komponen <i>Data Flow Diagram</i>	39
Tabel 3.1 Alur Perhitungan <i>Use Case Point</i>	43
Tabel 3.2 Alur Perhitungan <i>Function Point</i>	46
Tabel 4.1 Pengguna Pasien	49
Tabel 4.2 Pengguna Admin.....	49
Tabel 4.3 Pengguna Admin Counter	50
Tabel 4.4 Pengguna Admin, Admin Counter, Pasien	50
Tabel 4.5 Skenario <i>Use Case Login</i>	54
Tabel 4.6 Skenario <i>Use Case Lihat Data User</i>	54
Tabel 4.7 Skenario <i>Use Case Tambah Data User</i>	55
Tabel 4.8 Skenario <i>Use Case Ubah Data User</i>	55
Tabel 4.9 Skenario <i>Use Case Hapus Data User</i>	56
Tabel 4.10 Skenario <i>Use Case Baca Pesan Masuk</i>	56
Tabel 4.11 Skenario <i>Use Case Hapus Pesan Masuk</i>	57
Tabel 4.12 Skenario <i>Use Case Baca Pesan Keluar</i>	57
Tabel 4.13 Skenario <i>Use Case Hapus Pesan Keluar</i>	58



Tabel 4.14 Skenario <i>Use Case</i> Lihat Kuota SMS	58
Tabel 4.15 Skenario <i>Use Case</i> Ubah Kuota SMS	59
Tabel 4.16 Skenario <i>Use Case</i> Lihat Data Antrian	59
Tabel 4.17 Skenario <i>Use Case</i> Panggil Ulang Pasien.....	60
Tabel 4.18 Skenario <i>Use Case</i> Panggil Pasien Selanjutnya.....	60
Tabel 4.19 Skenario <i>Use Case</i> Tambah Data Video	61
Tabel 4.20 Skenario <i>Use Case</i> Hapus Data Video	61
Tabel 4.21 Skenario <i>Use Case</i> Tambah Data <i>Running Text</i>	62
Tabel 4.22 Skenario <i>Use Case</i> Hapus Data <i>Running Text</i>	62
Tabel 4.23 Skenario <i>Use Case</i> Mengambil Nomor Antrian.....	63
Tabel 4.24 Skenario <i>Use Case</i> Mengecek Nomor Antrian	63
Tabel 4.25 Skenario <i>Use Case</i> Melakukan Registrasi	63
Tabel 4.26 Skenario <i>Use Case</i> Lihat Data Admin	64
Tabel 4.27 Skenario <i>Use Case</i> Tambah Data Admin	64
Tabel 4.28 Skenario <i>Use Case</i> Ubah Data Admin.....	65
Tabel 4.29 Skenario <i>Use Case</i> Hapus Data Admin	66
Tabel 5.1 Dekripsi Aktor Pada Sistem	77
Tabel 5.2 Perhitungan Total <i>Unadjusted Actor Weight (UAW)</i>	77
Tabel 5.3 Transaksi <i>Use Case</i> Login.....	78
Tabel 5.4 Transaksi <i>Use Case</i> Lihat Data <i>User</i>	78
Tabel 5.5 Transaksi <i>Use Case</i> Tambah Data <i>User</i>	78
Tabel 5.6 Transaksi <i>Use Case</i> Ubah Data <i>User</i>	79
Tabel 5.7 Transaksi <i>Use Case</i> Hapus Data <i>User</i>	79
Tabel 5.8 Transaksi <i>Use Case</i> Baca Pesan Masuk	79
Tabel 5.9 Transaksi <i>Use Case</i> Hapus Pesan Masuk	80
Tabel 5.10 Transaksi <i>Use Case</i> Baca Pesan Keluar	80
Tabel 5.11 Transaksi <i>Use Case</i> Hapus Pesan Keluar	80
Tabel 5.12 Transaksi <i>Use Case</i> Lihat Kuota SMS	80
Tabel 5.13 Transaksi <i>Use Case</i> Ubah Kuota SMS	80
Tabel 5.14 Transaksi <i>Use Case</i> Lihat Data Antrian	81
Tabel 5.15 Transaksi <i>Use Case</i> Panggil Ulang Pasien.....	81
Tabel 5.16 Transaksi <i>Use Case</i> Panggil Pasien Selanjutnya	81



Tabel 5.17 Transaksi <i>Use Case</i> Tambah Data Video	81
Tabel 5.18 Transaksi <i>Use Case</i> Hapus Data Video	82
Tabel 5.19 Transaksi <i>Use Case</i> Tambah Data <i>Running Text</i>	82
Tabel 5.20 Transaksi <i>Use Case</i> Hapus Data <i>Running Text</i>	82
Tabel 5.21 Transaksi <i>Use Case</i> Mengambil Nomor Antrian.....	82
Tabel 5.22 Transaksi <i>Use Case</i> Mengecek Nomor Antrian	82
Tabel 5.23 Transaksi <i>Use Case</i> Melakukan Registrasi.....	83
Tabel 5.24 Transaksi <i>Use Case</i> Lihat Data Admin	83
Tabel 5.25 Transaksi <i>Use Case</i> Tambah Data Admin	83
Tabel 5.26 Transaksi <i>Use Case</i> Ubah Data Admin.....	84
Tabel 5.27 Transaksi <i>Use Case</i> Hapus Data Admin	84
Tabel 5.28 Perhitungan Total <i>Unadjusted Use Case Weight (UUCW)</i>	84
Tabel 5.29 Perhitungan <i>Technical Complexity Factor</i>	85
Tabel 5.30 Perhitungan <i>Environment Complexity Factor</i>	86
Tabel 5.31 Perhitungan <i>Use Case Point (UCP)</i>	86
Tabel 5.32 <i>Hours of Effort</i> ke Dalam 3 Aktivitas	87
Tabel 5.33 Penetapan Standar Gaji.....	88
Tabel 5.34 Perhitungan Biaya Total	88
Tabel 5.35 Tipe Fungsi Pengguna Pada Sistem Antrian RSUD Dr. Soetrasno Rembang	89
Tabel 5.36 Penentuan FTR, DET dan RET Sistem Antrian RSUD Dr. Soetrasno Rembang	90
Tabel 5.37 Perhitungan <i>Unadjusted Function Point (UFP)</i>	91
Tabel 5.38 Perhitungan <i>Technical Complexity Factor</i>	92
Tabel 5.39 Eksponen Jones	93
Tabel 5.40 Pembagian Estimasi <i>Effort</i>	93
Tabel 5.41 Penetapan Standar Gaji.....	94
Tabel 5.42 Perhitungan Total Estimasi Biaya	95
Tabel 5.43 Analisis Hasil <i>Use Case Point, Function Point</i> dan <i>Guesstimate</i>	96
Tabel 5.44 Analisis Perbandingan Durasi Pengerjaan.....	97
Tabel 5.45 Analisis Perbandingan Estimasi Biaya	97

DAFTAR GAMBAR

Gambar 2.1 <i>Project Life Cycle</i> dan <i>Software Development Life Cycle</i>	25
Gambar 2.2 Langkah-langkah Metode <i>Use Case Point</i>	28
Gambar 3.1 Alur Penelitian.....	41
Gambar 4.1 Denah RSUD Dr. Soetrasno Rembang.....	48
Gambar 4.2 <i>Use Case Diagram</i> Sistem Antrian RSUD Dr. Soetrasno Rembang	53
Gambar 4.3 <i>Context Diagram</i> Sistem Antrian RSUD Dr. Soetrasno Rembang	66
Gambar 4.4 Data Flow Diagram Level 1 Sistem Antrian RSUD Dr. Soetrasno Rembang	67
Gambar 4.5 <i>Data Flow Diagram Level 2</i> Melakukan <i>Login</i> Sistem Antrian RSUD Dr. Soetrasno Rembang	68
Gambar 4.6 <i>Data Flow Diagram Level 2</i> Mengelola <i>SMS Gateway</i> Sistem Antrian RSUD Dr. Soetrasno Rembang.....	69
Gambar 4.7 <i>Data Flow Diagram Level 2</i> Mengelola <i>Data User</i> Sistem Antrian RSUD Dr. Soetrasno Rembang.....	70
Gambar 4.8 <i>Data Flow Diagram Level 2</i> Mengelola <i>Data Antrian</i> Sistem Antrian RSUD Dr. Soetrasno Rembang.....	70
Gambar 4.9 <i>Data Flow Diagram Level 2</i> Mengelola <i>Data Admin</i> Sistem Antrian RSUD Dr. Soetrasno Rembang.....	71
Gambar 4.10 <i>Data Flow Diagram Level 2</i> Mengelola <i>Monitor</i>	72
Gambar 5.1 <i>Work Breakdown Structure Guesstimate</i>	73
Gambar 5.2 <i>Work Breakdown Structure Use Case Point dan Function Point</i>	75
Gambar 5.3 Langkah-langkah Metode <i>Use Case Point</i>	76



BAB 1 PENDAHULUAN

1.1 Latar belakang

Semakin pesatnya perkembangan teknologi informasi saat ini, maka semakin banyak pula proyek pengembangan perangkat lunak bermunculan. Dari tahun 2013 hingga saat ini, CV Pabrik Teknologi mengalami peningkatan tiga kali lipat per tahunnya dalam melakukan pengembangan perangkat lunak dalam bentuk *website* maupun *mobile apps* baik milik instansi atau perusahaan maupun milik pribadi. Oleh karena itu, perlu adanya manajemen proyek perangkat lunak yang baik agar teknologi informasi yang dikembangkan sesuai dengan kebutuhan pengguna. Di dalam manajemen proyek terdapat proses perencanaan, pengorganisasian, pengarahan dan pengendalian sumber daya organisasi agar tercapainya tujuan dengan sumber daya tertentu dan waktu tertentu (Santosa, 2003). Selain itu manajemen proyek harus mempertimbangkan tiga aspek yaitu ruang lingkup (*scope*), waktu (*time*) dan biaya (*cost*) agar proyek dapat dikatakan sukses (Schwalbe, 2014).

Salah satu faktor penting untuk menentukan keberhasilan suatu proyek pembuatan perangkat lunak yaitu estimasi biaya. Estimasi biaya merupakan proses memprediksi usaha yang dibutuhkan untuk mengembangkan perangkat lunak (Fan et al, 2015). Selain itu, akan lebih baik dalam setiap pengerjaan proyek berdasarkan *Work Breakdown Structure* (WBS) untuk mendetailkan setiap pengerjaan proyek. WBS merupakan dasar perencanaan sebuah proyek agar proyek tersebut berjalan dengan efektif dikarenakan pada WBS proyek dapat mengidentifikasi, memperkirakan, menjadwalkan dan dianggarkan sesuai dengan kebutuhan proyek. WBS dapat menghubungkan antara ruang lingkup (*scope*), penjadwalan (*schedule*) dan biaya (*cost*) sebuah proyek (Rev, 2003).

Penelitian yang berjudul “Estimasi Biaya Pembuatan Modul *Enterprise Resource Planning* (ERP) untuk Unit Bisnis Pabrik Gula di PT Perkebunan XYZ Dengan Metode *Use Case Point*” menjelaskan bahwa sulitnya pihak manajer proyek dalam menentukan estimasi biaya serta sumber daya lainnya secara tepat agar biaya yang di alokasikan tidak berlebih atau bahkan terlalu sedikit. Oleh karena itu perlu adanya bahan pertimbangan dalam menganalisis estimasi biaya pengembangan ERP (Prassida, 2012).

Sedangkan penelitian yang berjudul “Perkiraan Biaya Pembuatan *Enterprise Resource Planning* (ERP) untuk Unit Bisnis Pabrik Gula Pada PT Perkebunan XYZ Dengan Menggunakan Metode *Function Point*” menjelaskan bahwa terjadinya *under-estimate* dan *over-estimate* terjadi karena adanya kesalahan dalam pengalokasian sumber daya. Oleh karena itu perlu adanya perkiraan biaya dan usaha yang baik serta sesuai dengan kebutuhan (Daniari, 2011).

Berdasarkan pada Lampiran A, masalah yang dihadapi CV Pabrik Teknologi dalam pengerjaan suatu proyek yaitu dalam perhitungan estimasi biaya belum memiliki standar perhitungan biaya yang baik, maka perlu adanya metode estimasi biaya yang digunakan sebagai standar untuk perhitungan biaya total yang dibutuhkan selama pengembangan perangkat lunak dilakukan. CV Pabrik Teknologi hanya sebatas memperkirakan biaya proyek dengan cara perkiraan atau *Guesstimate*. Metode *Guesstimate* merupakan menentukan estimasi jadwal, anggaran proyek dengan menebak atau hanya memperkirakan (*picking number out of the air*) (Marchewka, 2003). Selain itu dalam manajemen proyek, CV Pabrik Teknologi hanya menggunakan *Excel*, maka akan lebih baik ketika pengerjaan proyek



mengacu kepada *Work Breakdown Structure* (WBS) agar pengerjaan proyek lebih terstruktur serta menggunakan *Gantt Chart* agar dapat memantau kemajuan proyek.

Penelitian ini menggunakan proyek Sistem Antrian RSUD Dr. Soetrasno Rembang yang yang di kembangkan di bulan April tahun 2017 oleh CV Pabrik Teknologi. Berdasarkan permasalahan proyek, penelitian ini menjabarkan penjadwalan berdasarkan WBS menggunakan *Gantt Chart*. Seluruh pekerjaan yang terkandung dalam WBS akan diidentifikasi, diperkirakan dan dijadwalkan (Rev, 2003). Kemudian penelitian ini juga menggunakan dua metode estimasi biaya yaitu metode *Use Case Point* dan metode *Function Point*. *Use Case Point* merupakan metode yang dikembangkan oleh *Gustav Karner* pada tahun 1993 dari *Rational Software*. Metode ini bertujuan untuk menyediakan metode estimasi yang berorientasi pada objek proyek dan merupakan adaptasi dari *Function Point Analysis* (FPA) (Ribu, 2001). Perhitungan *Use Case Point* membutuhkan pendekatan yang terdokumentasi dengan baik untuk memperkirakan aktivitas apa saja yang terjadi selama pengembangan *software* menggunakan *Use Case Diagram* serta *Use Case Scenario*. *Use Case Point* digunakan untuk menerjemahkan kebutuhan fungsional perangkat lunak yang setelah itu dapat dilakukan perhitungan estimasi *effort* dan biaya total. Estimasi *effort* atau usaha merupakan proses perkiraan waktu dan biaya yang dibutuhkan untuk menyelesaikan proyek atau tugas dengan mempertimbangkan permintaan *stakeholder* (Dalal, 2015). Perhitungan matematika yang digunakan oleh metode *Use Case Point* akan menghasilkan total *hours of effort* dan estimasi biaya total dalam satuan Rupiah. Sedangkan metode *Function Point* merupakan metode yang diperkenalkan oleh *Allan J. Albercht* pada tahun 1979. Metode ini menggunakan pendekatan berorientasi pada fungsionalitas *software* menggunakan *Data Flow Diagram* untuk memprediksi estimasi usaha yang terkait dengan pengembangan *software*. Perhitungan matematika yang digunakan dalam metode *Function Point* akan menghasilkan estimasi *effort* dan estimasi biaya total dalam satuan Rupiah (Longstreet, 2005).

Penelitian ini akan memberikan bahan pertimbangan kepada CV Pabrik Teknologi dalam menyelesaikan masalah yaitu dengan menggunakan metode *Use Case Point* dan *Function Point*. Kemudian akan dilakukan perbandingan antara metode *Use Case Point*, *Function Point* dan *Guesstimate*. Penelitian ini juga menjabarkan penjadwalan pengerjaan proyek Sistem Antrian RSUD Dr. Soetrasno Rembang menggunakan *Gantt Chart* berdasarkan WBS. Oleh karena itu penelitian ini mengangkat tema dengan judul “Perbandingan Evaluasi Biaya Sistem Antrian RSUD Dr. Soetrasno Rembang Menggunakan Metode *Use Case Point* dan *Function Point* (Studi Kasus: CV Pabrik Teknologi)”.



1.2 Rumusan masalah

Berdasarkan latar belakang, maka pokok masalah yang akan diteliti adalah:

1. Bagaimana pembagian lingkup kerja Sistem Antrian RSUD Dr. Soetrasno Rembang dengan pendekatan *Work Breakdown Structure (WBS)*?
2. Bagaimana hasil penjadwalan pengerjaan Sistem Antrian RSUD Dr. Soetrasno Rembang menggunakan *Gantt Chart* berdasarkan *Work Breakdown Structure (WBS)*?
3. Bagaimana hasil penerapan metode *Use Case Point* dalam mengestimasi biaya untuk pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang?
4. Bagaimana hasil penerapan metode *Function Point* dalam mengestimasi biaya untuk pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang?
5. Bagaimana hasil perbandingan perbandingan evaluasi biaya Sistem Antrian RSUD Dr. Soetrasno Rembang menggunakan metode *Use Case Point*, *Function Point* dan *Guesstimate*?

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah:

1. Menjelaskan pembagian lingkup kerja Sistem Antrian RSUD Dr. Soetrasno Rembang dengan pendekatan *Work Breakdown Structure (WBS)*.
2. Menjelaskan hasil penjadwalan pengerjaan Sistem Antrian RSUD Dr. Soetrasno Rembang menggunakan *Gantt Chart* berdasarkan *Work Breakdown Structure (WBS)*.
3. Menjelaskan hasil penerapan metode *Use Case Point* dalam mengestimasi biaya untuk pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang.
4. Menjelaskan hasil penerapan metode *Function Point* dalam mengestimasi biaya untuk pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang.
5. Menjelaskan hasil perbandingan perbandingan evaluasi biaya Sistem Antrian RSUD Dr. Soetrasno Rembang menggunakan metode *Use Case Point*, *Function Point* dan *Guesstimate*.

1.4 Manfaat

Manfaat yang diperoleh dari penelitian ini adalah:

1. Menambah bahasan keilmuan mengenai pembagian lingkup kerja Sistem Antrian RSUD Dr. Soetrasno Rembang dengan pendekatan *Work Breakdown Structure (WBS)*.
2. Menambah bahasan keilmuan mengenai penjadwalan pengerjaan Sistem Antrian RSUD Dr. Soetrasno Rembang menggunakan *Gantt Chart* berdasarkan *Work Breakdown Structure (WBS)*.
3. Menambah bahasan keilmuan mengenai estimasi biaya menggunakan metode *Use Case Point*.
4. Menambah bahasan keilmuan mengenai estimasi biaya menggunakan metode *Function Point*.
5. Menambah informasi yang berkaitan dengan pertimbangan pengambilan keputusan pihak manajemen terkait estimasi biaya untuk mencapai tujuan pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang.



1.5 Batasan masalah

Agar masalah yang diteliti tidak menyimpang dan dapat dipahami maka perlu adanya batasan masalah. Adapun batasan masalah dalam penelitian ini adalah:

1. Ruang lingkup penelitian ini hanya dibatasi pada penjadwalan pengerjaan proyek menggunakan *Gantt Chart* pada RSUD Dr. Soetrasno Rembang.
2. Ruang lingkup penelitian ini hanya dibatasi sampai tiga tahap *Project Life Cycle* (PLC) yaitu pada tahap *Define project goal, Plan Project* dan *Execute project plan*.
3. Pembuatan *User Story, Use Case Diagram* dan *Data Flow Diagram* berdasarkan hasil wawancara dan validasi dengan pihak CV Pabrik Teknologi.

1.6 Sistematika pembahasan

Sistematika pembahasan skripsi ini terdiri dari enam bab yaitu Bab I Pendahuluan, Bab II Landasan Kepustakaan, Bab III Metodologi Penelitian, Bab IV Pengumpulan Data, Bab V Analisis dan Perhitungan dan Bab VI Penutup.

Bab I: Pendahuluan

Membahas tentang latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian serta sistematika penulisan.

Bab II: Landasan Kepustakaan

Menjelaskan dasar teori dan referensi yang akan digunakan untuk mendukung penelitian ini.

Bab III: Metodologi Penelitian

Mendeskripsikan tentang metode, jenis dan sumber data, serta teknik analisis data yang digunakan.

Bab IV: Hasil

Berisi tentang data-data yang akan dianalisis menggunakan metode *Use Case Point* dan metode *Function Point*.

Bab V: Pembahasan

Menjelaskan proses pembuatan penjadwalan pengerjaan *software* menggunakan *Work Breakdown Structure* (WBS) serta menjelaskan proses pengolahan dan analisis data menggunakan metode *Use Case Point* dan metode *Function Point* sehingga menghasilkan analisis estimasi biaya total pengembangan Sistem Antrian RSUD Dr. Soetrasno dan kemudian melakukan analisis hasil dengan membuat perbandingan terhadap metode *Use Case Point*, metode *Function Point* dan *Guesstimate*.

Bab VI: Penutup

Berisi kesimpulan dari hasil rumusan masalah serta saran yang dapat dijadikan sebagai bahan pertimbangan untuk Sistem Antrian RSUD Dr. Soetrasno dan proyek-proyek selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Tujuan diambilnya beberapa referensi dari penelitian terdahulu seperti penelitian yang telah diambil dan dilakukan pada organisasi atau perusahaan dalam melakukan penjabaran pengerjaan proyek dan menentukan estimasi biaya adalah untuk mendukung penelitian ini yang disajikan pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka

Judul Penelitian	Objek	Metode	Hasil
1 <i>Use Case Point Estimation Technique in Software Development</i>	<i>ForEx Spot</i> (aplikasi untuk melakukan perdagangan)	<i>Use Case Point</i>	Dalam penelitian ini, penggunaan metode <i>Use Case Point</i> diusulkan untuk mencapai akurasi yang lebih di tahap awal dengan faktor-faktor kompleksitas seperti <i>Technical</i> dan <i>Environment Complexity Factor</i> . Dengan metode ini, biaya dan waktu pengembangan proyek dapat dihitung secara efektif. Hasil dari perhitungan estimasi biaya dengan metode <i>Use Case Point</i> pada <i>ForEx Spot</i> adalah total estimasi biaya sebesar Rp 5.854.000.
2 Implementasi Metode <i>Function Point</i> Untuk Mengukur <i>Volume Software</i>	<i>AttendMaster</i> (Sistem Presensi Karyawan yang dapat melayani bisnis kecil sampai menengah dengan karyawan berjumlah 10.100 orang)	<i>Function Point</i>	Penelitian ini menjelaskan mengenai kelebihan dari metode <i>Function Point</i> adalah memiliki kemampuan untuk menyediakan perkiraan <i>size</i> proyek dalam bentuk sumber daya yang dibutuhkan serta menjelaskan mengenai metode <i>Function Point</i> dapat dijadikan sebagai alternatif untuk menghitung <i>size</i> perangkat lunak berdasarkan kompleksitas pada <i>AttendMaster</i> dengan hasil CFP = 81, RCAF = 41, FP = 85.86.
3 <i>A Work Breakdown Structure for Implementing and Costing an ERP Project</i>	<i>Enterprise Resource Planning (ERP)</i>	<i>Work Breakdown Structure</i> menggunakan <i>Gantt Chart</i>	Penelitian ini menganjurkan dalam implementasi menggunakan SAP, disarankan untuk melanjutkan SAP dan metodologi lainnya ke dalam WBS agar semua aspek dalam implementasi ERP dapat diatasi dengan baik. WBS digunakan untuk mendefinisikan aktivitas proyek ERP dari tingkat tertinggi hingga terendah diperhitungkan secara detail.



Dalam penelitian *Use Case Point Estimation Technique in Software Development* menekankan pada pendekatan algoritmik baru untuk mempertimbangkan faktor teknis dan lingkungan sehingga estimasi biaya lebih akurat. Penggunaan metode *Use Case Point* diusulkan untuk mencapai akurasi yang lebih di tahap awal dengan faktor-faktor kompleksitas seperti *Technical* dan *Environment Complexity Factor*. Dengan metode ini, biaya dan waktu pengembangan proyek dapat dihitung secara efektif. *Use Case Point* merupakan teknik terbaik dalam mengestimasi biaya dengan keuntungan seperti: 1) Setiap *Use Case* memiliki tujuan atau ringkasan mengenai sistem yang akan ditawarkan; 2) Setiap *Use Case* menjelaskan dengan baik bahwa sistem tersebut dapat melakukan fitur-fitur tersebut atau tidak; 3) Setiap *Use Case* akan menjadi fungsionalitas sistem yang akan menyediakan *framework* untuk menganalisis waktu pengembangan dan anggaran (Johri et al., 2016).

Penelitian yang berjudul *Implementasi Metode Function Point untuk Mengukur Volume Software* menjelaskan pada saat sebelum melakukan estimasi sumber daya dan biaya pada suatu proyek, maka seorang pengembang perlu mengetahui *size* perangkat lunak yang akan dikembangkan terlebih dahulu guna mendukung estimasi sumber daya dan biaya yang akan dilakukan. Selain itu, dahulu para pengembang menggunakan metode *Lines of Code* (LOC) atau KLOC (*100 Lines of Code*). Namun metode LOC dan KLOC memiliki beberapa kekurangan seperti penggunaan bahasa pemrograman ketika pengembangan, gaya pengkodean *programmer* serta LOC baru bisa ditentukan jika proyek telah selesai pada tahap pengkodean. Hal ini menyebabkan metode *Function Point* dirasa lebih unggul karena untuk mengukur *size* perangkat lunak baiknya dilakukan lebih awal sebelum proyek melangkah lebih jauh dan metode *Function Point* dapat mengetahui *size* sebuah perangkat lunak lebih awal. Perhitungan matematika pada metode *Function Point* yaitu menghitung *Crued Function Point* (CFP) untuk mengukur proses informasi yang dapat dibagi menjadi tiga kategori yaitu *Simple, Average, Complex*. Kemudian menghitung faktor kompleksitas relatif untuk menyimpulkan kompleksitas dari suatu sistem perangkat lunak. Menghitung *Function Point* dengan perhitungan dari nilai CFP dan nilai RCAF (Kusrini et al, 2015).

Penelitian dengan judul *A Work Breakdown Structure for Implementing and Costing an ERP Project* menyajikan *Work Breakdown Structure* (WBS) untuk implementasi secara bertahap dari *Enterprise Resource Planning* (ERP). Dalam pengerjaan ERP, terdapat beberapa masalah terkait perencanaan proyek serta penentuan biaya proyek. Oleh karena itu dokumen WBS sangat berperan penting dalam menyajikan rincian pekerjaan yang harus diselesaikan, rincian alokasi sumber daya, pengukuran dan pengendalian proyek sehingga menghasilkan perencanaan yang baik (Momoh et al., 2008).

2.2 Profil CV Pabrik Teknologi

CV Pabrik Teknologi merupakan sebuah perusahaan berbasis jasa yang berfokus pada bidang jasa teknologi informasi. CV Pabrik Teknologi merupakan salah satu perusahaan rintisan atau *startup* yang ada di kota Malang. CV Pabrik Teknologi didirikan pada tanggal 25 Januari 2013 dan mulai beroperasi secara resmi pada tanggal 12 April 2014 karena telah memiliki badan hukum tetap.

Kantor CV Pabrik Teknologi berlokasi di Jl. Joyo Tambaksari 32 kota Malang, Jawa Timur. Berdasarkan tujuan awal perusahaan ini dibentuk, ruang lingkup kegiatan dari CV Pabrik Teknologi meliputi berbagai layanan. Jenis layanan yang dihasilkan oleh CV Pabrik Teknologi, antara lain: pengembangan *website* tiket pesawat *online* (*widget*, pemesanan, keagenan),



pengembangan *website* profil baik pribadi, bisnis, institusi ataupun korporasi, pengembangan *website* aplikasi untuk institusi, korporasi maupun bisnis, pengembangan *software* administrasi, pengembangan aplikasi *mobile* untuk berbagai keperluan dan instalasi jaringan pada perkantoran. Saat ini, kegiatan utama CV Pabrik Teknologi adalah melayani pembuatan *website* dan aplikasi berbasis *mobile*.

2.2.1 Visi CV Pabrik Teknologi

Menjadi perusahaan IT yang terdepan dalam menghasilkan produk dan berperan serta dalam memajukan kehidupan bangsa.

2.2.2 Misi CV Pabrik Teknologi

1. Menghasilkan produk IT yang berkualitas dan bermanfaat untuk khalayak umum.
2. Mengembangkan kerjasama dengan berbagai pihak.
3. Berintegrasi dengan berbagai bidang.
4. Membantu meningkatkan daya saing bagi klien dan *stakeholder*.

2.3 Manajemen Proyek

Manajemen merupakan proses perencanaan, pengorganisasian, memimpin dan mengendalikan kegiatan anggota dan sumber daya lainnya agar tercapainya sasaran organisasi yang telah ditentukan (Soeharto, 2001 disitasi dari Noerlina, 2008).

Secara garis besar, proyek memiliki empat tahapan berikut: 1) Tahap Konsepsi, pada tahap ini dilakukan penyusunan, perumusan gagasan, analisis pendahuluan dan melakukan studi kelayakan; 2) Tahap Pendefinisian, menyiapkan rencana proyek secara detail dan menentukan spesifikasi proyek secara rinci; 3) Tahap Akuisisi, melakukan desain, pengadaan fasilitas pendukung dan material untuk tahap selanjutnya, produksi serta implementasi dan 4) Tahap Operasi, merupakan tahap akhir suatu proyek dimana proyek diserahkan kepada pengguna (Santosa, 2003 disitasi dari Noerlina, 2008).

Manajemen proyek merupakan kegiatan yang melibatkan perencanaan, pemantauan, pengendalian anggota proyek dan peristiwa yang terjadi dalam pengembangan perangkat lunak dari konsep operasional awal hingga akhir (Pressman, 2010).

Manajemen proyek merupakan kegiatan mengatasi kendala, mengembangkan software dengan kualitas tinggi dan memastikan proyek yang dijalankan telah sesuai dengan yang direncanakan. Selain itu kriteria keberhasilan dalam manajemen proyek memang bervariasi tergantung dari proyek yang dijalankan, namun untuk sebagian besar proyek, tujuan terpenting adalah: 1) Menyerahkan hasil proyek kepada pelanggan tepat pada waktunya; 2) *Software* yang dikembangkan memenuhi kebutuhan pelanggan; 3) Menjaga agar tim pengembang nyaman dan bekerja dengan baik dan 4) Hindari biaya berlebih dalam anggaran (Sommerville, 2009).

Manajemen proyek terdapat 10 bidang pengetahuan yaitu Manajemen ruang lingkup proyek (*Scope*), Manajemen waktu proyek (*Time*), Manajemen biaya proyek (*Cost*), Manajemen mutu proyek (*Quality*), Manajemen sumber daya manusia (*Human resource*), Manajemen komunikasi (*Communications*), Manajemen resiko (*Risk*), Manajemen pengadaan (*Procurement*), Manajemen pemangku kepentingan proyek (*Stakeholder*) dan Manajemen integrasi proyek (*Integration*).



1. Manajemen ruang lingkup proyek (*Scope*), melakukan pendefinisian dan pengelolaan seluruh pekerjaan yang harus diselesaikan agar proyek berjalan dengan sukses.
2. Manajemen waktu proyek (*Time*), melakukan perkiraan, penjadwalan dan memastikan waktu yang diperlukan untuk menyelesaikan proyek.
3. Manajemen biaya proyek (*Cost*), melakukan persiapan dan pengelolaan anggaran untuk proyek.
4. Manajemen mutu proyek (*Quality*), memastikan proyek dilakukan sesuai dengan kebutuhan dan memastikan kebutuhan yang telah didefinisikan sudah terpenuhi.
5. Manajemen sumber daya manusia (*Human resource*), mengelola seluruh sumber daya manusia (SDM) yang terlibat dalam proyek secara efektif.
6. Manajemen komunikasi (*Communications*), mengumpulkan, menyebarkan dan menyimpan informasi proyek.
7. Manajemen resiko (*Risk*), identifikasi, analisis serta menanggapi resiko yang terkait dengan proyek.
8. Manajemen pengadaan (*Procurement*), yaitu kegiatan pengadaan barang dan jasa untuk proyek.
9. Manajemen pemangku kepentingan proyek (*Stakeholder*), identifikasi dan analisis mengenai keterlibatan stakeholder dalam proyek.
10. Manajemen integrasi proyek (*Integration*), memastikan seluruh fungsi apakah mempengaruhi atau dipengaruhi dengan bidang-bidang lainnya (Schwalbe, 2014).

2.4 Project Cost Management

Dalam manajemen biaya proyek, memperkirakan biaya atau tugas tertentu dengan melibatkan perkiraan durasi melibatkan lima langkah yaitu: 1) Mendefinisikan sumber daya apa yang dibutuhkan untuk melakukan pekerjaan; 2) Menentukan jumlah sumber daya yang dibutuhkan; 3) Mendefinisikan biaya penggunaan masing-masing sumber; 4) Menghitung biaya tugas atau aktivitas dan 5) Memastikan bahwa sumber daya setara (Marchewka, 2003).

2.5 Software Cost Estimation

Estimasi biaya merupakan proses memprediksi usaha yang dibutuhkan untuk mengembangkan perangkat lunak. Estimasi biaya perangkat lunak memperkirakan estimasi usaha (biasanya secara *person-month*), durasi proyek (dalam waktu kalender) dan biaya (bisa dalam satuan *dollar*, rupiah atau mata uang lain). Sebagian besar metode estimasi biaya menghasilkan perkiraan usaha yang kemudian dapat dikonversi menjadi durasi proyek dan biaya. Konversi menjadi biaya bisa dilakukan dengan menghitung rata-rata gaji per staff yang terlibat dalam proyek lalu mengalikannya dengan perkiraan usaha yang dibutuhkan.

Proses estimasi biaya melibatkan tujuh langkah yaitu: 1) Mendefinisikan tujuan estimasi biaya; 2) Menentukan kebutuhan data dan sumber daya; 3) Mendefinisikan kebutuhan perangkat lunak; 4) Mendetailkan sebanyak mungkin kriteria perangkat lunak yang layak; 5) Gunakan beberapa teknik estimasi biaya untuk memanfaatkan kombinasi dari keduanya; 6)

Bandingkan perkiraan yang berbeda; 7) Setelah proyek dimulai, pantau biaya aktual serta kemajuannya (Leung et al., 2002).

2.6 Project Life Cycle (PLC)

Project Life Cycle (PLC) merupakan kumpulan dari fase-fase yang memetakan *life cycle* suatu proyek dari awal sampai akhir untuk mendefinisikan, membangun dan memberikan sistem informasi dari proyek. Pada umumnya, *Project Life Cycle* pada IT adalah *Software Development Life Cycle (SDLC)*. Tahapan dalam *Project Life Cycle (PLC)* ada lima tahap yaitu *Define Project Goal*, *Plan Project*, *Execute Project Plan*, *Close Project* dan *Evaluate Project* (Marchewka, 2003).

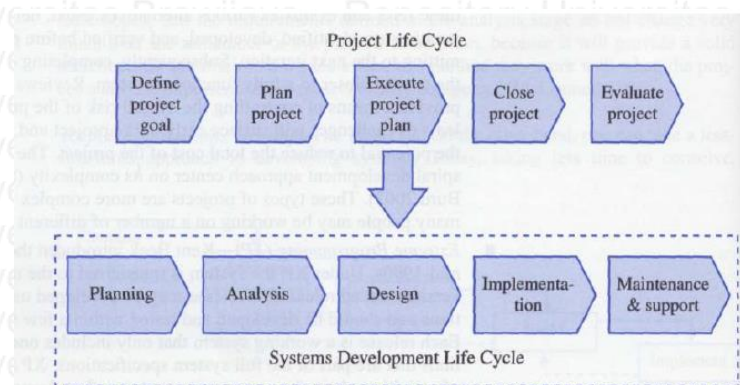
Define Project Goal, mendeskripsikan seluruh tujuan proyek yang dijadikan langkah pertama dalam sebuah proyek. Tujuan ini harus fokus pada nilai bisnis untuk organisasi. Tujuan yang terdefinisi dengan baik memberikan fokus yang jelas untuk tim proyek.

Plan Project, setelah tujuan proyek didefinisikan, maka proses pengembangan rencana proyek akan jauh lebih mudah. *Deliverables*, tugas, sumber daya dan waktu untuk menyelesaikan setiap tugas harus ditetapkan untuk setiap tahap pengerjaan proyek. Rencana awal harus mendefinisikan lingkup, jadwal dan anggaran yang sudah disepakati.

Execute Project Plan, setelah tujuan dan rencana proyek telah didefinisikan, saatnya untuk mengimplementasikan rencana proyek. Jadwal, lingkup, perkembangan, anggaran dan SDM harus dikelola untuk memastikan bahwa proyek telah mencapai tujuannya. Perkembangan proyek harus didokumentasikan dan dibandingkan dengan rencana awal proyek. Pada akhir fase ini tim proyek mengimplementasikan atau memberikan produk yang telah diselesaikan untuk organisasi.

Close Project, pada tahap ini memastikan bahwa semua tugas telah diselesaikan sesuai dengan rencana dan sesuai kesepakatan antara *Project Manager* dengan klien. Oleh karena itu perlu adanya pengakuan secara formal oleh klien bahwa mereka akan menerima produk dan membayar produk yang telah diterima. Pada tahap ini sering ditutup dengan laporan akhir dan presentasi ke klien dan mendokumentasikan semua *deliverable* telah selesai dan sesuai dengan spesifikasi.

Evaluate Project, pada tahap ini tim proyek bisa melakukan evaluasi, bisa dengan cara mendokumentasikan pengalaman sebelumnya yang dapat dijadikan pembelajaran untuk proyek-proyek selanjutnya (Marchewka, 2003).



Gambar 2.1 Project Life Cycle dan Software Development Life Cycle

Sumber: Marchewka (2003)



2.7 Software Development Life Cycle (SDLC)

SDLC atau *Software Development Life Cycle* merupakan urutan kegiatan pengembangan sistem dari satu kegiatan ke kegiatan berikutnya. SDLC merupakan bagian dari PLC karena banyak dari kegiatan pengembangan sistem informasi terjadi di tahap *Execute*. *Software Development Life Cycle* memiliki lima fase yaitu *Planning, Analysis, Design, Implementation* dan *Maintenance*.

Planning, mengidentifikasi dan menanggapi masalah atau kesempatan dan menggabungkan proses *project management* dengan kegiatan *system development*. Perencanaan memastikan bahwa tujuan, lingkup, anggaran, jadwal, teknologi, *system development*, metode dan *tools* yang dibutuhkan berada di tempat.

Analysis, mencoba menyelidiki masalah atau kesempatan secara lebih detail. Umumnya, sistem analisis akan bertemu dengan berbagai *stakeholders* (pengguna, *manager*, pelanggan) untuk mempelajari lebih lanjut tentang masalah atau kesempatan.

Design, tim proyek menggunakan *requirements* sebagai masukan untuk merancang arsitektur yang mendukung sistem. Arsitektur ini meliputi merancang jaringan, konfigurasi perangkat keras, *database*, antarmuka pengguna dan program perangkat lunak.

Implementation, mencakup pengembangan dan pembangunan sistem, pengujian serta instalasi. Selain itu juga memastikan pelatihan dan dokumentasi harus ada di tempat.

Maintenance and Support, untuk tahap ini mungkin tidak termasuk tahapan proyek saat ini, sehingga masih menjadi pertimbangan penting. Setelah pengerjaan perubahan sistem selesai dilakukan, bentuk pemeliharaan dan peningkatan terhadap sistem sering diminta untuk memperbaiki kesalahan yang ditemukan dalam sistem (Marchewka, 2003).

2.8 Work Breakdown Structure (WBS)

Work Breakdown Structure (WBS) adalah penyampaian kumpulan tahapan pengerjaan proyek yang ditampilkan dalam tampilan grafis untuk mengatur dan pembagian lingkup kerja dalam proyek. Semua pekerjaan dalam WBS akan diidentifikasi, estimasi, dijadwalkan dan dianggarkan. Oleh karena itu, WBS berisi dasar ruang lingkup proyek yang dibutuhkan untuk mencapai tujuan teknis pengerjaan. Pada umumnya, WBS merupakan kerangka *multi-level* yang menyelenggarakan dan menampilkan elemen grafis yang mewakili pekerjaan yang harus dicapai.

Elemen-elemen yang terkandung di dalam WBS yang harus diketahui yaitu: 1) Didefinisikan, dapat dijelaskan dan mudah dipahami oleh seluruh anggota proyek; 2) Dikelola, berarti suatu pekerjaan dimana wewenang dan tanggung jawab tertentu dapat ditugaskan kepada seorang pekerja; 3) Perkiraan, dapat memperkirakan durasi waktu, biaya yang diperkirakan dari sumber daya manusia yang dibutuhkan dalam menyelesaikan sebuah proyek; 4) Bebas, tidak bergantung pada elemen lain yang sedang berlangsung; 5) Terintegrasi, terintegrasi dengan elemen-elemen dalam pekerjaan proyek dengan perkiraan biaya dan jadwal; 6) Terukur, dapat digunakan untuk mengukur kemajuan proyek; 7) Adaptasi, bersifat fleksibel sehingga jika terjadi penambahan atau penghapusan lingkup pengerjaan, dapat ditampung dalam WBS (Rev, 2003).



2.9 Gantt Chart

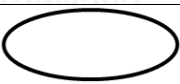

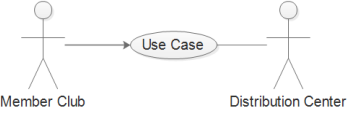
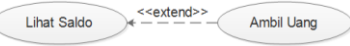
Gantt Chart merupakan format untuk menampilkan informasi mengenai penjadwalan proyek dengan daftar *task* atau kegiatan proyek yang sesuai dengan tanggal mulai dan tanggal selesai dalam bentuk kalender (Schwalbe, 2014).

2.10 Use Case

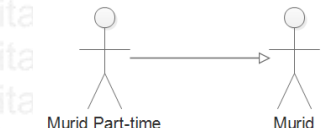

User story merupakan deskripsi pendek yang ditulis oleh pengguna mengenai sistem yang mereka butuhkan. *User story* terdiri dari alur teknis pada sistem. Alur teknis tersebut akan digunakan oleh pengembang untuk menerjemahkan atau menganalisis kebutuhan pengguna dan spesifikasi sistem. Kebutuhan yang didefinisikan dalam *user story* bisa berisi satu atau lebih alur teknis (Schwalbe, 2014).

Use Case merupakan representasi dari nilai yang ditampilkan sistem pada aktor artinya *use case* bukan fungsi maupun fitur. *Use case* memiliki nama, deskripsi singkat dan deskripsi rinci. Deskripsi tersebut menggambarkan apa yang dapat dilakukan sistem untuk memenuhi kebutuhan dan bagaimana aktor menggunakan sistem. Namun, *use case* hanya menjelaskan apa saja yang dilakukan aktor dan sistem tapi tidak menjelaskan bagaimana aktor dan sistem melakukan kegiatan tersebut. *Use case scenario* terdiri dari nama *use case*, *Brief Description* (deskripsi singkat), *Basic Flow* (aliran normal), *Alternate Flow* (aliran alternatif), *Sub-flow*, *pre-condition* dan *post condition* (Bittner, 2002). Skenario merupakan perwujudan dari *use case* yang merepresentasikan *main flow* dan *alternative flow* dari sebuah *use case*. Skenario dipicu oleh pilihan, kondisi kesalahan, pelanggaran keamanan, dan lain-lain dari *use case* yang sama (Bredemeyer, 2001). Komponen-komponen *Use Case Diagram* disajikan pada Tabel 2.2.

Tabel 2.2 Komponen *Use Case Diagram*

	Nama	Gambar	Fungsi
1	<i>Use Case</i>		Mendeskripsikan fungsi dari sistem yang dilihat dari sudut pandang pengguna
2	<i>Actor</i>		Sesuatu yang berinteraksi dengan sistem, bisa berupa orang, sistem atau organisasi
3	<i>Association</i>		Mendeskripsikan relasi aktor dengan <i>use case</i> . Panah tutup menggambarkan aktor 'member club' melakukan <i>use case</i> tersebut. Asosiasi tanpa panah menggambarkan adanya interaksi dari <i>use case</i> ke aktor yang menerima hasil dari <i>use case</i> tersebut
4	<i>Extend</i>		Mendeskripsikan situasi dimana sebuah <i>use case</i> termasuk dalam <i>use case</i> lainnya

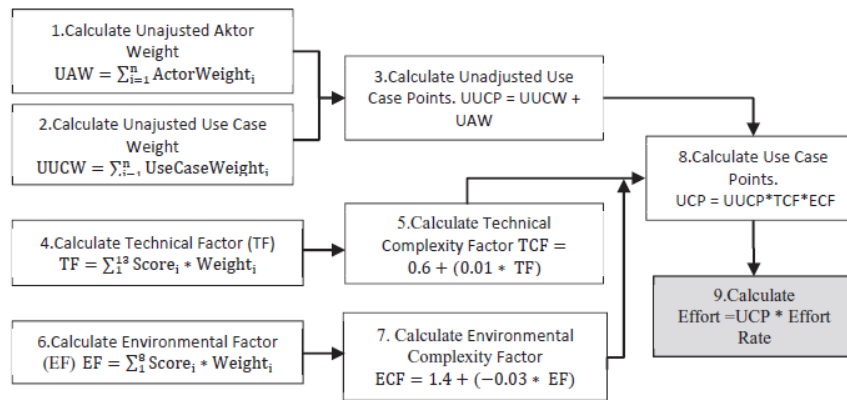


5	<i>Generalization</i>		Menggambarkan situasi dimana satu lebih umum daripada yang lain
6	<i>Include</i>		Menggambarkan situasi dimana sebuah use case termasuk dalam use case lainnya

Sumber: Kendall et al (2011)

2.11 Metode Use Case Point

Metode *Use Case Point* pertama kali dipopulerkan oleh Gustav Karner pada tahun 1993 yang merupakan turunan dari metode *Function Point Analysis* (FPA) yang bertujuan untuk menyediakan metode estimasi sederhana dengan berorientasi pada objek proyek perangkat lunak (Karner, 1993 disitasi dari Sholiq, 2013). Dalam metode *Use Case Point*, langkah-langkah yang perlu dilakukan disajikan pada Gambar 2.6.



Gambar 2.2 Langkah-langkah Metode Use Case Point

Sumber: Sholiq (2015)

2.11.1 Unadjusted Actor Weights (UAW)

Langkah pertama adalah mengidentifikasi dan mengklasifikasikan jenis aktor. Jenis aktor diklasifikasikan menjadi tiga kategori yaitu *Simple*, *Average* dan *Complex*. *Simple actor* merupakan merupakan aktor yang berinteraksi melalui API seperti *Command Prompt*. *Average actor* adalah aktor yang berinteraksi melalui protokol seperti TCP/IP, FTP, HTTP atau disebut aktor yang melakukan penyimpanan data (*file*, RDBMS). Sedangkan *Complex Actor* merupakan aktor yang berinteraksi melalui GUI atau halaman web (Ribu, 2001).

Tabel 2.3 Tipe dan Bobot Aktor pada Use Case Point

Tipe Aktor	Bobot
<i>Simple</i>	1
<i>Average</i>	2
<i>Complex</i>	3

Sumber: Ribu (2001)

Unadjusted Actor Weights (UAW) diperoleh dari berapa banyak aktor dari masing-masing tipe aktor kemudian dikali dengan total bobot faktor masing-masing sesuai dengan Tabel 2.3.



2.11.2 Unadjusted Use Case Weights (UUCW)

Tabel 2.4 Tipe dan Bobot *Use Case* pada *Use Case Point*

Tipe <i>Use Case</i>	Deskripsi	Bobot
<i>Simple</i>	≤ 3 transaksi	5
<i>Average</i>	4 sampai 7 transaksi	10
<i>Complex</i>	Lebih dari 7 transaksi	15

Sumber: Ribu (2001)

Setiap *Use Case* diklasifikasikan ke dalam salah satu dari tiga jenis yaitu *Simple*, *Average* dan *Complex* seperti yang ditunjukkan pada Tabel 2.4. *Simple Use Case* mengandung maksimum 3 transaksi. *Average Use Case* berisi antara 4 sampai 7 transaksi dan *Complex Use Case* mengandung lebih dari 7 transaksi. *Unadjusted Use Case Weights* (UUCW) dihitung dengan mengalikan jumlah bobot masing-masing *Use Case* lalu dijumlahkan.

2.11.3 Unadjusted Use Case Point (UUCP)

Unadjusted Use Case Point (UUCP) diperoleh dari penjumlahan *Unadjusted Actor Weights* (UAW) dengan *Unadjusted Use Case Weights* (UUCW) seperti pada Persamaan 2.1.

$$UUCP = UAW + UUCW \quad (2.1)$$

2.11.4 Technical Complexity Factors (TCF)

Tabel 2.5 *Technical Factors*

	<i>Technical Factor</i>	Bobot
T1	<i>Distributed System Required</i>	2
T2	<i>Response Time</i>	2
T3	<i>End User Efficiency</i>	1
T4	<i>Complex Internal Processing Required</i>	1
T5	<i>Reusable Code</i>	1
T6	<i>Easy to Install</i>	0,5
T7	<i>Easy to Use</i>	0,5
T8	<i>Portable</i>	2
T9	<i>Easy to Change</i>	1
T10	<i>Concurrent</i>	1
T11	<i>Security Features</i>	1
T12	<i>Access for Third Parties</i>	1
T13	<i>Special Training Required</i>	1

Sumber: Ribu (2001)

Technical Complexity Factors (TCF) berisi 13 faktor yang memiliki pengaruh besar pada kinerja proyek. Setiap nilai faktor merupakan pengaruh terhadap produktivitas perangkat lunak yang memiliki peringkat antara 0 sampai 5 (0 berarti tidak ada pengaruh dan 5 berarti pengaruh yang kuat). TCF dihitung dengan mengalikan nilai setiap bobot faktor (T1-T13) kemudian menambahkan semua angka-angka untuk mendapatkan jumlah yang disebut TF lalu menerapkan Persamaan 2.2.

$$TCF = 0,6 + (0,01 * TF) \quad (2.2)$$



2.11.5 Environment Complexity Factor (ECF)

Tabel 2.6 Environment Factors

Environment Factor		Bobot
F1	<i>Familiarity With The Project</i>	1,5
F2	<i>Application Experience</i>	0,5
F3	<i>OO Programming Experience</i>	1
F4	<i>Lead Analyst Capability</i>	0,5
F5	<i>Motivation</i>	1
F6	<i>Stable Requirements</i>	2
F7	<i>Part Time Staff</i>	-1
F8	<i>Difficult Programming Language</i>	-1

Sumber: Ribu (2001)

Environment Factor (EF) dihitung dengan mengalikan nilai faktor (F1-F8) dengan bobot nilai masing-masing kemudian dijumlahkan untuk mendapatkan jumlah yang disebut *EFactor* seperti yang ditunjukkan pada Persamaan 2.3.

$$ECF = 1.4 + (-0,03 * EF) \quad (2.3)$$

2.11.6 Estimasi Effort

Pada langkah terakhir didapatkan hasil *Use Case Point* dengan mengalikan *Unadjusted Use Case Point* dengan *Technical Complexity Factor*.

$$UCP = UUCP * TCF * ECF \quad (2.4)$$

Untuk mengubah nilai UCP menjadi nilai *effort* yaitu *Hours of Effort* maka diperlukan pengembangan perangkat lunak sesuai dengan metrik *Use Case Point*. Karner (1993) menyatakan bahwa nilai 20 *staff hours* yang dibutuhkan untuk setiap *use case*. Dengan asumsi *p* adalah *staff hours per use case* (Saleh, 2011).

$$Hours\ of\ Effort = UCP * p \quad (2.5)$$

2.11.7 Estimasi Biaya

Effort yang telah didapatkan pada tahap sebelumnya dibagi menjadi tiga aktivitas yang dijalankan dalam pengembangan perangkat lunak yaitu *Software Development*, *Ongoing Activity* dan *Quality dan Testing*.

1. *Software Development*, meliputi analisis kebutuhan pengguna seperti permintaan dan spesifikasi, desain, implementasi, pengujian integrasi yang mencapai sekitar 42% dari total *effort*.
2. *Ongoing Activity*, atau disebut aktivitas yang berkesinambungan seperti manajemen proyek, manajemen konfigurasi, dokumentasi, penerimaan dan penyebaran yang mencapai sekitar 21% dari total *effort*.
 - 1) *Project Management*, PM melakukan kegiatan yang berhubungan dengan manajemen proyek meliputi memantau kemajuan proyek secara ketat, melaporkan manajemen resiko, memperbarui jadwal proyek secara berkala.
 - 2) *Configuration Management*, melakukan pemeliharaan, pengelolaan, pengendalian dan pengembangan perangkat lunak. Idealnya kegiatan



tersebut dilakukan dengan menggunakan alat konfigurasi otomatis yang memperbarui status secara berkala. Hasil alat tersebut kemudian dianalisis oleh tim manajemen tertentu.

- 3) *Documentation*, membuat buku manual instalasi, operasi atau *template* standar. Biasanya ditulis oleh pengembang perangkat lunak.
 - 4) *Training & Support*, pada fase ini tim pengembang membutuhkan beberapa dukungan teknis dan pelatihan dalam memecahkan masalah teknis yang mungkin terjadi guna meningkatkan efisiensi dan produktivitas.
3. *Quality dan Testing*, aktivitas yang berhubungan dengan kualitas dan pengujian seperti pengujian integrasi, penjaminan kualitas, evaluasi dan pengujian yang mencapai sekitar 37% dari total *effort*.
- 1) *Integration Testing*, melakukan uji integrasi antara modul, sistem dan komponen eksternal. Kemudian melakukan hasil analisis dan menangani masalah yang terjadi. *Milestone* dari fase ini adalah perangkat lunak terintegrasi.
 - 2) *Quality Assurance*, memastikan seluruh aktivitas yang terjadi pada setiap fase telah sesuai dengan standar internal dan menghasilkan *deliverables* yang benar.
 - 3) *Evaluation & Testing*, melakukan kegiatan evaluasi dan pengujian, termasuk kegiatan validasi dan verifikasi. Proses ini dilakukan oleh tim pengembang (Saleh, 2011).

2.12 Metode Function Point

Konsep dari *Function Point* pertama kali diperkenalkan oleh Allan J. Albercht di tahun 1979 yang kemudian dilakukan perbaikan di tahun 1984. Pada tahun 1986 sejak berdirinya *Internasional Function Point Counting Practices (IFPUG)*, beberapa versi dari *Function Point Counting Practices Manual* diterbitkan.

Metode *Function Point* merupakan metode yang berorientasi pada ukuran fungsi perangkat lunak untuk menganalisis estimasi biaya serta usaha yang dibutuhkan dalam pengembangan perangkat lunak. Untuk menentukan nilai *Function Point*, tahap-tahap yang harus dilakukan sebagai berikut (Marthaler, 2005).

2.12.1 Menghitung Nilai Function Point

Langkah pertama, identifikasi fungsi-fungsi sebagai parameter perhitungan terhadap proyek dengan pendekatan *Data Flow Diagram*.

Langkah kedua, mengklasifikasikan tipe fungsi pengguna berdasarkan karakteristik kompleksitas yang dimiliki dengan pendekatan *Data Flow Diagram*. Elemen tipe fungsi pengguna ada lima tipe yaitu: 1) *External Input (EI)* merupakan proses dasar yang memproses data atau mengendalikan informasi yang berasal dari luar batas aplikasi. Tujuan utama EI adalah mempertahankan satu atau lebih ILF dan/atau mengubah perilaku sistem; 2) *External Outputs (EO)* merupakan sebuah proses yang melakukan pengiriman data atau informasi yang ditujukan kepada *user*. Tujuan utama EO ialah menyajikan informasi kepada pengguna melalui pemrosesan secara logika. Logika pemrosesan harus mengandung



setidaknya satu rumus atau perhitungan matematis, membuat data turunan, mempertahankan satu atau lebih ILF atau mengubah perilaku sistem; 3) *External Inquiry* (EQ) merupakan sebuah proses yang melakukan pengiriman data atau informasi diluar batas sistem seperti menampilkan data dilayar *user*. Tujuan utama EQ adalah menyajikan informasi kepada pengguna melalui pengambilan informasi data atau kontrol dari ILF dan EIF. Logika pemrosesan tidak berisi rumus atau perhitungan matematis dan tidak menghasilkan data turunan, serta tidak ada ILF yang dipertahankan selama pemrosesan dan tidak mengubah perilaku sistem; 4) *Internal Logical File* (ILF) merupakan sekelompok data informasi pengguna yang diidentifikasi dan tersimpan dalam batas aplikasi. Tujuan utama ILF adalah mempertahankan data yang dipelihara melalui satu atau lebih proses dasar dari aplikasi; 5) *External Interface Files* (EIF) merupakan sekumpulan data atau informasi yang dapat dikenali dan direferensikan oleh aplikasi, namun dipelihara dalam batas aplikasi lain. Tujuan utama EIF adalah menyimpan data yang direferensikan melalui satu atau lebih proses dasar di dalam batas aplikasi. Artinya, sebuah EIF yang dihitung untuk sebuah aplikasi harus berada dalam ILF aplikasi lain (Marthaler, 2005).

Langkah ketiga, penentuan bobot kompleksitas *Function Point* yang terdiri dari: 1) *Data Element Type* (DET) yaitu merupakan jenis elemen data yang bersifat unik yang dapat dikenali dan tidak berulang; 2) *Record Element Type* (RET) yaitu sub kelompok elemen data pengguna yang dikenali dalam ILF atau EIF; 3) *File Type Reference* (FTR) yaitu berkas *logical* internal yang dibaca atau dikelola oleh fungsi transaksional atau *file* antarmuka eksternal yang dibaca oleh fungsi transaksional.

Kemudian untuk mempermudah dalam mengidentifikasi FTR maka dilakukan dengan cara mengidentifikasi ILF dan EIF dikarenakan memiliki jumlah yang sama. Bobot kompleksitas tiap elemen fungsi disajikan pada Tabel 2.7, Tabel 2.8 dan Tabel 2.9.

Tabel 2.7 Bobot Kompleksitas *External Input*

FTR	DET		
	1-4	5-15	16 >
0-1	Low	Low	Average
2	Low	Average	High
= 3	Average	High	High

Sumber: Marthaler (2005)

Tabel 2.8 Bobot Kompleksitas *External Output* dan *External Inquiries*

FTR	DET		
	1-5	6-19	20 >
0-1	Low	Low	Average
2-3	Low	Average	High
4 >	Average	High	High

Sumber: Marthaler (2005)

Tabel 2.9 Bobot Kompleksitas *Logical Internal File* dan *External Interface File*

FTR	DET		
	1-19	20-50	51 >
1	Low	Low	Average
2-5	Low	Average	High
>= 6	Average	High	High



Sumber: Marthaler (2005)

Selanjutnya menentukan nilai *Unadjusted Function Point*, nilai dari setiap pembobotan kompleksitas pada elemen *Function Point* yang diperoleh akan dikalikan dengan masing-masing bobot kompleksitas. Bobot kompleksitas berdasarkan kriteria dari setiap kategori ditunjukkan pada Tabel 2.10.

Tabel 2.10 Perhitungan *Unadjusted Function Point*

<i>Elemen FP</i>	<i>Low</i>	<i>Average</i>	<i>High</i>	<i>Total</i>
<i>External Input</i>	3 x ...	4 x ...	6 x ...	
<i>External Output</i>	4 x ...	5 x ...	7 x ...	
<i>External Inquiries</i>	3 x ...	4 x ...	6 x ...	
<i>Internal Logical Files</i>	7 x ...	10 x ...	15 x ...	
<i>External Interface Files</i>	5 x ...	7 x ...	10 x ...	
<i>Unadjusted Function Point (UFP)</i>				

Sumber: Marthaler (2005)

Penelitian ini juga menggunakan aturan penentuan bobot kompleksitas dan identifikasi *Data Element Type* (DET), *Record Element Type* (RET) dan *File Type Reference* (FTR) yang akan disajikan pada Tabel 2.11.

Tabel 2.11 Aturan Perhitungan Bobot Kompleksitas

Tahap	Aturan
1	<p>Aturan perhitungan ILF</p> <ol style="list-style-type: none"> 1. Sekelompok data atau informasi yang dapat diidentifikasi pengguna 2. Sekelompok data yang dipelihara melalui proses dasar di dalam batas aplikasi
2	<p>Aturan perhitungan EIF</p> <ol style="list-style-type: none"> 1. Sekelompok data atau informasi yang dapat diidentifikasi oleh pengguna 2. Sekelompok data yang direferensikan dari luar aplikasi 3. Sekelompok data yang tidak dikelola oleh aplikasi 4. Sekelompok data yang dipelihara dalam ILF aplikasi lain
3	<p>Aturan perhitungan kompleksitas ILF</p> <p>Aturan perhitungan DET untuk ILF dan EIF</p> <ol style="list-style-type: none"> 1. Hitung DET untuk setiap informasi pengguna yang dapat dikenali dan tidak berulang serta disimpan atau diambil dari ILF atau EIF melalui proses dasar (misalnya nomor akun yang disimpan di beberapa <i>field</i> dihitung sebagai 1 DET, hasil perhitungan dari proses dasar misal menghitung nilai pajak penjualan untuk pesanan pelanggan yang disimpan dalam ILF dihitung sebagai 1 DET pada ILF pesanan pelanggan, mengakses harga item yang disimpan ke file tagihan oleh pengguna dihitung sebagai 1 DET) 2. Bila dua aplikasi menyimpan dan/atau mereferensikan



Tabel 2.11 Aturan Perhitungan Bobot Kompleksitas (Lanjutan)

	Tahap	Aturan
		<p>ILF atau EIF yang sama, namun masing-masing menyimpan/mereferekan DET secara terpisah, maka terhitung hanya DET yang digunakan (misalnya aplikasi A secara khusus mengidentifikasi dan menggunakan alamat sebagai: alamat jalan, kota, negara dan kode pos. Aplikasi B dapat melihat alamat sebagai 1 blok data tanpa memperhatikan komponen secara individual. Artinya aplikasi A akan menghitung 4 DET, aplikasi B menghitung 1 DET)</p> <p>3. Hitung DET untuk setiap potongan data yang dibutuhkan oleh pengguna untuk menjalin hubungan dengan ILF atau EIF lainnya (misalnya dalam aplikasi SDM, informasi karyawan disimpan dalam ILF. Nama pekerjaan karyawan disertakan sebagai bagian dari informasi karyawan. DET ini terhitung karena diperlukan untuk menghubungkan seorang karyawan dengan pekerjaan yang ada dalam organisasi. Jenis elemen data ini disebut sebagai <i>foreign key</i>)</p> <p>Aturan perhitungan RET untuk ILF dan EIF</p> <p>1. Hitung RET untuk setiap sub kelompok opsional atau wajib dari ILF atau EIF</p> <p>ATAU</p> <p>Jika tidak ada sub kelompok, hitung ILF atau EIF sebagai 1 RET</p>
4	Aturan perhitungan EO (External Output)	<p>1. Logika pemrosesan dari proses dasar yang mengandung setidaknya 1 rumus atau perhitungan matematis</p> <p>2. Logika pemrosesan dari proses dasar yang menciptakan data turunan</p> <p>3. Logika pemrosesan dari proses dasar yang mempertahankan setidaknya 1 ILF</p> <p>4. Logika pemrosesan dari proses dasar yang mengubah perilaku sistem</p>
5	Aturan perhitungan EQ (External Inquiry)	<p>1. Logika pemrosesan dari proses dasar harus mengambil data atau kontrol dari ILF atau EIF</p> <p>2. Logika pemrosesan dari proses dasar tidak mengandung rumus atau perhitungan matematis</p> <p>3. Logika pemrosesan dari proses dasar tidak menciptakan data turunan</p> <p>4. Logika pemrosesan dari proses dasar tidak mengubah perilaku sistem</p>
6	Aturan perhitungan kompleksitas EI (External Input)	<p>Aturan perhitungan FTR untuk EI:</p> <p>1. Hitung FTR untuk setiap ILF yang dipertahankan</p> <p>2. Hitung FTR untuk setiap ILF atau EIF yang dibaca selama pemrosesan EI</p> <p>3. Hitung 1 FTR untuk setiap ILF yang dipertahankan dan dibaca selama pemrosesan EI</p> <p>Aturan perhitungan DET untuk EI:</p> <p>1. Hitung 1 DET untuk setiap <i>field</i> yang tidak berulang yang dapat dikenali yang masuk atau keluar dari batas aplikasi</p>

Tabel 2.11 Aturan Perhitungan Bobot Kompleksitas (Lanjutan)

	Tahap	Aturan
		<p>dan diminta untuk melengkapi EI (contoh: <i>jobname</i> dan <i>paygrade</i> adalah 2 <i>field</i> yang diberikan pengguna saat menambahkan pekerjaan)</p> <ol style="list-style-type: none"> 2. Jangan menghitung <i>field</i> yang diambil atau turunan dari sistem dan disimpan di ILF serta tidak melewati batas aplikasi (contoh: ketika pesanan pelanggan ditambahkan ke sistem, harga unit secara otomatis diambil untuk setiap item yang dipesan dan disimpan pada catatan tagihan. Harga satuan tidak akan dihitung sebagai DET untuk EI karena tidak melewati batas saat pengguna menambahkan pesanan pelanggan) 3. Hitung 1 DET untuk kemampuan mengirim pesan notifikasi seperti pesan kesalahan saat proses terjadi, pesan konfirmasi bahwa pemrosesan telah selesai atau verifikasi bahwa pemrosesan harus dilanjutkan (contoh: terhitung 1 DET yaitu yang mencakup semua notifikasi dari sistem yang menunjukkan kondisi kesalahan, konfirmasi bahwa pemrosesan telah selesai atau verifikasi bahwa pemrosesan harus dilanjutkan) 4. Hitung 1 DET untuk kemampuan menentukan tindakan yang akan diambil meskipun ada beberapa metode lain untuk memanggil proses yang sama (contoh: jika pengguna dapat melakukan tambah karyawan dengan mengklik tombol OK atau dengan menekan tombol PF, itu dihitung 1 DET karena menggambarkan kemampuan memulai proses)
7	<p>Aturan perhitungan kompleksitas EO</p>	<p>Aturan perhitungan FTR untuk EO/EQ:</p> <ol style="list-style-type: none"> 1. Hitung 1 FTR untuk setiap ILF atau EIF yang dibaca selama pemrosesan 2. Hanya ada 1 FTR untuk setiap ILF yang dipertahankan selama pemrosesan 3. Hitung hanya 1 FTR untuk setiap ILF yang dipertahankan dan dibaca selama pemrosesan <p>Aturan perhitungan DET untuk EO/EQ:</p> <ol style="list-style-type: none"> 1. Hitung 1 DET untuk setiap <i>field</i> yang dapat dikenali dan tidak berulang dan memasuki batas aplikasi dan dapat menentukan kapan dan bagaimana data diambil atau dihasilkan oleh proses (contoh: untuk menghasilkan daftar karyawan, nama karyawan adalah <i>field</i> yang diberikan pengguna saat menunjukkan daftar karyawan mana yang akan diajukan) 2. Hitung 1 DET untuk setiap bidang yang dapat dikenali dan tidak berulang dan keluar dari batas aplikasi (contoh: nomor rekening atau tanggal yang tersimpan secara fisik di berbagai <i>field</i> dihitung sebagai 1 DET. Pada diagram lingkaran memiliki label kategori dan numerik, itu terhitung 2 DET – 1 DET untuk menentukan kategori, 1 DET untuk nilai numerik) 3. Jika DET masuk dan keluar dari batas aplikasi, hitung





	<p>hanya satu kali untuk proses dasar</p> <ol style="list-style-type: none"> 4. Hitung 1 DET untuk kemampuan mengirim pesan notifikasi di luar batas aplikasi untuk menunjukkan adanya kesalahan saat pemrosesan, konfirmasi bahwa pemrosesan telah selesai atau verifikasi bahwa pemrosesan harus dilanjutkan (contoh: jika pengguna mencoba meminta daftar namun tidak memiliki hak akses ke informasi tersebut, maka hal itu terhitung 1 DET untuk respon sistem) 5. Hitung 1 DET untuk kemampuan menentukan tindakan yang akan diambil meskipun ada beberapa metode untuk memanggil proses yang sama (misalnya jika pengguna dapat memulai buat laporan dengan mengklik tombol OK atau dengan menekan tombol PF, itu terhitung 1 DET untuk kemampuan memulai laporan) 6. Jangan menghitung <i>field</i> yang diambil atau diturunkan oleh sistem dan disimpan di ILF selama pemrosesan berlangsung serta tidak melewati batas aplikasi (contoh: ketika slipgaji dicetak, <i>field</i> status pada karyawan dalam ILF diperbarui untuk menunjukkan bahwa slipgaji telah dicetak. Jangan menghitung <i>field</i> status sebagai DET karena tidak melewati batas) 7. Jangan menghitung literal pada DET misalnya judul laporan, identifikasi layar, judul kolom, judul lapangan 8. Jangan hitung variabel <i>paging</i> atau perangko yang dihasilkan oleh sistem (misalnya nomor halaman, <i>field</i> tanggal dan waktu yang ditampilkan)
--	---

Sumber: Marthaler (2005)

Langkah selanjutnya, melakukan identifikasi karakteristik perangkat lunak atau *Technical Complexity Factor* (TCF). Kemudian untuk menentukan nilai TCF, yang perlu dilakukan adalah menentukan faktor teknis dengan cara memberikan nilai pada tiap faktor dengan bobot antara 0 sampai 5.

Tabel 2.12 Bobot *Technical Complexity Factor*

Score	System Influence
0	No Influence
1	Incidental Influence
2	Moderate Influence
3	Average Influence
4	Significant Influence
5	Strong Influence

Sumber: Marthaler (2005)

Tabel 2.13 *Technical Factors*

Subject		Grade					
1	Data Communications	0	1	2	3	4	5
2	Distributed Data Processing	0	1	2	3	4	5
3	Performance	0	1	2	3	4	5
4	Heavily Used Configuration	0	1	2	3	4	5
5	Transaction Rate	0	1	2	3	4	5



Tabel 2.13 Technical Factors (Lanjutan)

6	Subject	Grade					
		0	1	2	3	4	5
6	Online Data Entry	0	1	2	3	4	5
7	End-User Efficiency	0	1	2	3	4	5
8	Online Update	0	1	2	3	4	5
9	Complex Processing	0	1	2	3	4	5
10	Reusability	0	1	2	3	4	5
11	Installation Ease	0	1	2	3	4	5
12	Operation Ease	0	1	2	3	4	5
13	Multiple Sites	0	1	2	3	4	5
14	Facilitate Change	0	1	2	3	4	5
Total							

Sumber: Marthaler (2005)

$$TCF = 0.65 + 0.01 \times RCAF \tag{2.6}$$

Setelah memperoleh nilai *Unadjusted Function Point* dan nilai *Technical Complexity Factor* maka langkah selanjutnya menentukan nilai dari *Function Point* sebuah perangkat lunak yang akan dikembangkan yaitu dengan cara memasukkan nilai *Unadjusted Function Point* dan *Technical Complexity Factor* pada Persamaan 2.7.

$$FP = UFP \times TCF \tag{2.7}$$

2.12.2 Estimasi Effort (Usaha)

Setelah memperoleh nilai *Function Point*, maka langkah selanjutnya menghitung usaha yang diperlukan untuk pengembangan perangkat lunak berdasarkan *person-month*. Perhitungan ini menggunakan fungsi estimasi eksponensial oleh Jones ada pada Persamaan 2.8 dengan *f* merupakan nilai *Function Point*, *j* merupakan nilai *Jone's First Order*.

$$m = f^{3 \cdot j} / 27 \tag{2.8}$$

Tabel 2.14 Jone's First Order Estimate Exponent

Kind of Software	Organization's Skill/Abilities		
	Best In Class	Average	Worst In Class
System	0.43	0.45	0.48
Business	0.41	0.43	0.46
Shrink-wrap	0.39	0.42	0.45

Sumber: Connel (2006)

Jenis-jenis perangkat lunak yaitu: (1) **System**, yaitu *operating systems, device drivers, compiler, embedded software, firmware, real-time system*, dan lain-lain; (2) **Bussines**, in *house system for an organization, payroll, stock control, accounting system*, dan lain-lain; (3) **Shrink-wrap**, *packed and sold commercially* (Connel, 2006).

Kemampuan atau keterampilan organisasi yaitu: (1) **Best in Class**, yaitu perusahaan atau orang-orang yang memiliki kemampuan diatas rata-rata dengan pengalaman yang cukup banyak dalam pengembangan perangkat lunak; (2) **Average**, yaitu perusahaan atau orang-orang yang memiliki kemampuan rata-rata dalam menangani pengembangan perangkat lunak; (3) **Worst in class**, yaitu perusahaan atau orang-orang yang memiliki kemampuan tidak cukup baik dan tidak memiliki pengalaman dalam menangani pengembangan



perangkat lunak. Menghitung lama durasi proyek menggunakan Persamaan 2.9 dengan s merupakan *schedule months for optimal schedule*, f merupakan nilai *effort* dan j merupakan nilai *Jone's First Order* (Connel, 2006).

$$s = f^j \quad (2.9)$$

2.12.3 Estimasi Biaya

Effort yang telah didapatkan pada tahap sebelumnya dibagi menjadi tiga aktivitas yang dijalankan dalam pengembangan perangkat lunak yaitu *Software Development*, *Ongoing Activity* dan *Quality dan Testing*.

1. *Software Development*, meliputi analisis kebutuhan pengguna seperti permintaan dan spesifikasi, desain, implementasi, pengujian integrasi yang mencapai sekitar 42% dari total *effort*.
2. *Ongoing Activity*, atau disebut aktivitas yang berkesinambungan seperti manajemen proyek, manajemen konfigurasi, dokumentasi, penerimaan dan penyebaran yang mencapai sekitar 21% dari total *effort*.
 - 1) *Project Management*, PM melakukan kegiatan yang berhubungan dengan manajemen proyek meliputi memantau kemajuan proyek secara ketat, melaporkan manajemen resiko, memperbarui jadwal proyek secara berkala.
 - 2) *Configuration Management*, melakukan pemeliharaan, pengelolaan, pengendalian dan pengembangan perangkat lunak. Idealnya kegiatan tersebut dilakukan dengan menggunakan alat konfigurasi otomatis yang memperbarui status secara berkala. Hasil alat tersebut kemudian dianalisis oleh tim manajemen tertentu.
 - 3) *Documentation*, membuat buku manual instalasi, operasi atau *template* standar. Biasanya ditulis oleh pengembang perangkat lunak.
 - 4) *Training & Support*, pada fase ini tim pengembang membutuhkan beberapa dukungan teknis dan pelatihan dalam memecahkan masalah teknis yang mungkin terjadi guna meningkatkan efisiensi dan produktifitas.
3. *Quality dan Testing*, aktivitas yang berhubungan dengan kualitas dan pengujian seperti pengujian integrasi, penjaminan kualitas, evaluasi dan pengujian yang mencapai sekitar 37% dari total *effort*.
 - 1) *Integration Testing*, melakukan uji integrasi antara modul, sistem dan komponen eksternal. Kemudian melakukan hasil analisis dan menangani masalah yang terjadi. *Milestone* dari fase ini adalah perangkat lunak terintegrasi.
 - 2) *Quality Assurance*, memastikan seluruh aktivitas yang terjadi pada setiap fase telah sesuai dengan standar internal dan menghasilkan *deliverables* yang benar.
 - 3) *Evaluation & Testing*, melakukan kegiatan evaluasi dan pengujian, termasuk kegiatan validasi dan verifikasi. Proses ini dilakukan oleh tim pengembang (Saleh, 2011).


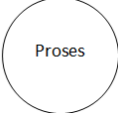
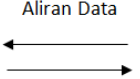



2.13 Data Flow Diagram

Data Flow Diagram (DFD) merupakan gambaran aliran informasi serta transformasi yang diaplikasikan sebagai pindahnya data dari *input* ke *output*. DFD juga digunakan untuk merepresentasikan sebuah perangkat lunak ke level abstraksi. DFD menyediakan mekanisme pemodelan fungsional dan pemodelan aliran informasi. DFD dibagi menjadi beberapa level yang menggambarkan bertambahnya aliran informasi dan detail fungsional.

Data Flow Diagram dibagi menjadi beberapa tingkat level yaitu: 1) Level 0 atau *Context Diagram* merupakan level tertinggi dari suatu *Data Flow Diagram*, yang mempresentasikan seluruh elemen pada perangkat lunak sebagai sebuah lingkaran dengan *input* dan *output* dengan arah panah masuk dan keluar; 2) Level 1 merupakan dekomposisi dari *Data Flow Diagram* Level 0, pada Level 1 mungkin terdiri dari 5 atau 6 lingkaran dengan panah yang saling terhubung; 3) Level 2 merupakan dekomposisi dari proses-proses yang terdapat dalam *Data Flow Diagram* Level 1; dan seterusnya (Pressman, 2010).

Tabel 2.15 Komponen Data Flow Diagram

Simbol	Deskripsi
	Entitas eksternal berupa orang atau unit yang berinteraksi dengan sistem namun berada di luar sistem.
	Orang atau unit yang menggunakan atau mentransformasikan data. Komponen fisik tidak dapat didefinisikan.
	Aliran data dari sumber ke tujuan yang arahnya bersifat khusus.
	Tempat penyimpanan data yang dapat dilihat oleh proses.

Sumber: Shalahuddin et al. (2014)

2.14 Peran dan Tanggung Jawab Profesi TI

Peran dan tanggung jawab profesi pada TI yang mengacu pada *Project Team Assignment* (Form Penugasan Tim Proyek) yang sesuai dengan deskripsi fase-fase pada metode *Use Case Point* dan *Function Point* yaitu salah satunya *Project Manager*, *System Analyst*, *Tester* atau *Test Analyst*.

Project Manager bertugas: 1) Membuat rencana proyek; 2) Menugaskan pekerjaan ke anggota tim proyek; 3) Mengawasi kerja tim proyek dan jalannya proyek; 4) Membuat laporan *progress*; 5) Melaporkan setiap kemajuan proyek beserta hasilnya kepada *stakeholder*; 6) Memastikan proyek berjalan sesuai jadwal dan ruang lingkup yang telah disetujui; 7) Memastikan proyek berjalan sukses; 8) Menjembatani *System Analyst* dengan *Software Engineer* agar sistem yang dihasilkan sesuai dengan yang diharapkan.

System Analyst bertugas: 1) Melakukan analisis sistem dengan cara wawancara dengan *stakeholder*; 2) Membuat *Context Diagram*, *DFD Levelled*, *ERD*; 3) Melaporkan hasil pekerjaan ke *Project Manager*; 4) Membuat dokumentasi analisis dan desain sistem.

Tester atau *Test Analyst* bertugas: 1) Melakukan pengujian terhadap program aplikasi yang telah dibuat; 2) Membuat *list* mengenai hal-hal yang perlu diperbaiki; 3) Memberikan



saran atau solusi terhadap masalah yang dihadapi; 4) Membuat dokumentasi mengenai hasil pengujian terhadap program aplikasi (Yunautama, 2008).

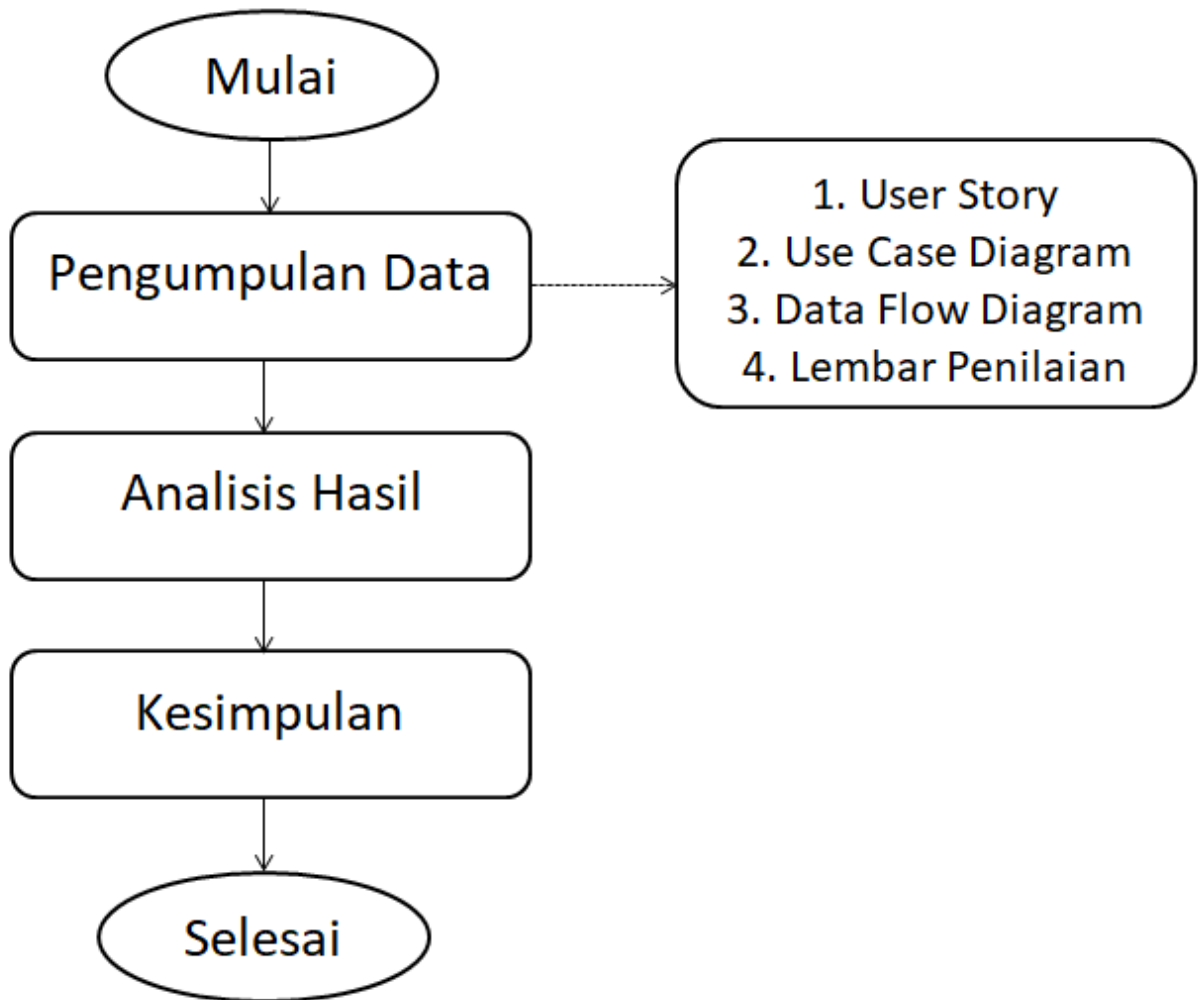
Software Engineer bertugas: 1) Melakukan perancangan dan pengembangan berbagai jenis *software* (termasuk OS, jaringan terdistribusi); 2) Mengubah program agar dapat dieksekusi di *hardware*; 3) Melakukan analisis logika matematis dalam fase pengembangan; 4) Memungkinkan untuk melakukan pengujian dan evaluasi sistem (Farr, 2007).

2.15 Metode *Guesstimate*

Metode *Guesstimate* merupakan menentukan estimasi jadwal, anggaran proyek dengan menebak atau hanya memperkirakan (*picking number out of the air*). *Guesstimate* bukan cara terbaik untuk menentukan jadwal dan anggaran suatu proyek, karena perkiraan lebih didasarkan perasaan bukan bukti dan kebanyakan orang terlalu optimis dengan *Guesstimate* yang mereka tentukan. (Marchewka, 2003).

BAB 3 METODOLOGI

Metode penelitian yang dilakukan dimulai dari studi pustaka, pengumpulan data, kemudian analisis hasil menggunakan pendekatan WBS dan *Gantt Chart* serta metode estimasi biaya yaitu *Use Case Point* dan *Function Point*. Berdasarkan jenis data yang diperlukan secara umum, penelitian ini merupakan penelitian primer yaitu penelitian yang membutuhkan data atau informasi melalui lembar penilaian atau wawancara yang berasal dari sumber pertama atau biasa disebut dengan responden (Sarwono, 2006). Penelitian ini menggunakan pendekatan campuran yaitu kuantitatif dan kualitatif. Kuantitatif yaitu penelitian yang bertujuan untuk menguji teori, menunjukkan hubungan antar variabel, membangun fakta, mendeskripsikan fakta dan meramal hasilnya. Sedangkan kualitatif yaitu penelitian yang bertujuan untuk mengembangkan pengertian, konsep-konsep yang akhirnya menjadi teori yang dikenal sebagai "*Grounded Theory Research*" (Sarwono, 2006). Alur penelitian yang dilakukan dalam penelitian ini disajikan pada Gambar 3.1.



Gambar 3.1 Alur Penelitian



3.1 Studi Pustaka

Studi pustaka dilakukan dengan mengumpulkan informasi yang sumbernya dari buku, jurnal penelitian dan internet. Tujuannya yaitu untuk menggali teori-teori maupun informasi yang dapat mendukung penelitian. Informasi yang didapat mengenai hal-hal berikut: (1) Manajemen proyek sistem informasi berupa penjelasan umum manajemen proyek; (2) *Work Breakdown Structure* (WBS) mengenai penjabaran sebuah pekerjaan; (3) Metode *Use Case Point* berupa penjelasan umum dan perhitungan *Use Case Point*; (4) Metode *Function Point* berupa penjelasan umum dan perhitungan *Function Point*.

3.2 Pengumpulan Data

Pada penelitian ini, data yang digunakan yaitu data kualitatif dan kuantitatif. Data kualitatif merupakan data yang bersifat deskriptif atau data yang dikategorikan seperti diagram, catatan-catatan lapangan atau wawancara ketika penelitian dilakukan. Sedangkan data kuantitatif merupakan data yang bersifat kuantitatif atau angka-angka statistik yang dapat berbentuk variabel-variabel dan operasionalisasinya menggunakan skala ukuran tertentu (Sarwono, 2006). Data kuantitatif yang didapat yaitu lembar penilaian *Technical* dan *Environment Complexity Factor*. Sedangkan data kualitatif yang didapat berupa hasil wawancara yang dibutuhkan dalam proses pembuatan *User Story*, *Use Case Diagram* dan *Data Flow Diagram*. Data yang dibutuhkan diperoleh langsung dari CV Pabrik Teknologi.

Selain itu tujuan dari pengumpulan data yaitu untuk mengetahui informasi mengenai manajemen proyek yang ada di CV Pabrik Teknologi, informasi mengenai *software* yang diselesaikan oleh CV Pabrik Teknologi dan informasi mengenai alokasi SDM dan *timeline* pengerjaan Sistem Antrian RSUD Dr. Soetrasno Rembang.

3.2.1 Wawancara

Penelitian ini membutuhkan pengumpulan data melalui wawancara, yaitu membutuhkan ketersediaan dari *Project Manager* untuk dimintai keterangan mengenai proyek yang akan diselesaikan. Wawancara ini berisi beberapa pertanyaan mengenai masalah yang dihadapi ketika pengerjaan proyek, sistem yang akan dikembangkan seperti analisis kebutuhan fungsionalitas sistem yang hasilnya akan diubah ke bentuk *User Story*, *Use Case Diagram* dan *Data Flow Diagram* karena penelitian ini membutuhkan *Use Case Diagram* untuk menentukan bobot dari *Unadjusted Use Case Point* dan *Data Flow Diagram* untuk *Unadjusted Function Point*. Selain itu dalam wawancara ini menanyakan beberapa hal mengenai proyek tersebut seperti cara penentuan nilai proyek, alokasi SDM dan waktu (*timeline*) guna memahami perencanaan proyek yang ada pada CV Pabrik Teknologi. Laporan hasil wawancara terlampir pada Lampiran A.

3.2.2 Lembar Penilaian

Penelitian ini juga menggunakan lembar penilaian untuk mengambil data di CV Pabrik Teknologi. Mekanisme penyebaran lembar penilaian untuk metode *Use Case Point* dan metode *Function Point* menggunakan tiga tahapan.

Tahap pertama yaitu melakukan uji validitas lembar penilaian, pada tahap ini dilakukan pengukuran validitas guna memperkuat keakuratan data lembar penilaian. Pengukuran validitas menggambarkan apakah data yang dikumpulkan benar-benar mengukur apa yang



peneliti inginkan. Penelitian ini melakukan dua jenis pengukuran validitas yaitu *Face Validity* dan *Content Validity* yang dilakukan oleh dua orang *expert judgement*.

- 1) *Face Validity*, untuk validitas ini bisa dilakukan oleh seorang ahli yang memiliki pengetahuan yang baik guna menilai seberapa baik item-item lembar penilaian dan mengukur apakah sudah sesuai dengan definisi-definisi yang relevan.
- 2) *Content Validity*, validitas ini mengacu pada seberapa baik konten lembar penilaian. Validitas ini penting dilakukan karena peneliti memiliki banyak pilihan dalam membuat konten lembar penilaian. Penilaian konten dapat dilakukan dengan memberikan *rating* oleh ahli (*expert*) (Recker, 2012).

Kemudian tahap kedua, penyebaran lembar penilaian. Lembar penilaian disebar kepada responden yang dalam penelitian ini adalah tim pengembang dari CV Pabrik Teknologi yang akan mengembangkan Sistem Antrian RSUD Dr. Soetrasno Rembang. Lembar penilaian disebar untuk mendapatkan informasi terkait perhitungan nilai TCF dan ECF pada *Use Case Point*, serta nilai TCF pada *Function Point*.

Tahap terakhir yaitu mendapatkan skor pada setiap lembar penilaian untuk metode *Use Case Point* dan *Function Point*. Pada metode *Use Case Point*, skor di setiap indikator akan dikalikan dengan bobot yang hasil akhirnya berupa nilai TF untuk lembar penilaian *Technical Complexity Factor* dan nilai EF untuk lembar penilaian *Environment Complexity Factor*. Pada metode *Function Point* seluruh skor di setiap indikator akan dijumlahkan sehingga mendapatkan nilai RCAF untuk lembar penilaian *Technical Complexity Factor*.

3.3 Analisis Hasil

Analisis hasil penelitian ini dilakukan menggunakan *Gantt Chart* berdasarkan *Work Breakdown Structure* untuk menjabarkan penjadwalan pengerjaan Sistem Antrian RSUD Dr. Soetrasno Rembang. Kemudian menggunakan metode perhitungan *Use Case Point* dan metode *Function Point*. Kemudian akan dilakukan perbandingan terhadap kedua metode dan *Guesstimate*. Tahapan perhitungan dengan *Use Case Point* disajikan pada Tabel 3.1 dan perhitungan *Function Point* disajikan pada Tabel 3.2.

Tabel 3.1 Alur Perhitungan *Use Case Point*

Tahap	Input	Proses	Output
1	<i>Use Case Diagram</i>	<ol style="list-style-type: none"> 1. Menghitung <i>Unadjusted Actor Weights</i> (UAW) 2. Menghitung <i>Unadjusted Use Case Weights</i> (UUCW) 	Nilai untuk perhitungan <i>Unadjusted Use Case Point</i> (UUCP)
2	<ol style="list-style-type: none"> 1. Nilai <i>Unadjusted Actor Weights</i> (UAW) 2. Nilai <i>Unadjusted Use Case Weights</i> (UUCW) 	Perhitungan nilai <i>Unadjusted Use Case Point</i> (UUCP) yaitu nilai <i>Unadjusted Actor Weights</i> (UAW) dan nilai <i>Unadjusted Use Case Weights</i> (UUCW)	Nilai <i>Unadjusted Use Case Point</i> (UUCP)
3	Lembar penilaian <i>Technical Complexity Factor</i>	Perhitungan hasil lembar penilaian yaitu bobot <i>Distributed System Required, Response Time, End User Efficiency, Complex Internal</i>	Nilai untuk perhitungan <i>Complexity Factor</i>

Tabel 3.1 Alur Perhitungan *Use Case Point* (Lanjutan)

Tahap	Input	Proses	Output
		<i>Processing Required, Reusable Code, Easy to Install, Easy to Use, Portable, Easy to Change, Concurrent, Security Features, Access for Third Parties dan Special Training Required.</i>	
4	Lembar penilaian <i>Environment Complexity Factor</i>	Perhitungan hasil lembar penilaian yaitu bobot <i>Familiarity With The Project, Application Experience, OO Programming Experience, Lead Analyst Capability, Motivation, Stable Requirements, Part Time Staff, Difficult Programming Language</i>	Nilai untuk perhitungan <i>Complexity Factor</i>
5	Nilai <i>Technical Complexity Factor (TCF)</i> dan <i>Environment Complexity Factor (ECF)</i>	Perhitungan <i>Complexity Factor</i> yaitu nilai TCF dan nilai ECF	Nilai <i>Complexity Factor</i>
6	Nilai <i>Unadjusted Use Case Point (UUCP)</i> , nilai <i>Complexity Factor</i>	Melakukan perhitungan nilai <i>Unadjusted Use Case Point (UUCP)</i> , nilai <i>Complexity Factor</i>	Nilai <i>Use Case Point (UCP)</i>
7	Nilai <i>Use Case Point (UCP)</i>	Konversi nilai <i>Use Case Point (UCP)</i> ke <i>hours of effort</i>	Nilai <i>hours of effort</i>
8	Nilai <i>hours of effort</i>	Membagi <i>effort</i> ke dalam 3 (tiga) aktivitas yaitu <i>Software Development, Ongoing Activity, Quality and Testing</i>	tiga aktivitas utama
9	Tiga aktivitas utama	Perhitungan biaya per aktivitas berdasarkan standar gaji <i>Kelly Service, Inc.</i>	Nilai estimasi biaya per aktivitas
10	Nilai estimasi biaya per aktivitas	Perhitungan estimasi biaya total	Estimasi biaya total

Sumber: Diadaptasi dari Sholih (2013)

Keterangan:

- Melakukan proses pembuatan *Use Case Diagram* dan *Use Case Scenario* sesuai dengan hasil wawancara dan validasi diagram dengan narasumber dari CV Pabrik Teknologi. Setelah itu melakukan analisis terhadap *Use Case Diagram* dan *Use Case Scenario* untuk mendapatkan nilai *Unadjusted Use Case Weight (UUCW)* yaitu mengukur kompleksitas dari *use case* apakah *Simple, Average* atau *Complex*, serta nilai *Unadjusted Actor Weight (UAW)* yaitu mengukur kompleksitas dari aktor apakah *Simple, Average* atau *Complex*. Kemudian langkah berikutnya memasukkan nilai kompleksitas ke dalam persamaan rumus *Use Case Weight (UUCW)* dan *Unadjusted Actor Weight (UAW)*.



2. Pada langkah ini, nilai *Use Case Weight* (UUCW) dan nilai *Unadjusted Actor Weight* (UAW) akan dihitung menggunakan persamaan rumus untuk mendapatkan nilai *Unadjusted Use Case Point* (UUCP). Penelitian ini menggunakan aturan penentuan transaksi berdasarkan *actor goals* atau disebut *semantic point*. Jika melihat transaksi dari *semantic point*, sebagian besar use case memiliki *goals* atau tujuan yang sama. *Semantic point* telah membedakan 12 jenis transaksi yaitu *Create, Retrieve, Update, Delete, Link, Delete Link, Asynchronous Retrieve, Dynamic Retrieve, Dynamic Retrieve, Transfer, Check Object, Complex Internal Activity* dan *Change State* (Ochodek et al., 2011).
3. Selanjutnya mendapatkan nilai *Technical Complexity Factor* (TCF). Untuk mengetahui faktor-faktor dan skalanya, telah dijelaskan pada Bab 2 mengenai beberapa indikator dan bobot untuk setiap faktor teknis. Pemberian nilai terhadap masing-masing faktor membutuhkan penilaian objektif melalui penyebaran lembar penilaian kepada beberapa responden, responden dari lembar penilaian ini adalah tim pengembang Sistem Antrian RSUD Dr. Soetrasno Rembang.
4. Langkah berikutnya mendapatkan nilai *Environment Complexity Factor* (ECF). Untuk mengetahui faktor-faktor dan skalanya, telah dijelaskan pada Bab 2 mengenai beberapa indikator dan bobot untuk setiap faktor lingkungan. Untuk pemberian nilai terhadap masing-masing faktor membutuhkan penilaian objektif melalui penyebaran lembar penilaian kepada beberapa responden, responden dari lembar penilaian ini adalah tim pengembang Sistem Antrian RSUD Dr. Soetrasno Rembang.
5. Selanjutnya, melakukan penjumlahan nilai terhadap faktor-faktor dari kedua lembar penilaian sehingga mendapatkan nilai *Technical Complexity Factor* (TCF) dan nilai *Environment Complexity Factor* (ECF).
6. Pada langkah ini, nilai *Unadjusted Use Case Point* (UUCP) dan nilai *Complexity Factor* dimasukkan kepada perhitungan rumus sehingga menghasilkan nilai *Use Case Point* (UCP).
7. Kemudian nilai *Use Case Point* (UCP) dikonversi ke *hours of effort* dengan cara mengalikan nilai *Use Case Point* (UCP) dengan nilai *staff-hour per use case point*.
8. Setelah mendapatkan nilai *hours of effort*, maka langkah selanjutnya membagi *effort* ke dalam tiga aktivitas yaitu *Software Development, Ongoing Activity, Quality and Testing*.
9. Selanjutnya menghitung biaya per aktivitas yang diperoleh dengan mengalikan total gaji pada tiap aktivitas. Standar gaji yang digunakan mengacu kepada *Indonesia Salary Guide 2016* oleh *Kelly Service*.
10. Hasil akhir dari metode ini adalah estimasi biaya yang merepresentasikan jumlah biaya dalam satuan Rupiah.

Tabel 3.2 Alur Perhitungan *Function Point*

Tahap	Input	Proses	Output
1	Fungsi-fungsi proyek	Melakukan identifikasi fungsi-fungsi dengan menggunakan <i>Data Flow Diagram</i> (DFD) sebagai parameter proyek.	<i>Data Flow Diagram</i> (DFD)
2	<i>Data Flow Diagram</i> (DFD)	Menentukan Tipe Fungsi Pengguna	Nilai EI, EO, EQ, ILF dan EIF
3	Nilai EI, EO, EQ, ILF dan EIF	Penentuan bobot tiap fungsi yang sebelumnya telah di kelompokkan berdasarkan tingkat kompleksitas kemudian dikalikan dengan faktor kali kompleksitasnya.	Nilai <i>Unadjusted Function Point</i> (UFP)
4	<i>Complexity Adjustment Factor</i>	Menentukan bobot berdasarkan hasil lembar penilaian <i>Technical Complexity Factor</i> (TCF)	Nilai <i>Technical Complexity Factor</i> (TCF)
5	Nilai <i>Unadjusted Function Point</i> (UFP) dan Nilai <i>Technical Complexity Factor</i> (TCF)	Perhitungan nilai <i>Function Point</i> yaitu dengan persamaan nilai UFP dan TCF	Nilai <i>Function Point</i>
6	Nilai <i>Function Point</i>	Konversi nilai <i>Function Point</i> ke <i>person-month</i>	Nilai <i>person-month</i>
7	Nilai <i>person-month</i>	Membagi <i>effort</i> ke dalam 3 (tiga) aktivitas yaitu <i>Software Development, Ongoing Activity, Quality and Testing</i>	Tiga aktivitas utama
8	Tiga aktivitas utama	Perhitungan biaya per aktivitas berdasarkan standar gaji <i>Kelly Service, Inc.</i>	Nilai estimasi biaya per aktivitas
9	Nilai estimasi biaya per aktivitas	Perhitungan estimasi biaya total	Estimasi biaya total

Sumber: Diadaptasi dari Marthaler (2005)

Keterangan:

1. Pada langkah ini melakukan identifikasi fungsi-fungsi sebagai parameter dari *Data Flow Diagram*. Proses pembuatan *Data Flow Diagram* sesuai dengan hasil wawancara dengan narasumber dari CV Pabrik Teknologi yang kemudian dilakukan validasi diagram tersebut.
2. Kemudian dari *Data Flow Diagram* menentukan tipe fungsi pengguna yaitu *External Input* (EI), *External Output* (EO), *External Inquiry* (EQ), *Internal Logical File* (ILF) dan *External Interface File* (EIF).
3. Setelah mengelompokkan masing-masing fungsi maka diberi bobot sesuai dengan tingkat kompleksitasnya, kemudian setiap nilai dikali dengan bobot kompleksitas. Nilai total seluruh fungsi disebut nilai *Unadjusted Function Point*.



4. Pada langkah ini dilakukan perhitungan kompleksitas faktor teknik, dengan beberapa indikator pengukuran sesuai dengan penjelasan pada Bab 2. Pemberian skor pada masing-masing indikator membutuhkan ketersediaan seluruh tim pengembang Sistem Antrian RSUD Dr. Soetrasno Rembang. Pada tahap ini akan didapatkan nilai *Technical Complexity Factor* (TCF) dari penjumlahan seluruh skor dari setiap indikator.
5. Untuk mendapatkan nilai *Function Point*, maka dilakukan perkalian antara nilai *Unadjusted Function Point* (UFP) dan nilai *Technical Complexity Factor* (TCF).
6. Langkah selanjutnya yaitu menghitung estimasi usaha yaitu nilai *person-month* dengan persamaan yang terlampir pada Bab 2.
7. Sebelum menghitung estimasi biaya total yang dibutuhkan untuk pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang, maka nilai *person-month* yang telah didapatkan pada tahap sebelumnya dikelompokkan menjadi 3 aktivitas yaitu: 1) *Software Development*; 2) *Ongoing Activity*; dan 3) *Quality and Testing*.
8. Setelah itu melakukan estimasi biaya per aktivitas yang didapat dengan perkalian antara usaha dengan tarif standar personal berlaku dengan menggunakan standar internasional dari Kelly Service tahun 2016.
9. Langkah terakhir yaitu menjumlah seluruh estimasi biaya per aktivitas sehingga didapat nilai estimasi biaya total yang dibutuhkan untuk pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang.

3.4 Kesimpulan

Setelah dilakukannya analisis hasil, maka penelitian ini dapat mengambil kesimpulan yang dapat dijadikan bahan pertimbangan bagi CV Pabrik Teknologi dalam mengestimasi biaya.

BAB 4 HASIL

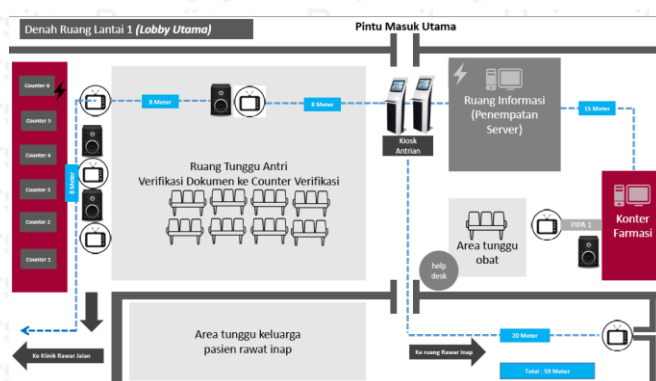
4.1 Mekanisme Wawancara

Langkah pertama dalam melakukan penelitian ini yaitu dengan melakukan wawancara. Tujuan dilakukannya wawancara adalah untuk pengumpulan data kualitatif dan kuantitatif mengenai proyek Sistem Antrian RSUD Dr. Soetrasno Rembang yang akan diselesaikan oleh CV Pabrik Teknologi. Penelitian ini membutuhkan ketersediaan dari *Project Manager* sebagai narasumber untuk dimintai keterangan mengenai proyek yang akan diselesaikan dikarenakan *Project Manager* mengetahui seluruh informasi terkait proyek Sistem Antrian RSUD Dr. Soetrasno Rembang.

4.2 Hasil Wawancara

Hasil dari wawancara yang dilakukan kepada *Project Manager* mengenai Sistem Antrian RSUD Dr. Soetrasno Rembang. Informasi yang didapat melalui kegiatan wawancara ini yaitu data kualitatif mengenai kebutuhan fungsionalitas Sistem Antrian yang akan dikembangkan guna pembuatan *User Story*, *Use Case Diagram* dan *Data Flow Diagram*, serta mengenai informasi terkait manajemen proyek dan penjadwalan pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang seperti alokasi waktu dan sumber daya manusia yang terlampir pada Lampiran A.

Proses bisnis pada RSUD Dr. Soetrasno Rembang yang dijadikan sebagai salah satu acuan pengerjaan proyek oleh CV Pabrik Teknologi yaitu Pasien datang melalui pintu masuk utama, jika pasien belum mendapatkan nomor antrian melalui *website* atau SMS, maka pasien mengambil nomor antrian dari kiosk antrian. Setelah itu pasien dapat menunggu di ruang antri yang telah di sediakan. Di sebelah kiri ruangan terdapat 6 counter yang akan mengarahkan pasien ke klinik yang akan dituju. Di sebelah kanan ruangan terdapat ruang informasi dan counter farmasi untuk pengambilan obat beserta ruang tunggu obat. Dari pintu masuk utama lurus ke depan terdapat ruang rawat inap dan ruang tunggu keluarga pasien rawat inap.



Gambar 4.1 Denah RSUD Dr. Soetrasno Rembang

Sumber: CV Pabrik Teknologi

Fitur-fitur yang ada pada Sistem Antrian RSUD Dr. Soetrasno Rembang berdasarkan hasil wawancara kepada *Project Manager* yaitu Ambil Nomor Antrian, Cek Antrian, SMS Gateway, Monitor, Data User, Data Admin, Lihat Daftar Antrian Saat Ini, Panggil Ulang Pasien dan Panggil Pasien Selanjutnya.



Kebutuhan-kebutuhan fungsional pada Sistem Antrian RSUD Dr. Soetrasno berdasarkan wawancara kepada *Project Manager* akan disajikan pada Tabel 4.1, 4.2, 4.3 dan 4.4.

1. Pengguna: Pasien

Tabel 4.1 Pengguna Pasien

	Nama Fungsi	Deskripsi
1	Melakukan Registrasi	Pasien melakukan registrasi agar terdaftar pada sistem dengan memasukkan nomor HP, sistem mengirimkan password ke nomor HP pasien.
2	Ambil Nomor Antrian	Pasien dapat mengambil nomor antrian melalui sistem.
3	Cek Antrian	Pasien dapat melihat nomor antrian pasien dan nomor antrian yang dipanggil saat ini.

Sumber: CV Pabrik Teknologi (2017)

2. Pengguna: Admin

Tabel 4.2 Pengguna Admin

	Nama Fungsi	Deskripsi
1	Baca Pesan Masuk	Admin dapat membaca pesan masuk.
2	Hapus Pesan Masuk	Admin dapat menghapus pesan masuk.
3	Baca Pesan Keluar	Admin dapat membaca pesan keluar.
4	Hapus Pesan Keluar	Admin dapat menghapus pesan keluar.
5	Ubah Kuota SMS	Admin dapat memperbarui jumlah kuota SMS.
6	Tambah Video	Admin menambah video yang tampil pada monitor.
7	Hapus Video	Admin dapat menghapus video yang tampil pada monitor.
8	Tambah Running Text	Admin menambah running text yang tampil pada monitor.
9	Hapus Running Text	Admin dapat menghapus running text yang tampil pada monitor.
10	Lihat Data <i>User</i>	Admin dapat melihat seluruh data <i>user</i> yang sudah terdaftar dalam sistem.
11	Tambah Data <i>User</i>	Admin dapat membuat <i>user</i> baru dengan memasukkan data nomor telepon, nomor aktivasi dan password.
12	Ubah Data <i>User</i>	Admin dapat melakukan ubah data <i>user</i> .
13	Hapus Data <i>User</i>	Admin dapat menghapus data <i>user</i> .
14	Lihat Data Admin	Admin dapat melihat seluruh data admin yang sudah terdaftar dalam sistem.
15	Tambah Data Admin	Admin dapat membuat admin baru dengan memasukkan data nama, jabatan, status pelayanan, password, counter.
16	Hapus Data Admin	Admin dapat menghapus data admin.
17	Ubah Data Admin	Admin dapat mengubah data admin

Sumber: CV Pabrik Teknologi (2017)



3. Pengguna: Admin Counter

Tabel 4.3 Pengguna Admin Counter

	Nama Fungsi	Deskripsi
1	Lihat Daftar Antrian	Admin Counter dapat melihat daftar antrian saat ini.
2	Panggil Ulang Pasien	Admin Counter melakukan pemanggilan ulang nomor urut pasien.
3	Panggil Pasien Selanjutnya	Admin Counter melakukan pemanggilan pasien dengan nomor antrian selanjutnya.

Sumber: CV Pabrik Teknologi (2017)

4. Pengguna: Admin, Admin Counter, Pasien

Tabel 4.4 Pengguna Admin, Admin Counter, Pasien

	Nama Fungsi	Deskripsi
1	<i>Login</i>	Agar dapat mengakses website, Admin, Admin Counter memasukkan <i>username</i> dan <i>password</i> . Sedangkan pasien memasukkan nomor HP dan <i>password</i> .

Sumber: CV Pabrik Teknologi (2017)

4.3 Mekanisme Lembar Penilaian

Pada penelitian ini tujuan pengambilan data melalui lembar penilaian adalah untuk mengumpulkan data kuantitatif guna menghitung estimasi biaya proyek Sistem Antrian RSUD Dr. Soetrasno Rembang menggunakan metode *Use Case Point* dan metode *Function Point*. Data kuantitatif yang didapat berupa bobot nilai dari pengisian lembar penilaian yang berisi faktor-faktor yang berhubungan dengan teknis dan lingkungan dari pengembangan perangkat lunak.

4.4 Hasil Lembar Penilaian

Berdasarkan mekanisme lembar penilaian tersebut, pada metode *Use Case Point* didapat hasil berupa perhitungan nilai *Technical Complexity Factor* (TCF) dan nilai *Environment Technical Factor* (ECF) yang akan dimasukkan ke dalam rumus sehingga menghasilkan nilai *Complexity Factor*. Untuk perhitungan *Technical Complexity Factor* (TCF) meliputi 13 faktor yaitu *Distributed System Required*, *Response Time*, *End User Efficiency*, *Complex Internal Processing Required*, *Reusable Code*, *Easy to Install*, *Easy to Use*, *Portable*, *Easy to Change*, *Concurrent*, *Security Features*, *Access for Third Parties* dan *Special Training Required*. Kemudian untuk perhitungan *Environment Complexity Factor* (ECF) meliputi 8 faktor yaitu *Familiarity With The Project*, *Application Experience*, *OO Programming Experience*, *Lead Analyst Capability*, *Motivation*, *Stable Requirements*, *Part Time Staff* dan *Difficult Programming Language*. Model jawaban pada lembar penilaian tersebut yaitu dalam bentuk skala yaitu 0 (*No Influence*), 1 (*Incidental Influence*), 2 (*Moderate Influence*), 3 (*Average Influence*), 4 (*Significant Influence*) dan 5 (*Strong Influence*).

Sedangkan pada metode *Function Point* didapat hasil perhitungan nilai *Technical Complexity Factor* (TCF) yang meliputi 14 faktor yaitu: *Data Communications*, *Distributed Data Processing*, *Performance*, *Heavily Used Configuration*, *Transaction Rate*, *Online Data Entry*, *End-User Efficiency*, *Online Update*, *Complex Processing*, *Reusability*, *Installation Ease*, *Operation Ease*, *Multiple Sites* dan *Facilitate Change*. Model jawaban pada lembar penilaian tersebut yaitu dalam bentuk skala yaitu 0 sampai 5.



4.5 User Story

Rumah Sakit Umum Daerah dr. R. Soetrasno Rembang merupakan rumah sakit yang memiliki pelayanan medis seperti pelayanan kedokteran, medis khusus, rujukan kesehatan, rawat jalan dan rawat darurat maupun rawat tinggal.

Untuk memudahkan para admin counter dalam menangani antrian pasien, maka pihak RSUD Dr. Soetrasno Rembang membutuhkan sistem antrian berbasis web yang dapat dikelola langsung oleh pihak rumah sakit serta memudahkan pasien untuk mengambil nomor antrian. Oleh karena itu, pihak rumah sakit meminta CV Pabrik Teknologi untuk membangun sebuah aplikasi berbasis web yang dapat memudahkan pengelolaan antrian pasien. CV Pabrik Teknologi merupakan sebuah perusahaan berbasis jasa yang berfokus pada bidang jasa teknologi informasi. CV Pabrik Teknologi merupakan salah satu perusahaan rintisan atau *startup* yang ada di kota Malang. Jenis layanan yang dihasilkan oleh CV Pabrik Teknologi, antara lain: pengembangan *website* tiket pesawat *online* (*widget*, pemesanan, keagenan), pengembangan *website* profil baik pribadi, bisnis, institusi ataupun korporasi, pengembangan *website* aplikasi untuk institusi, korporasi maupun bisnis, pengembangan *software* administrasi, pengembangan aplikasi *mobile* untuk berbagai keperluan dan instalasi jaringan pada perkantoran.

Aplikasi yang dikembangkan berdasarkan proses bisnis yang ada di RSUD Dr. Soetrasno Rembang. Pasien datang melalui pintu masuk utama, jika pasien belum mendapatkan nomor antrian melalui website atau SMS, maka pasien mengambil nomor antrian dari kios antrian. Setelah itu pasien dapat menunggu di ruang antri yang telah di sediakan. Di sebelah kiri ruangan terdapat 6 counter yang akan mengarahkan pasien ke poli yang akan dituju. Jika monitor menampilkan nomor antrian yang dipanggil, maka pasien menuju ke konter yang ada di ruangan. Setelah itu pihak konter akan mengarahkan ke poli yang dituju pasien. Di sebelah kanan ruangan terdapat ruang informasi dan counter farmasi untuk pengambilan obat beserta ruang tunggu obat. Dari pintu masuk utama lurus ke depan terdapat ruang rawat inap dan ruang tunggu keluarga pasien rawat inap.

Admin dapat mengelola data admin (admin dan admin counter) user (pasien) dengan memasukkan data ke dalam sistem seperti ID_admin, Nama, Password, Jabatan, Counter, ID_antrian_counter. Sedangkan untuk data *user* admin memasukkan data seperti ID_user, No_telepon, No_aktivasi, Password. Admin juga dapat mengelola SMS Gateway meliputi data pesan masuk dan data pesan keluar serta dapat memperbarui jumlah kuota SMS melalui web. Admin dapat membaca dan menghapus pesan masuk dan pesan keluar. Pesan masuk yang diterima yaitu pesan yang dikirim oleh pasien untuk mendapatkan nomor antrian. Sedangkan pesan keluar berisi password yang dikirimkan ke nomor pasien. Namun jika pasien belum mengambil nomor antrian maka jika pasien mengecek nomor sistem hanya akan menampilkan nomor antrian yang sedang dipanggil saja. Data pesan masuk disimpan dalam database yang meliputi ID_pesan, TextDecode dan SenderNumber. Sedangkan data pesan keluar meliputi DestinationNumber, ID_pesan, TextDecode.

Pasien yang ingin mengambil nomor antrian melalui web, diharuskan melakukan registrasi dengan memasukkan nomor HP. Lalu sistem mengirimkan *password* ke nomor HP pasien sehingga pasien bisa login ke dalam web. Saat melakukan login, pasien memasukkan nomor HP dan password. Setelah berhasil login, pasien dapat mengambil nomor antrian dan mengecek nomor antrian yang dipanggil saat ini. Jika pasien berhasil login ke dalam website, maka pasien dapat mengambil nomor antrian serta mengecek nomor antrian pasien dan nomor antrian yang dipanggil saat ini.



Jika admin counter berhasil login ke dalam website dengan memasukkan *username* dan *password*, maka sistem akan menampilkan data antrian. Admin counter akan memanggil ulang pasien jika pasien tidak datang ke counter setelah nomor antrian dipanggil, kemudian jika pasien tidak hadir maka admin counter memanggil nomor antrian selanjutnya. Admin counter juga harus memasukkan status pelayanan mengenai pasien apakah pasien tidak hadir atau sudah dilayani.

Di ruang antri pasien dapat melihat monitor yang akan menampilkan video, *running text* dan data antrian. Untuk mengelola monitor tersebut admin dapat menambah dan menghapus data video dan data *running text* yang akan ditampilkan di monitor. Admin dapat mengelola monitor tersebut melalui *website*. Data yang akan disimpan pada *database* Video yaitu *ID_Video* sedangkan pada *database* Kalimat yaitu *ID_Kalimat*.

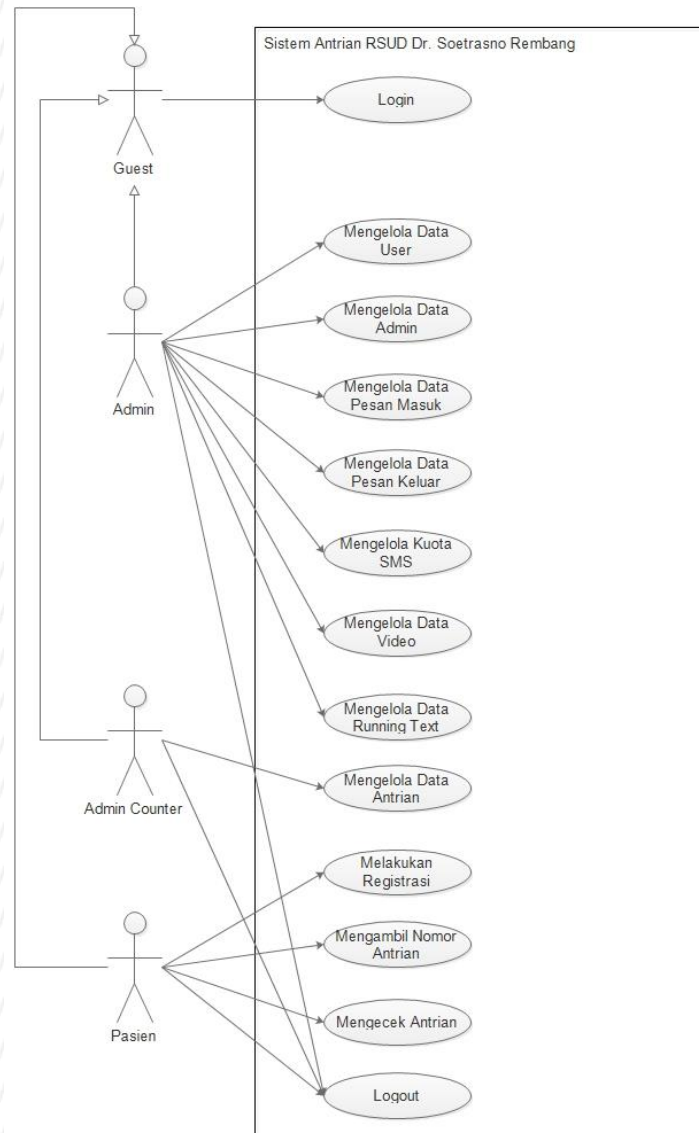
4.6 Pembuatan *Use Case Diagram*

Peneliti membuat *Use Case Diagram* berdasarkan *User Story* yang telah divalidasi oleh tim pengembang dari CV Pabrik Teknologi dan kemudian melakukan validasi *Use Case Diagram*. Diagram ini dibutuhkan dalam penelitian ini untuk menghitung estimasi proyek Sistem Antrian RSUD Dr. Soetrasno Rembang menggunakan metode *Use Case Point*.

Setiap *use case* yang didefinisikan akan dijelaskan menggunakan *Use Case Scenario* dalam bentuk tabel yang berisi nama dan tujuan *use case*, aktor yang terkait dengan *use case* tersebut, skenario utama dan alternatif. *Use Case Diagram* yang telah divalidasi disajikan pada Gambar 4.2.



4.6.1 Use Case Diagram



Gambar 4.2 Use Case Diagram Sistem Antrian RSUD Dr. Soetrasno Rembang

Keterangan Gambar 4.2:

Untuk menjadi *user* yang dapat mengakses Sistem Antrian RSUD Dr. Soetrasno Rembang, *guest* harus login terlebih dahulu. Setelah berhasil login, *user* dapat berinteraksi dengan sistem. *User* admin dapat mengelola data *user* meliputi tambah, ubah dan hapus data *user*, data pesan masuk dan pesan keluar meliputi tambah dan hapus data, jumlah kuota SMS, data video dan data running text pada Sistem Antrian RSUD Dr. Soetrasno Rembang. Pengguna operator counter dapat mengelola data antrian pada Sistem Dr. Soetrasno Rembang. Pengguna pasien dapat mengambil nomor antrian dan cek antrian untuk melihat nomor antrian pasien dan nomor antrian yang dipanggil saat ini melalui Sistem Antrian RSUD Dr. Soetrasno Rembang. Untuk data video, running text dan data antrian akan ditampilkan pada monitor.



4.6.2 Use Case Scenario

Use Case Scenario merupakan perwujudan dari *use case* yang merepresentasikan *main flow* dan *alternative flow* dari sebuah *use case*. Tujuan dibuatnya *Use Case Scenario* untuk menentukan jumlah transaksi setiap *use case* berdasarkan alur sukses dan alur alternatif. Berikut merupakan skenario dari setiap *use case* pada Gambar 4.2 mengenai Sistem Antrian RSUD Dr. Soetrasno Rembang yang disajikan pada Tabel 4.5 sampai Tabel 4.25.

Tabel 0.5 Skenario Use Case Login

Flow of Events untuk Use Case Login	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana <i>guest</i> masuk ke dalam Sistem Antrian RSUD Dr. Soetrasno Rembang
Actor	<i>Guest</i>
Pre-Condition	Tidak ada
Basic Flow of Events	<ol style="list-style-type: none"> 1. Sistem menampilkan menu login untuk masuk ke dalam sistem 2. <i>Guest</i> memasukkan <i>username</i> dan <i>password</i> 3. Sistem melakukan autentikasi bagi <i>guest</i> yang akan masuk ke dalam sistem 4. Sistem menampilkan menu beranda jika <i>username</i> dan <i>password</i> valid dan terautentikasi oleh sistem {use case selesai}
Alternative Flows	<p>a. {username dan password} tidak valid</p> <p>Jika di <i>basic flow</i>, <i>guest</i> memasukkan <i>username</i> dan <i>password</i> tidak valid, sistem menampilkan pesan kesalahan. <i>Guest</i> bisa memilih untuk kembali ke awal basic flow atau membatalkan login, dan <i>use case</i> selesai</p>
Subflow	Tidak ada
Post-Conditions	Sistem menampilkan informasi login berhasil

Tabel 0.6 Skenario Use Case Lihat Data User

Flow of Events untuk Use Case Lihat Data User	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin melihat data <i>user</i>
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> 1. Admin memilih menu Data <i>User</i> lalu sistem menampilkan data <i>user</i> dan operasi detail, tambah, ubah dan hapus 2. Admin memilih operasi detail lalu sistem menampilkan detail data <i>user</i> {use case selesai}
Alternative Flows	Tidak Ada



Subflow	Tidak Ada
Post-Conditions	Admin berhasil melihat data <i>user</i>

Tabel 0.7 Skenario *Use Case* Tambah Data *User*

Flow of Events untuk Use Case Tambah Data User	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin tambah data <i>user</i>
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin memilih menu Data <i>User</i> lalu sistem menampilkan data <i>user</i> dan operasi detail, tambah, ubah dan hapus Admin memilih operasi tambah lalu sistem menampilkan menu untuk menambah data <i>user</i> Admin memasukkan data <i>user</i> {mengisi data user} Sistem menampilkan informasi bahwa data berhasil ditambahkan {use case selesai}
Alternative Flows	<p>a. Data yang dimasukkan tidak lengkap</p> <p>Pada {mengisi data user} di <i>basic flow</i>, apabila admin tidak mengisi salah satu data yang harus terisi maka sistem akan menampilkan informasi bahwa terdapat data yang belum terisi.</p>
Subflow	Tidak Ada
Post-Conditions	Admin berhasil tambah data <i>user</i>

Tabel 0.8 Skenario *Use Case* Ubah Data *User*

Flow of Events untuk Use Case Ubah Data User	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin mengubah data <i>user</i>
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin memilih menu Data <i>User</i> lalu sistem menampilkan data <i>user</i> dan operasi detail, tambah, ubah dan hapus Admin memilih operasi ubah lalu sistem menampilkan data <i>user</i> yang akan dilakukan perubahan Admin mengubah data <i>user</i> pada sistem dan menyimpan hasil perubahan {mengisi data user} Sistem menampilkan informasi bahwa data <i>user</i> berhasil diubah {use case selesai}
Alternative Flows	<p>a. Data yang dimasukkan tidak lengkap</p>



Tabel 0.8 Skenario Use Case Ubah Data User

	Pada {mengisi data user} di <i>basic flow</i> , apabila admin tidak mengisi salah satu data yang harus terisi maka sistem akan menampilkan informasi bahwa terdapat data yang belum terisi.
Subflow	Tidak Ada
Post-Conditions	Admin berhasil mengubah data user

Tabel 0.9 Skenario Use Case Hapus Data User

Flow of Events untuk Use Case Hapus Data User	
Brief Description	Use case ini menjelaskan bagaimana admin menghapus data user
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> 1. Admin memilih menu Data User lalu sistem menampilkan data user dan operasi detail, tambah, ubah dan hapus 2. Admin memilih operasi hapus lalu sistem melakukan operasi dan verifikasi penghapusan data user 3. Sistem menampilkan informasi bahwa data user berhasil dihapus {use case selesai}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin berhasil menghapus data user

Tabel 0.10 Skenario Use Case Baca Pesan Masuk

Flow of Events untuk Use Case Baca Pesan Masuk	
Brief Description	Use case ini menjelaskan bagaimana admin membaca pesan masuk
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> 1. Admin memilih sub menu pesan masuk dan sistem menampilkan data pesan masuk dan operasi detail dan hapus 2. Admin memilih detail pada salah satu pesan masuk lalu sistem menampilkan detail pesan masuk {use case selesai}
Alternative Flows	Tidak Ada

Tabel 0.10 Skenario Use Case Baca Pesan Masuk

Subflow	Tidak Ada
Post-Conditions	Admin berhasil baca data pesan masuk

Tabel 0.11 Skenario Use Case Hapus Pesan Masuk

Flow of Events untuk Use Case Hapus Pesan Masuk	
Brief Description	Use case ini menjelaskan bagaimana admin menghapus pesan masuk
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin memilih sub menu pesan masuk dan sistem menampilkan data pesan masuk dan operasi detail dan hapus Admin memilih operasi hapus lalu sistem melakukan operasi dan verifikasi penghapusan data pesan masuk Sistem menampilkan informasi bahwa data pesan masuk berhasil dihapus {use case selesai}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin berhasil menghapus data pesan masuk

Tabel 0.12 Skenario Use Case Baca Pesan Keluar

Flow of Events untuk Use Case Baca Pesan Keluar	
Brief Description	Use case ini menjelaskan bagaimana admin membaca pesan keluar
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin memilih sub menu pesan keluar dan sistem menampilkan data pesan keluar dan operasi detail dan hapus Admin memilih detail pada salah satu pesan keluar lalu sistem menampilkan detail pesan keluar {use case selesai}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin berhasil baca data pesan keluar

Tabel 0.13 Skenario *Use Case* Hapus Pesan Keluar

Flow of Events untuk Use Case Hapus Pesan Keluar	
Brief Description	Use case ini menjelaskan bagaimana admin menghapus pesan keluar
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin memilih sub menu pesan keluar dan sistem menampilkan data pesan keluar dan operasi detail dan hapus Admin memilih operasi hapus lalu sistem melakukan operasi dan verifikasi penghapusan data pesan keluar Sistem menampilkan informasi bahwa data pesan keluar berhasil dihapus {use case selesai}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin berhasil menghapus data pesan keluar

Tabel 0.14 Skenario *Use Case* Lihat Kuota SMS

Flow of Events untuk Use Case Lihat Kuota SMS	
Brief Description	Use case ini menjelaskan bagaimana admin melihat jumlah kuota SMS
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin memilih sub menu kuota SMS lalu sistem menampilkan jumlah kuota SMS dan tombol ubah {use case selesai}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin berhasil melihat jumlah kuota SMS

Tabel 0.15 Skenario *Use Case* Ubah Kuota SMS

Flow of Events untuk Use Case Lihat Kuota SMS	
Brief Description	Use case ini menjelaskan bagaimana admin mengubah jumlah kuota SMS
Actor	Admin

Tabel 0.14 Skenario *Use Case* Lihat Kuota SMS

Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin memilih sub menu Kuota SMS lalu sistem menampilkan jumlah kuota SMS dan operasi ubah Admin memilih operasi ubah lalu admin mengubah jumlah kuota SMS pada sistem dan menyimpan hasil perubahan {mengisi jumlah kuota SMS} Sistem menampilkan informasi bahwa jumlah kuota SMS berhasil diperbarui {use case selesai}
Alternative Flows	<p>a. Data yang dimasukkan tidak sesuai dengan format</p> <p>Pada {mengisi jumlah kuota SMS} pada <i>sub flow</i>, apabila admin mengisi selain format angka maka sistem akan menampilkan informasi bahwa data yang dimasukkan salah</p>
Subflow	Tidak Ada
Post-Conditions	Admin berhasil mengubah jumlah kuota SMS

Tabel 0.16 Skenario *Use Case* Lihat Data Antrian

Flow of Events untuk Use Case Lihat Data Antrian	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin counter melihat data antrian
Actor	Admin Counter
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin Counter memilih menu Data Antrian lalu sistem menampilkan data antrian dan operasi panggil ulang pasien dan panggil pasien selanjutnya {use case selesai}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin Counter berhasil melihat data antrian

Tabel 0.17 Skenario *Use Case* Panggil Ulang Pasien

Flow of Events untuk Use Case Panggil Ulang Pasien	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin counter melakukan operasi pemanggilan ulang pasien
Actor	Admin Counter
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin Counter memilih menu Data Antrian lalu sistem

Tabel 0.16 Skenario *Use Case* Lihat Data Antrian

	<p>menampilkan data antrian dan operasi panggil ulang pasien dan panggil pasien selanjutnya</p> <ol style="list-style-type: none"> Admin memilih operasi panggil ulang pasien lalu sistem melakukan pemanggilan ulang pasien Admin memasukkan status pelayanan Sistem menampilkan informasi pemanggilan ulang berhasil {use case selesai}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin Counter berhasil melakukan operasi pemanggilan ulang pasien

Tabel 0.18 Skenario *Use Case* Panggil Pasien Selanjutnya

Flow of Events untuk Use Case Panggil Pasien Selanjutnya	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin counter melakukan operasi panggil pasien selanjutnya
Actor	Admin Counter
Pre-Condition	Sistem menampilkan data antrian
Basic Flow of Events	<ol style="list-style-type: none"> Admin Counter memilih menu Data Antrian lalu sistem menampilkan data antrian dan operasi panggil ulang pasien dan panggil pasien selanjutnya Admin memilih operasi panggil pasien selanjutnya lalu sistem melakukan pemanggilan pasien selanjutnya Sistem menampilkan informasi pemanggilan ulang berhasil {use case selesai}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin Counter berhasil melakukan panggil pasien selanjutnya

Tabel 0.19 Skenario *Use Case* Tambah Data Video

Flow of Events untuk Use Case Tambah Data Video	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin tambah data video yang akan ditampilkan pada monitor
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin memilih menu Monitor lalu sistem menampilkan data video dan operasi tambah dan hapus

Tabel 0.19 Skenario *Use Case* Tambah Data Video

	<ol style="list-style-type: none"> 2. Admin memilih operasi tambah lalu sistem akan menampilkan menu untuk mengunggah <i>file</i> video 3. Sistem akan menampilkan informasi bahwa data video berhasil ditambahkan {<i>use case selesai</i>}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin berhasil tambah data video

Tabel 0.20 Skenario *Use Case* Hapus Data Video

Flow of Events untuk <i>Use Case</i> Hapus Data Video	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin menghapus data video
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> 1. Admin memilih menu Monitor lalu sistem menampilkan data video dan operasi tambah dan hapus 2. Admin memilih operasi hapus lalu sistem melakukan operasi hapus data video 3. Sistem menampilkan informasi bahwa data video berhasil dihapus {<i>use case selesai</i>}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin berhasil menghapus data video

Tabel 0.21 Skenario *Use Case* Tambah Data *Running Text*

Flow of Events untuk <i>Use Case</i> Tambah Data <i>Running Text</i>	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin tambah data <i>running text</i> yang akan ditampilkan pada monitor
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> 1. Admin memilih menu Monitor lalu sistem menampilkan data <i>running text</i> dan operasi tambah dan hapus 2. Admin memilih operasi tambah lalu sistem akan menampilkan menu untuk mengunggah <i>file running text</i> 3. Sistem akan menampilkan informasi bahwa data <i>running</i>

Tabel 0.20 Skenario *Use Case* Hapus Data Video

	text berhasil ditambahkan { <i>use case</i> selesai}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin berhasil tambah data <i>running text</i>

Tabel 0.22 Skenario *Use Case* Hapus Data *Running Text*

Flow of Events untuk <i>Use Case</i> Hapus Data <i>Running Text</i>	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin menghapus data <i>running text</i>
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> 1. Admin memilih menu Monitor lalu sistem menampilkan data <i>running text</i> dan operasi tambah dan hapus 2. Admin memilih operasi hapus lalu sistem melakukan operasi hapus data <i>running text</i> 3. Sistem menampilkan informasi bahwa data <i>running text</i> berhasil dihapus {<i>use case</i> selesai}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin berhasil menghapus data <i>running text</i>

Tabel 0.23 Skenario *Use Case* Mengambil Nomor Antrian

Flow of Events untuk <i>Use Case</i> Mengambil Nomor Antrian	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana pasien mengambil nomor antrian melalui Sistem Antrian RSUD Dr. Soetrasno Rembang
Actor	Pasien
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> 1. Pasien memilih memilih operasi antri kemudian sistem menampilkan informasi pengambilan nomor antrian berhasil {<i>use case</i> selesai}
Alternative Flows	Tidak ada
Subflow	Tidak ada

Tabel 0.23 Skenario *Use Case* Mengambil Nomor Antrian

Post-Conditions	Pasien berhasil melakukan proses mengambil nomor antrian
------------------------	--

Tabel 0.24 Skenario *Use Case* Mengecek Nomor Antrian

Flow of Events untuk Use Case Mengecek Nomor Antrian	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana pasien mengecek nomor antrian pasien dan nomor antrian yang dipanggil saat ini melalui Sistem Antrian RSUD Dr. Soetrasno Rembang
Actor	Pasien
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> 1. Pasien memilih operasi cek antrian lalu sistem menampilkan nomor antrian pasien dan antrian yang dipanggil saat ini {use case selesai}
Alternative Flows	<p>a. Nomor antrian pasien gagal ditampilkan</p> <p>Jika pasien belum mengambil nomor antrian maka sistem hanya akan menampilkan nomor antrian yang dipanggil saat ini</p>
Subflow	Tidak ada
Post-Conditions	Pasien berhasil melakukan proses mengecek nomor antrian

Tabel 0.25 Skenario *Use Case* Melakukan Registrasi

Flow of Events untuk Use Case Melakukan Registrasi	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana Pasien melakukan registrasi melalui <i>website</i>
Actor	Pasien
Pre-Condition	Tidak ada
Basic Flow of Events	<ol style="list-style-type: none"> 1. Sistem menampilkan menu registrasi untuk masuk ke dalam sistem 2. Pasien memasukkan nomor {mengisi nomor HP} 3. Sistem secara otomatis mengirimkan <i>password</i> ke nomor HP pasien 4. Sistem menampilkan informasi bahwa <i>password</i> dikirim ke nomor HP pasien {use case selesai}
Alternative Flows	<p>a. Pasien tidak menerima <i>password</i></p> <p>Pada {mengisi nomor HP} jika pasien memasukkan nomor yang tidak aktif, maka pasien gagal menerima <i>password</i> sehingga pasien harus memulai <i>basic flow</i> dari awal</p>

Tabel 0.25 Skenario *Use Case* Melakukan Registrasi

Subflow	Tidak Ada
Post-Conditions	Sistem menampilkan informasi registrasi berhasil

Tabel 0.26 Skenario *Use Case* Lihat Data Admin

Flow of Events untuk Use Case Lihat Data Admin	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin melihat data admin
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin memilih menu Data Admin lalu sistem menampilkan data admin dan operasi detail, tambah, ubah dan hapus Admin memilih operasi detail lalu sistem menampilkan detail data admin {use case selesai}
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin berhasil melihat data admin

Tabel 0.27 Skenario *Use Case* Tambah Data Admin

Flow of Events untuk Use Case Tambah Data Admin	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin tambah data admin
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> Admin memilih menu Data Admin lalu sistem menampilkan data admin dan operasi detail, tambah, ubah dan hapus Admin memilih operasi tambah lalu sistem menampilkan menu untuk menambah data admin Admin memasukkan data admin {mengisi data admin} Sistem menampilkan informasi bahwa data berhasil ditambahkan {use case selesai}
Alternative Flows	<p>a. Data yang dimasukkan tidak lengkap</p> <p>Pada {mengisi data admin} di <i>basic flow</i>, apabila admin tidak mengisi salah satu data yang harus terisi maka sistem akan menampilkan informasi bahwa terdapat data yang belum terisi.</p>
Subflow	Tidak Ada

Tabel 0.27 Skenario *Use Case* Tambah Data Admin

Post-Conditions	Admin berhasil tambah data admin
------------------------	----------------------------------

Tabel 0.28 Skenario *Use Case* Ubah Data Admin

Flow of Events untuk Use Case Ubah Data Admin	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin mengubah data admin
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> 1. Admin memilih menu Data Admin lalu sistem menampilkan data admin dan operasi detail, tambah, ubah dan hapus 2. Admin memilih operasi ubah lalu sistem menampilkan data admin yang akan dilakukan perubahan 3. Admin mengubah data admin pada sistem dan menyimpan hasil perubahan {mengisi data admin} 4. Sistem menampilkan informasi bahwa data admin berhasil diubah {use case selesai}
Alternative Flows	<p>a. Data yang dimasukkan tidak lengkap</p> <p>Pada {mengisi data admin} di <i>basic flow</i>, apabila admin tidak mengisi salah satu data yang harus terisi maka sistem akan menampilkan informasi bahwa terdapat data yang belum terisi.</p>
Subflow	Tidak Ada
Post-Conditions	Admin berhasil mengubah data admin

Tabel 0.29 Skenario *Use Case* Hapus Data Admin

Flow of Events untuk Use Case Hapus Data Admin	
Brief Description	<i>Use case</i> ini menjelaskan bagaimana admin menghapus data admin
Actor	Admin
Pre-Condition	Sistem menampilkan menu beranda
Basic Flow of Events	<ol style="list-style-type: none"> 1. Admin memilih menu Data Admin lalu sistem menampilkan data admin dan operasi detail, tambah, ubah dan hapus 2. Admin memilih operasi hapus lalu sistem melakukan operasi dan verifikasi penghapusan data admin 3. Sistem menampilkan informasi bahwa data admin berhasil dihapus {use case selesai}

Tabel 0.29 Skenario *Use Case* Hapus Data Admin

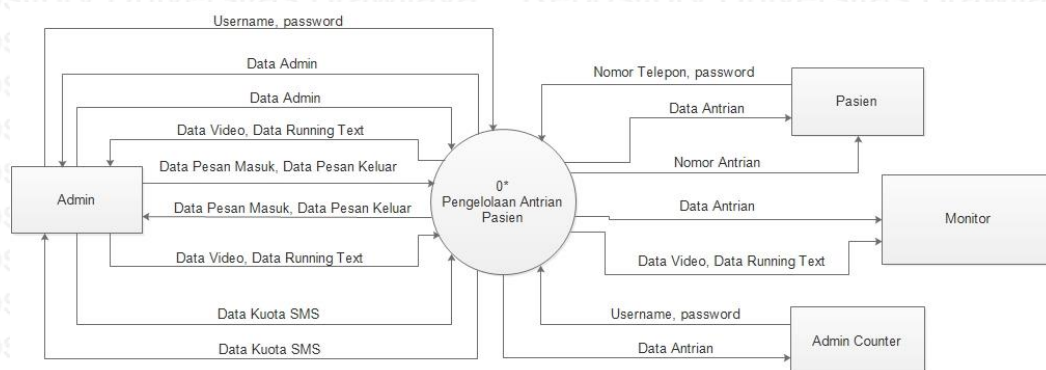
Alternative Flows	Tidak Ada
Subflow	Tidak Ada
Post-Conditions	Admin berhasil menghapus data admin

4.7 Pembuatan *Data Flow Diagram*

Dalam tahap ini, peneliti membuat *Data Flow Diagram* berdasarkan *User Story* dari hasil wawancara dan kemudian melakukan validasi *Data Flow Diagram* kepada narasumber dari CV Pabrik Teknologi. Diagram ini dibutuhkan dalam penelitian ini untuk menghitung estimasi proyek Sistem Antrian RSUD Dr. Soetrasno Rembang menggunakan metode *Function Point*. Berikut merupakan *Data Flow Diagram* Sistem Antrian RSUD Dr. Soetrasno Rembang.

4.7.1 *Context Diagram*

Context Diagram merupakan level tertinggi dari suatu *Data Flow Diagram*, yang mempresentasikan seluruh elemen pada perangkat lunak sebagai sebuah lingkaran dengan *input* dan *output* dengan arah panah masuk dan keluar (Pressman, 2001). *Context Diagram* pada Sistem Antrian RSUD Dr. Soetrasno Rembang yang disajikan pada Gambar 4.3.



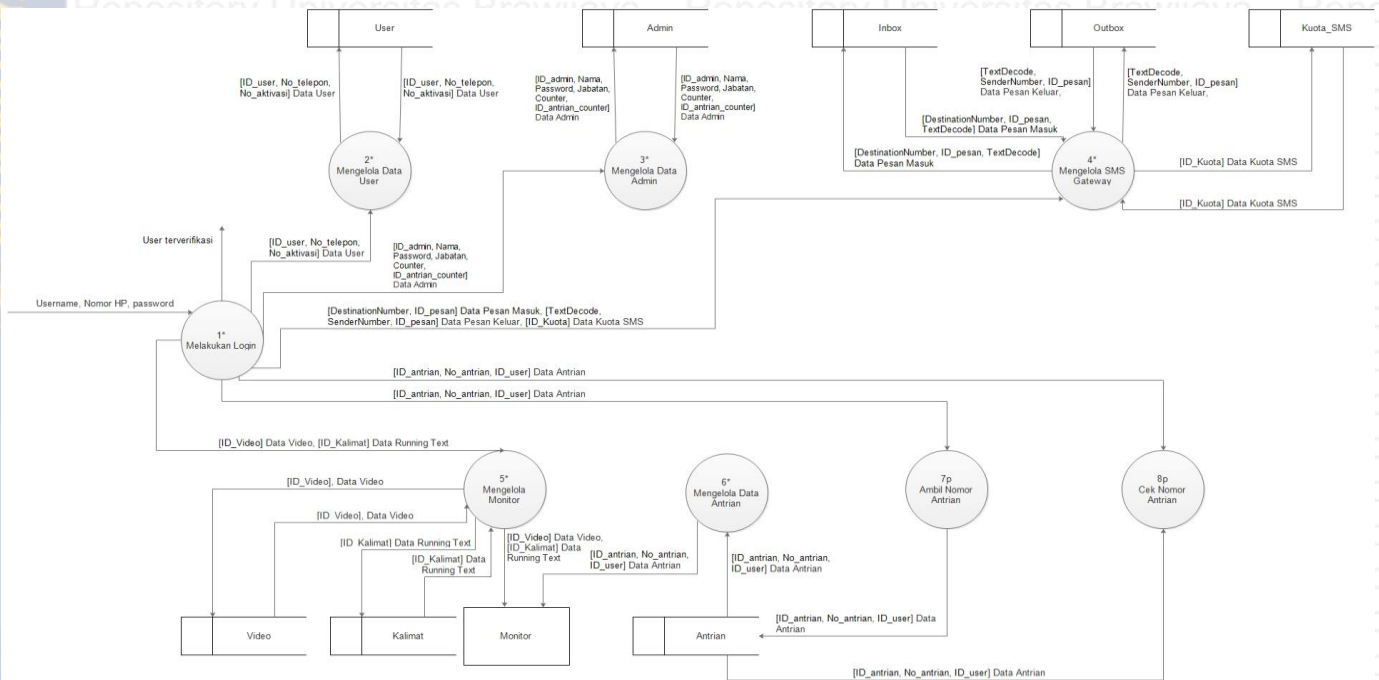
Gambar 4.3 *Context Diagram* Sistem Antrian RSUD Dr. Soetrasno Rembang

Keterangan Gambar 4.3:

User admin dapat mengakses data pesan masuk dan pesan keluar, jumlah kuota SMS, data *user*, data admin serta data video dan running text pada Sistem Antrian RSUD Dr. Soetrasno Rembang. *User* admin counter dapat melihat data antrian pada Sistem Antrian RSUD Dr. Soetrasno Rembang. *User* pasien dapat mengambil nomor antrian pada Sistem Dr. Soetrasno Rembang. Sistem Antrian RSUD Dr. Soetrasno Rembang dapat menampilkan data pesan masuk dan pesan keluar, data *user* serta data video dan running text, data antrian dan nomor antrian. Monitor dapat menampilkan data antrian, data video dan data running text.

4.7.2 Data Flow Diagram Level 1

Data Flow Diagram Level 1 merupakan dekomposisi dari *Context Diagram*, pada Data Flow Diagram Level 1 Sistem Antrian RSUD Dr. Soetrasno Rembang terdiri dari 8 proses yang saling terhubung yang disajikan pada Gambar 4.4.



Gambar 4.4 Data Flow Diagram Level 1 Sistem Antrian RSUD Dr. Soetrasno Rembang

Keterangan Gambar 4.4:

Pada proses Melakukan Login dapat menerima masukan *username*, nomor HP dan *password*. Pada proses Mengelola Data User dapat menerima masukan data user yang dapat menyimpan ke *database user* dan menerima data dari *database user*. Pada proses Mengelola Data Admin dapat dapat menerima masukan data admin yang dapat menyimpan ke *database admin* dan menerima data dari *database admin*.

Pada proses Mengelola SMS Gateway dapat mengelola data pesan masuk yang dapat disimpan dalam *database Inbox* dan data pesan keluar yang disimpan dalam *database Outbox*. Proses ini juga dapat menerima data pesan masuk dari *database Inbox* dan menerima data pesan keluar dari *database Outbox*. Proses ini juga memperbarui jumlah kuota SMS yang akan tersimpan di *database Kuota_SMS*. Pada proses Mengelola Monitor dapat mengelola data video yang akan tersimpan di *database Video* dan mengelola data *running text* akan tersimpan di *database Kalimat* yang akan ditampilkan pada Monitor.

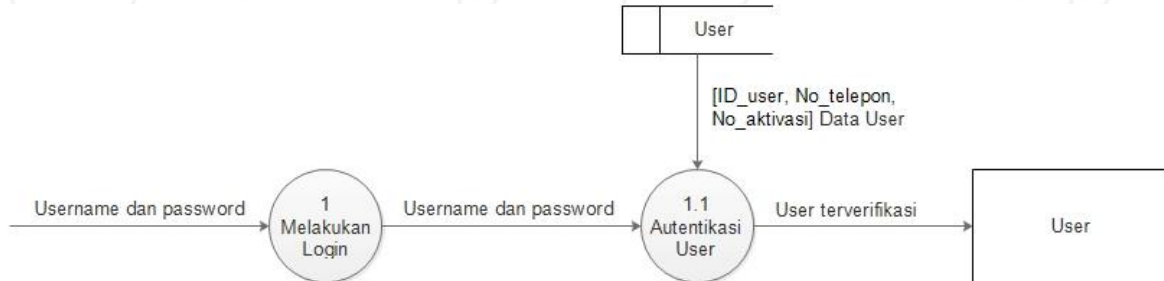
Pada proses Mengelola Data Antrian dapat menerima masukan data antrian yang dapat menyimpan ke *database antrian* dan menerima data dari *database antrian*. Data antrian juga dapat ditampilkan pada Monitor. Pada proses Ambil Nomor Antrian dapat menerima masukan data antrian dan menyimpan data antrian ke *database antrian*. Pada proses Cek Nomor Antrian dapat menerima masukan data antrian dan menampilkan data antrian.



4.7.3 Data Flow Diagram Level 2

Data Flow Diagram Level 2 merupakan dekomposisi dari proses-proses yang terdapat dalam Data Flow Diagram Level 1. Data Flow Diagram Level 2 Sistem Antrian RSUD Dr. Soetrasno Rembang terdiri dari 4 diagram yang disajikan pada Gambar 4.5, 4.6, 4.7 dan 4.8.

4.7.3.1 Data Flow Diagram Level 2 Melakukan Login

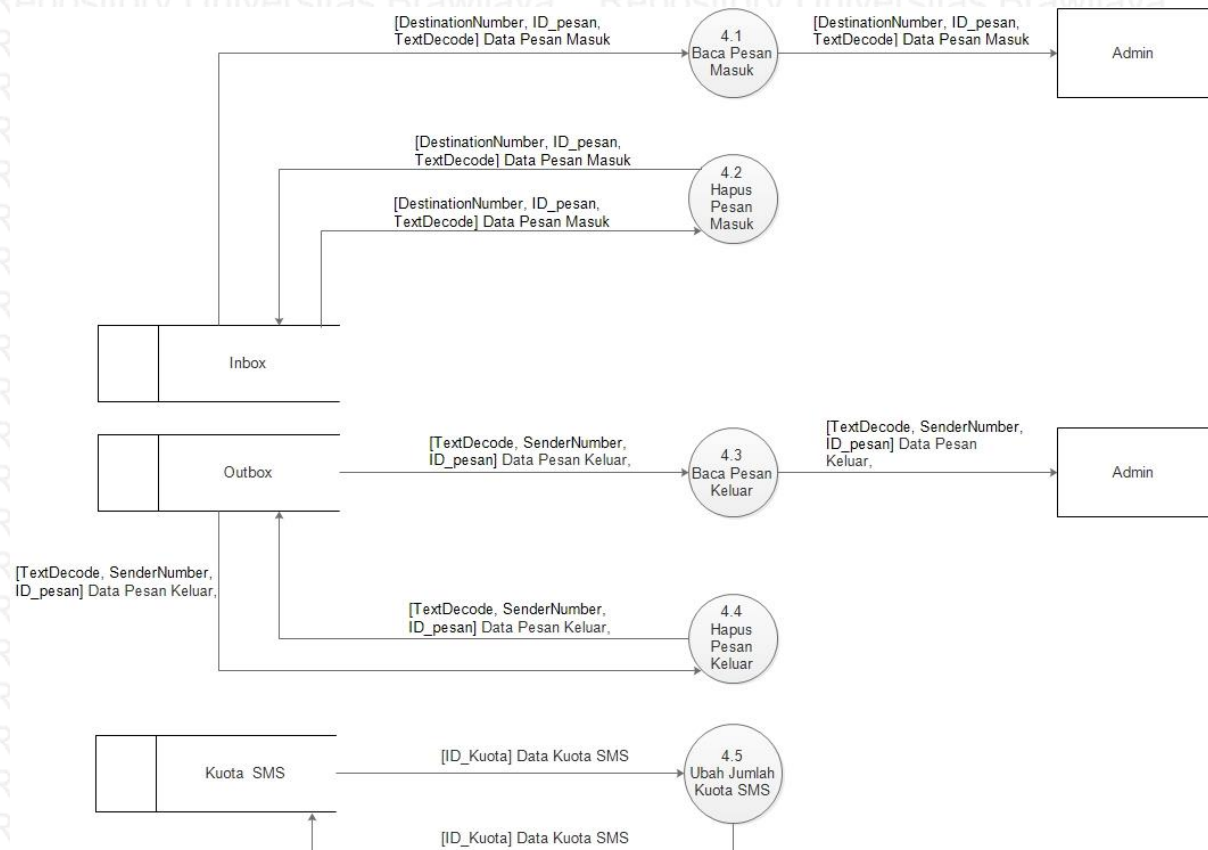


Gambar 4.5 Data Flow Diagram Level 2 Melakukan Login Sistem Antrian RSUD Dr. Soetrasno Rembang

Keterangan Gambar 4.5:

Pada proses melakukan login, *guest* memasukkan *username* dan *password*, kemudian pada proses selanjutnya dilakukan autentikasi user yang membutuhkan masukan dari proses sebelumnya, kemudian jika *guest* berhasil diautentikasi maka *guest* menjadi *user* yang berhasil login ke dalam sistem.

4.7.3.2 Data Flow Diagram Level 2 Mengelola SMS Gateway





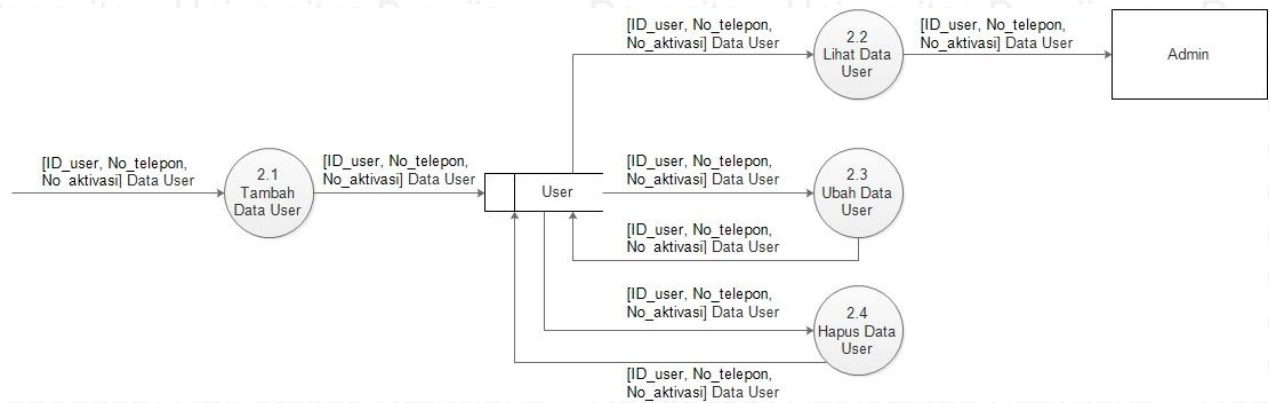
Gambar 4.6 Data Flow Diagram Level 2 Mengelola SMS Gateway Sistem Antrian RSUD Dr. Soetrasno Rembang

Keterangan Gambar 4.6:

Pada proses Baca Pesan Masuk dapat menerima masukan data pesan masuk yang menghasilkan data pesan masuk. Pada proses Hapus Pesan Masuk dapat menerima masukan data pesan masuk yang menghasilkan data pesan masuk serta dapat disimpan dalam *database Inbox*. Pada proses Baca Pesan Keluar dapat menerima masukan data pesan keluar yang menghasilkan data pesan keluar.

Pada proses Hapus Pesan Keluar dapat menerima masukan data pesan keluar yang menghasilkan data pesan keluar serta dapat disimpan dalam *database Outbox*. Pada proses Perbarui Jumlah Kuota SMS dapat memperbarui jumlah kuota SMS yang dapat disimpan pada database Kuota_SMS.

4.7.3.3 Data Flow Diagram Level 2 Mengelola Data User



Gambar 4.7 Data Flow Diagram Level 2 Mengelola Data User Sistem Antrian RSUD Dr. Soetrasno Rembang

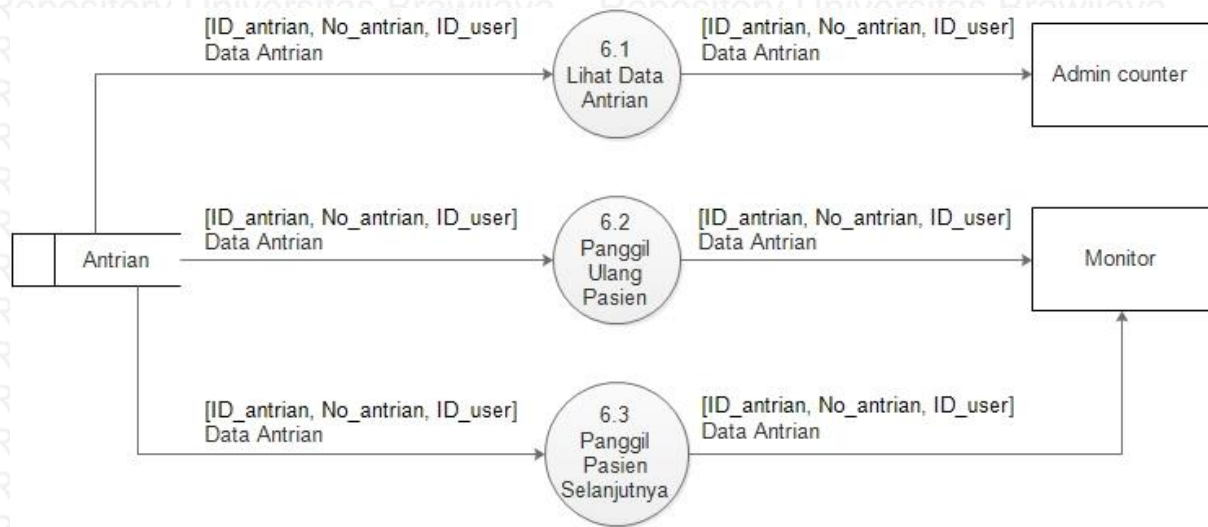
Keterangan Gambar 4.7:

Pada proses Buat Data User dapat menerima masukan data user dan menghasilkan data user yang disimpan dalam *database user*. Pada proses Lihat Data User dapat menerima masukan data user dan menghasilkan data user. Pada proses Ubah Data User dapat menerima masukan data user dan menghasilkan data user dan dapat memperbarui data user yang akan tersimpan dalam *database user*. Pada proses Hapus Data User dapat menerima masukan data user yang menghasilkan data user serta dapat disimpan dalam *database user*.



4.7.3.4 Data Flow Diagram Level 2 Mengelola Data Antrian

Berikut ini merupakan *Data Flow Diagram level 2* dari dekomposisi proses Mengelola Data Antrian yang disajikan pada Gambar 4.8.



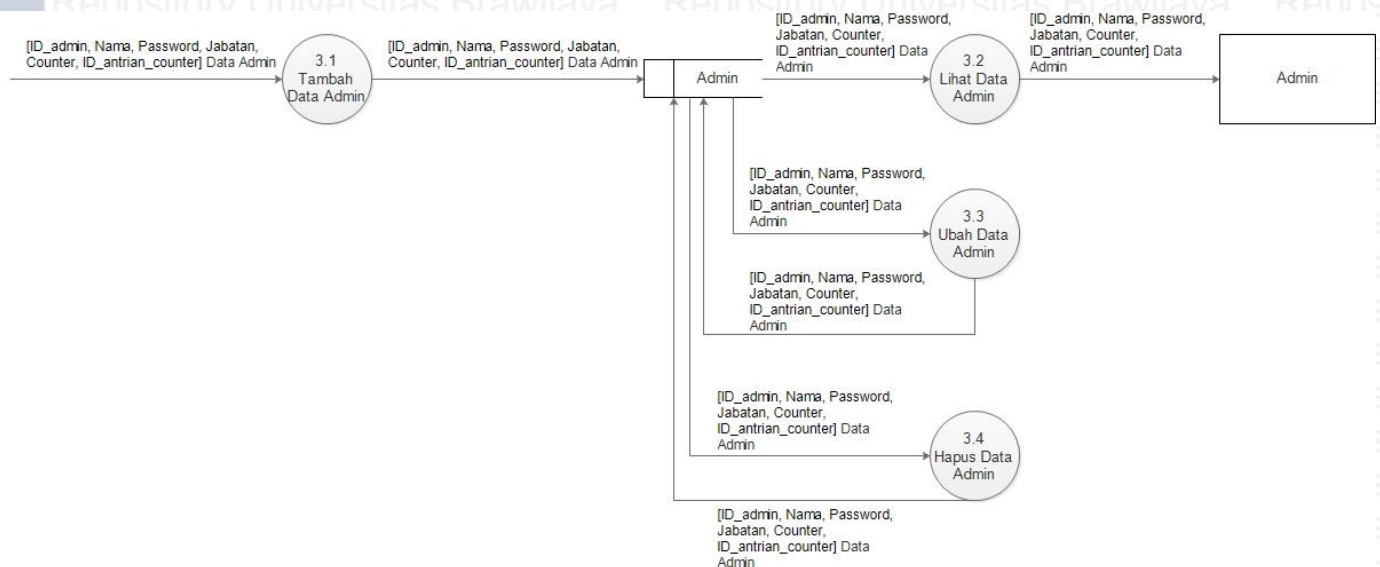
Gambar 4.8 Data Flow Diagram Level 2 Mengelola Data Antrian Sistem Antrian RSUD Dr. Soetrasno Rembang

Keterangan Gambar 4.8:

Pada proses Lihat Data Antrian dapat menerima masukan data antrian dari database antrian yang menghasilkan data antrian. Pada proses Panggil Ulang Pasien dapat menerima masukan data antrian dari database antrian yang menghasilkan data antrian yang akan ditampilkan pada Monitor. Pada proses Panggil Pasien Selanjutnya dapat menerima masukan data antrian dari database antrian yang menghasilkan data antrian yang akan ditampilkan pada Monitor.

4.7.3.5 Data Flow Diagram Level 2 Mengelola Data Admin

Berikut ini merupakan *Data Flow Diagram level 2* dari dekomposisi proses Mengelola Data Admin yang disajikan pada Gambar 4.9.





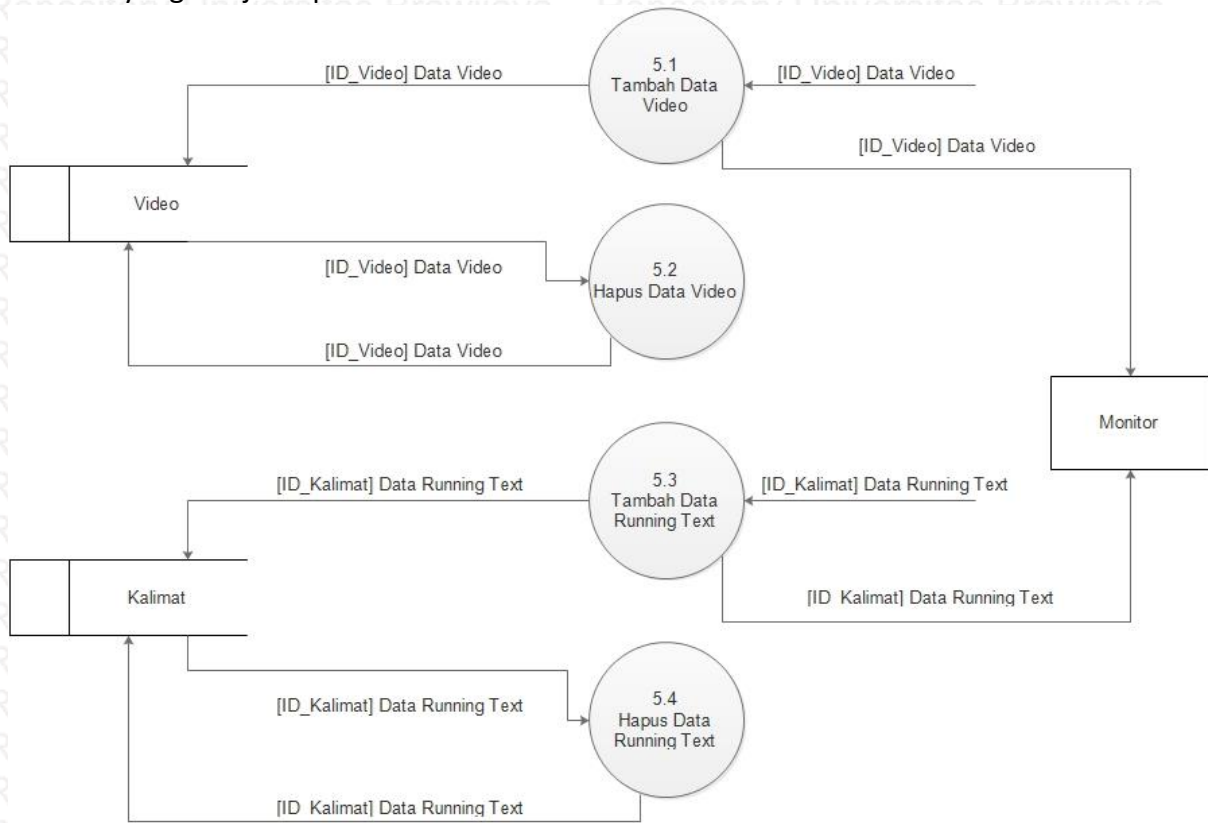
Gambar 4.9 Data Flow Diagram Level 2 Mengelola Data Admin Sistem Antrian RSUD Dr. Soetrasno Rembang

Keterangan Gambar 4.9:

Pada proses Buat Data Admin dapat menerima masukan data admin dan menghasilkan data admin yang disimpan dalam *database* admin. Pada proses Lihat Data Admin dapat menerima masukan data admin dan menghasilkan data admin. Pada proses Ubah Data Admin dapat menerima masukan data admin dan menghasilkan data admin dan dapat memperbarui data admin yang akan tersimpan dalam *database* admin. Pada proses Hapus Data Admin dapat menerima masukan data admin yang menghasilkan data admin serta dapat disimpan dalam *database* admin.

4.7.3.6 Data Flow Diagram Level 2 Mengelola Monitor

Berikut ini merupakan *Data Flow Diagram level 2* dari dekomposisi proses Mengelola Monitor yang disajikan pada Gambar 4.10.



Gambar 4.10 Data Flow Diagram Level 2 Mengelola Monitor Sistem Antrian RSUD Dr. Soetrasno Rembang

Keterangan Gambar 4.10:

Pada proses Tambah Data Video dapat menerima masukan data video dan menghasilkan data video yang disimpan dalam *database* video. Pada proses Hapus Data Video dapat menerima masukan data video yang menghasilkan data video yang dapat disimpan dalam *database* video. Pada proses Tambah Data *Running Text* dapat menerima masukan data *running text* dan menghasilkan data *running text* yang disimpan dalam *database* kalimat.



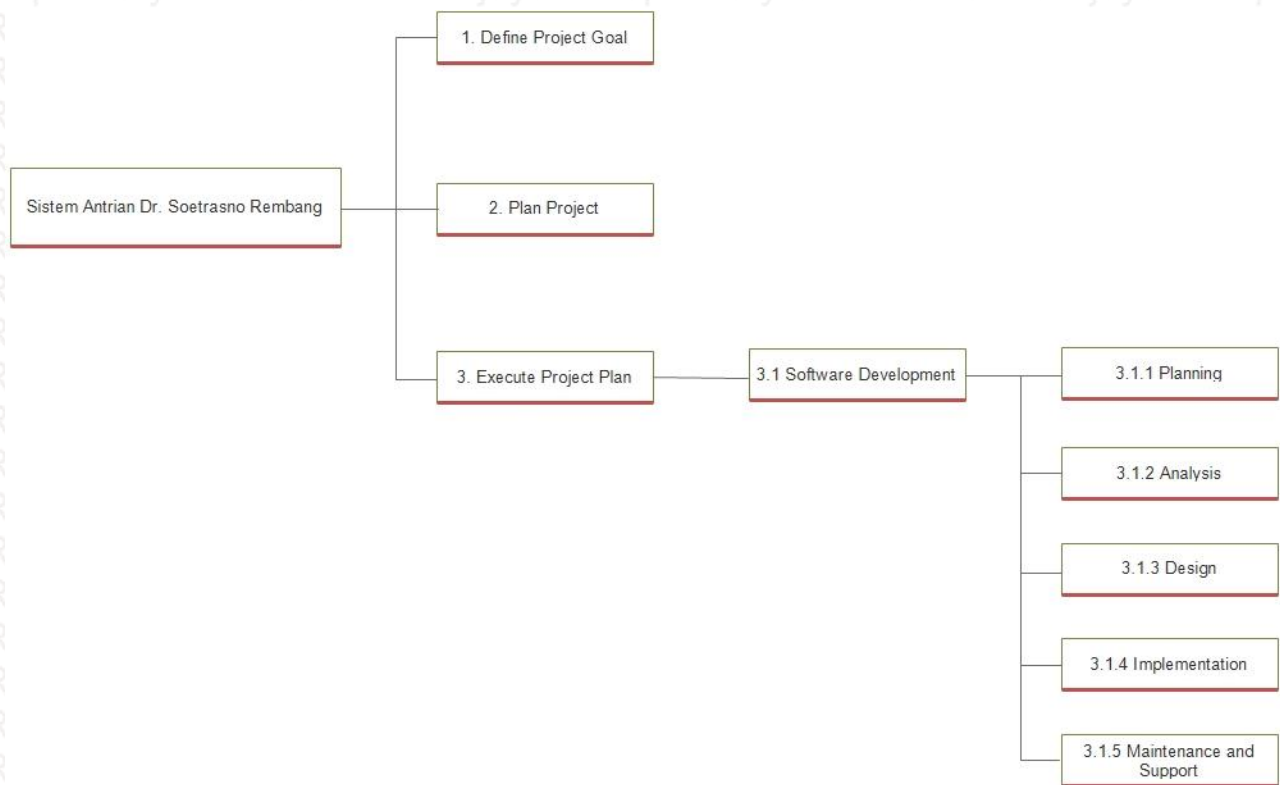
Pada proses Hapus Data *Running Text* dapat menerima masukan data *running text* yang menghasilkan data *running text* yang dapat disimpan dalam *database* kalimat. Seluruh data yang tersimpan dalam database video dan kalimat akan ditampilkan pada Monitor.

BAB 5 PEMBAHASAN

5.1 Work Breakdown Structure (WBS)

Work Breakdown Structure berisi dasar ruang lingkup proyek yang dibutuhkan untuk mencapai tujuan teknis pengerjaan. Proses pembuatan *Work Breakdown Structure* berdasarkan *Software Development Life Cycle* yang terlampir pada Bab 2 serta berdasarkan fase yang ada pada dokumen *Excel* yang dibuat oleh *Project Manager* dari CV Pabrik Teknologi.

Gambar 5.1 menunjukkan *Work Breakdown Structure* pada Metode *Guesstimate* memiliki empat level dimulai dari level pertama yaitu nama proyek yang akan dikembangkan yaitu Sistem Antrian RSUD Dr. Soetrasno Rembang. Pada level kedua terdiri dari tiga fase yaitu Fase *Define Project Goal*, *Plan Project* dan *Execute Project Plan*. Pada level ketiga terdiri dari satu fase yaitu fase *Software Development*. Pada level keempat terdiri dari lima fase yaitu fase *Requirement*, *Spesification*, *Design*, *Implementation* dan *Maintenance and Support*. Kemudian dilakukan penjabaran menggunakan *Gantt Chart* untuk mengetahui *task*, durasi, SDM dan biaya yang dibutuhkan pada setiap fase yang terlampir pada Lampiran F.



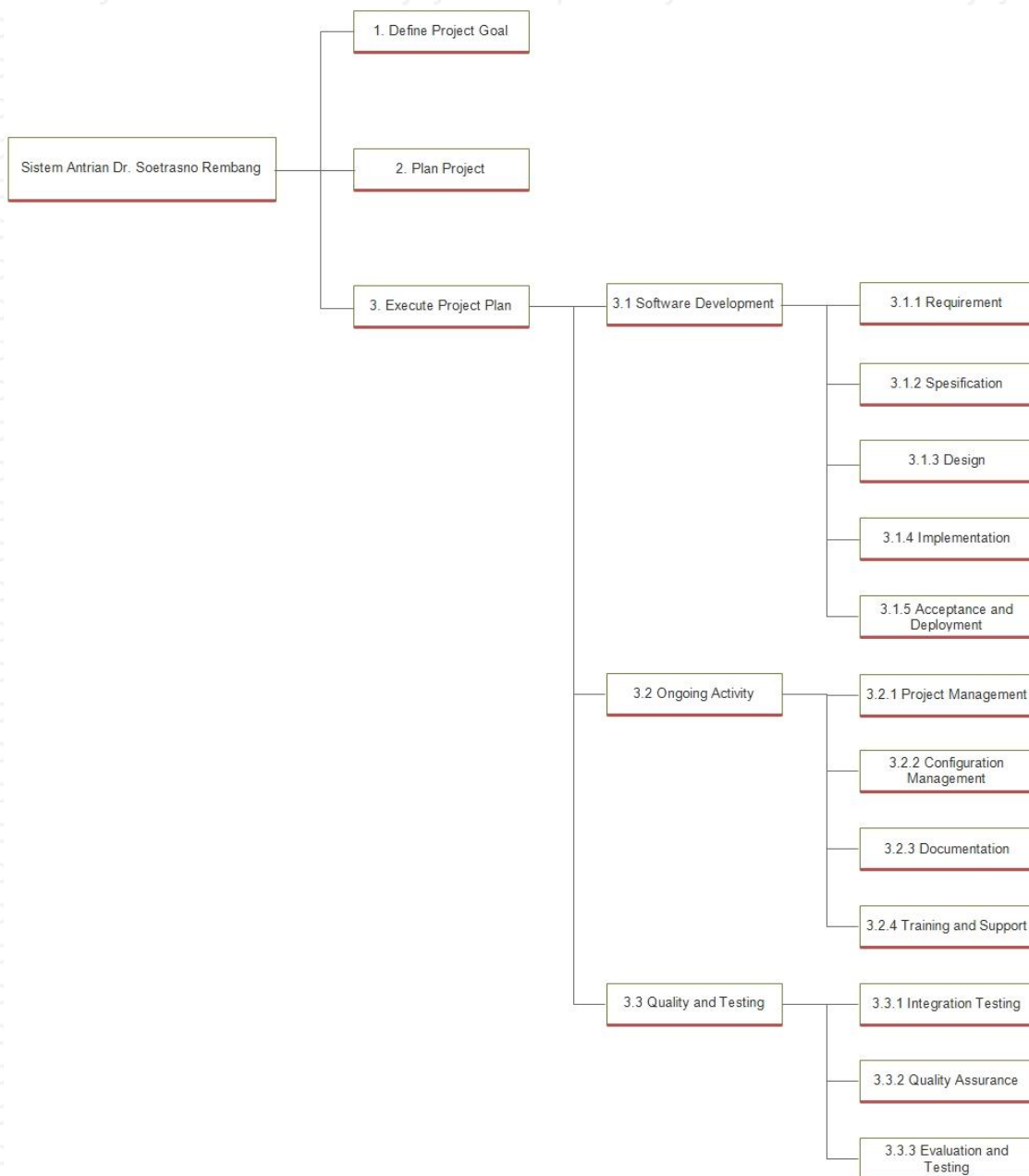
Gambar 5.1 Work Breakdown Structure Guesstimate

Planning, pada fase ini *Project Manager* mengidentifikasi kebutuhan dan perencanaan sistem dengan *stakeholder*. *Analysis*, pada fase ini *System Analyst* mencoba menyelidiki masalah atau kesempatan secara lebih detail dengan bertemu *stakeholders* (pengguna, *manager*, pelanggan) untuk mempelajari lebih lanjut tentang sistem yang dibutuhkan sehingga menghasilkan dokumen *requirement*. *Design*, *System Analyst* menggunakan *requirements* sebagai masukan untuk merancang arsitektur yang mendukung sistem.



Arsitektur ini meliputi membuat *Flow Chart* dan ER Diagram, *Mockup Front-end* dan *Back-end* sistem. *Implementation*, mencakup pengembangan dan pembangunan sistem, pengujian serta instalasi yang dilakukan oleh tim pengembang. *Maintenance and Support*, untuk tahap ini tim pengembang melakukan pemeliharaan dan peningkatan terhadap sistem sering diminta untuk memperbaiki kesalahan yang ditemukan dalam sistem.

Gambar 5.2 menunjukkan *Work Breakdown Structure* pada Metode *Use Case Point* dan *Function Point* yang memiliki empat level dimulai dari level pertama yaitu nama proyek yang akan dikembangkan yaitu Sistem Antrian RSUD Dr. Soetrasno Rembang. Pada level kedua terdiri dari tiga fase yaitu Fase *Define Project Goal*, *Plan Project* dan *Execute Project Plan*. Pada level ketiga terdiri dari tiga fase yaitu fase *Software Development*, *Ongoing Activity* dan *Quality and Testing*. Pada level keempat terdiri dari 12 fase yaitu fase *Requirement*, *Spesification*, *Design*, *Implementation*, *Acceptance and Deployment*, *Project Management*, *Configuration Management*, *Documentation*, *Training and Support*, *Integration Testing*, *Quality Assurance* dan *Evaluation and Testing*. Kemudian dilakukan penjabaran menggunakan *Gantt Chart* untuk mengetahui *task*, durasi, SDM dan biaya yang dibutuhkan pada setiap fase yang terlampir pada Lampiran G dan H.



Gambar 5.2 Work Breakdown Structure Use Case Point dan Function Point

Requirement, pada fase ini *System Analyst* mengidentifikasi kebutuhan dan perencanaan sistem dengan *stakeholder*. *Spesification*, pada fase ini *System Analyst* mencoba menyelidiki masalah atau kesempatan secara lebih detail dengan bertemu *stakeholders* (pengguna, *manager*, pelanggan) untuk mempelajari lebih lanjut tentang sistem yang dibutuhkan sehingga menghasilkan dokumen *requirement*. *Design*, *System Analyst* menggunakan *requirements* sebagai masukan untuk merancang arsitektur yang mendukung sistem. Arsitektur ini meliputi membuat *Flow Chart* dan ER Diagram, *Mockup Front-end* dan *Back-end* sistem. *Implementation*, mencakup pengembangan dan pembangunan sistem, pengujian serta instalasi yang dilakukan oleh *Software Engineer*. *Acceptance and Deployment*, untuk tahap ini *Software Engineer* melakukan pemeliharaan dan peningkatan terhadap sistem dan sering diminta untuk memperbaiki kesalahan yang ditemukan dalam sistem.

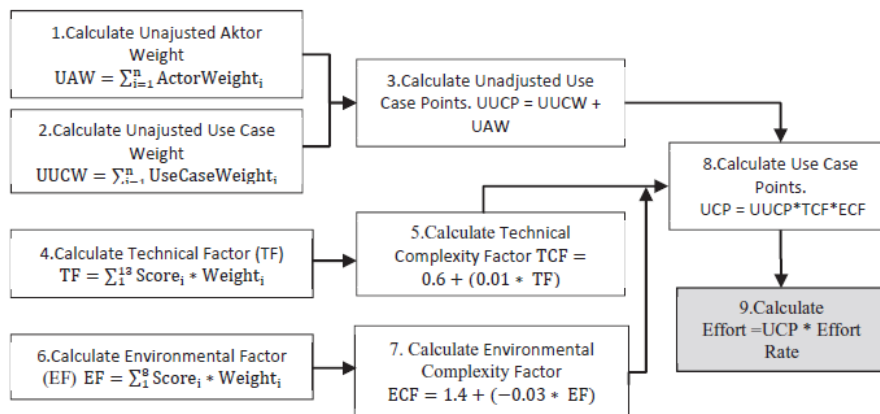


Project Management, PM melakukan kegiatan yang berhubungan dengan manajemen proyek meliputi memantau kemajuan proyek secara ketat, melaporkan manajemen resiko, memperbaiki jadwal proyek secara berkala. *Configuration Management, Software Engineer* melakukan pemeliharaan, pengelolaan, pengendalian dan pengembangan perangkat lunak. *Documentation*, membuat dokumen program yang dibuat oleh *Software Engineer*. *Training & Support*, pada fase ini *Software Engineer* membutuhkan beberapa dukungan teknis dan pelatihan dalam memecahkan masalah teknis yang mungkin terjadi guna meningkatkan efisiensi dan produktivitas.

Integration Testing, Test Analyst melakukan uji integrasi antara modul, sistem dan komponen eksternal. Kemudian melakukan hasil analisis dan menangani masalah yang terjadi. *Milestone* dari fase ini adalah perangkat lunak terintegrasi. *Quality Assurance*, PM memastikan seluruh aktivitas yang terjadi pada setiap fase telah sesuai dengan standar internal dan menghasilkan *deliverables* yang benar yaitu pada *Front-end* dan *Back-end*. *Evaluation & Testing, Test Analyst* melakukan kegiatan evaluasi dan pengujian, termasuk kegiatan validasi dan verifikasi.

5.2 Perhitungan Biaya dengan Metode Use Case Point

Proses perhitungan *Use Case Point* membutuhkan *Use Case Diagram* dan *Use Case Scenario*. Metode ini merupakan pendekatan yang terdokumentasi dengan baik dalam memperkirakan aktivitas pengembangan perangkat lunak (Dalal et al, 2015). Langkah-langkah yang perlu dilakukan dalam proses estimasi usaha menggunakan metode *Use Case Point* akan disajikan pada Gambar 5.3, kemudian setelah mendapatkan usaha langkah selanjutnya menggunakan *guideline* dari Kassem Saleh (2011) untuk mendapatkan estimasi biaya yang diperlukan untuk pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang yang telah terlampir pada Bab 2.



Gambar 5.3 Langkah-langkah Metode Use Case Point

Sumber: Sholiq (2015)

5.2.1 Perhitungan Unadjusted Use Case Point (UUCP)

Nilai *Unadjusted Use Case Point* (UUCP) berdasarkan hasil analisis pada *Use Case Diagram* dari Sistem Antrian RSUD Dr. Soetrasno Rembang. Untuk menghitung nilai *Unadjusted Use Case Point* (UUCP), maka perhitungan yang dilakukan terlebih dahulu adalah menghitung nilai *Unadjusted Actor Weight* (UAW) dan nilai *Unadjusted Use Case Weight* (UUCW) (Ribu, 2001).



5.2.1.1 Perhitungan *Unadjusted Actor Weight* (UAW)

Nilai *Unadjusted Actor Weight* (UAW) didapat dengan mengidentifikasi dan mengklasifikasikan jenis aktor. Jenis aktor diklasifikasikan menjadi tiga kategori yaitu; *Simple*, *Average* dan *Complex*. *Simple actor* merupakan aktor yang berinteraksi melalui API seperti *Command Prompt* dengan bobot 1. *Average actor* adalah aktor yang berinteraksi melalui protokol seperti TCP/IP, FTP, HTTP atau disebut aktor yang melakukan penyimpanan data (file, RDBMS) dengan bobot 2. Sedangkan *Complex Actor* merupakan aktor yang berinteraksi melalui GUI atau halaman web dengan bobot 3 (Ribu, 2001).

Aktor-aktor yang akan dikategorikan merupakan hasil dari analisis *Use Case Diagram* pada Sistem Antrian RSUD Dr. Soetrasno Rembang yang terlampir pada Bab 4. *Use Case Diagram* dibuat berdasarkan hasil wawancara dengan narasumber dari CV Pabrik Teknologi yang kemudian dilakukan validasi setelah *Use Case Diagram* selesai dibuat.

Karena kebutuhan aktor yang lebih mudah bekerja dengan *Web page* untuk menjalankan aplikasi yang akan dikembangkan yaitu Sistem Antrian RSUD Dr. Soetrasno Rembang maka terdapat 3 aktor yang berperan dalam sistem yang dikategorikan dalam tipe *Complex*. Penjelasan mengenai peran aktor-aktor dalam berinteraksi dengan sistem yang disajikan pada Tabel 5.1.

Tabel 5.1 Dekripsi Aktor Pada Sistem

	Nama Aktor	Deskripsi	Tipe
1	Admin	Mengelola Data <i>User</i> , Data Admin, Data Video, Data Running Text, Ubah Kuota SMS, Mengelola Data Pesan Masuk dan Pesan Keluar melalui <i>website</i>	<i>Complex</i>
2	Admin Counter	Mengelola Data Antrian melalui <i>website</i>	<i>Complex</i>
3	Pasien	Mengambil Nomor Antrian dan Mengecek Antrian melalui <i>website</i>	<i>Complex</i>

Setelah melakukan pengklasifikasian aktor pada *Use Case Diagram* Sistem Antrian RSUD Dr. Soetrasno Rembang, langkah selanjutnya yaitu melakukan perhitungan total *Unadjusted Actor Weight* (UAW) dengan mengalikan bobot dengan jumlah aktor sesuai dengan tipe aktor yang akan disajikan pada Tabel 5.2.

Tabel 5.2 Perhitungan Total *Unadjusted Actor Weight* (UAW)

Tipe	Bobot	Jumlah Aktor	Bobot x Jumlah Aktor
<i>Simple</i>	1	0	0
<i>Average</i>	2	0	0
<i>Complex</i>	3	3	9
Total UAW			9

5.2.1.2 Perhitungan *Unadjusted Use Case Weight* (UUCW)

Nilai *Unadjusted Use Case Weight* (UUCW) didapat dari hasil menentukan kategori *use case* yaitu sebagai *simple*, *medium* atau *complex* yang ditentukan berdasarkan jumlah transaksi sukses dan alternatif yang ada dalam *Use Case Scenario* yang terlampir pada Bab 4. Untuk *use case simple* memiliki ≤ 3 transaksi dalam *use case* dengan bobot kompleksitasnya



adalah 5, untuk *use case average* memiliki 4-7 transaksi dalam *use case* dengan bobot kompleksitasnya adalah 10 sedangkan *use case complex* memiliki >7 transaksi dalam *use case* dengan bobot kompleksitasnya adalah 15 (Sholih, 2015).

Penelitian ini menggunakan aturan penentuan transaksi berdasarkan *actor goals* atau disebut *semantic point*. Jika melihat transaksi dari *semantic point*, sebagian besar *use case* memiliki *goals* atau tujuan yang sama. *Semantic point* telah membedakan 12 jenis transaksi yaitu *Create, Retrieve, Update, Delete, Link, Delete Link, Asynchronous Retrieve, Dynamic Retrieve, Dynamic Retrieve, Transfer, Check Object, Complex Internal Activity* dan *Change State* (Ochodek et al., 2011).

Merujuk pada *Use Case Scenario* yang terlampir pada Bab 4, berikut merupakan analisis jumlah transaksi yang ada pada *use case* yang disajikan pada Tabel 5.3 sampai Tabel 5.27.

Tabel 5.3 Transaksi Use Case Login

<p>Transaksi 1:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan menu login untuk masuk ke dalam sistem 2. <i>Guest</i> memasukkan <i>username</i> dan <i>password</i> 3. Sistem melakukan autentikasi bagi <i>guest</i> yang akan masuk ke dalam sistem 4. Sistem menampilkan menu beranda jika <i>username</i> dan <i>password</i> valid dan terautentikasi oleh sistem <p>Transaksi 2:</p> <p>a. {username dan password} tidak valid</p> <p>Jika di <i>basic flow</i>, <i>guest</i> memasukkan <i>username</i> dan <i>password</i> tidak valid, sistem menampilkan pesan kesalahan. <i>Guest</i> bisa memilih untuk kembali ke awal <i>basic flow</i> atau membatalkan login, dan <i>use case</i> selesai</p>
<p>Total Transaksi : 2 transaksi</p>
<p>Kategori : Simple</p>

Tabel 5.4 Transaksi Use Case Lihat Data User

<p>Transaksi 1:</p> <ol style="list-style-type: none"> 1. Admin memilih menu Data User lalu sistem menampilkan data <i>user</i> dan operasi detail, tambah, ubah dan hapus 2. Admin memilih operasi detail lalu sistem menampilkan detail data <i>user</i>
<p>Total Transaksi : 1 transaksi</p>
<p>Kategori : Simple</p>

Tabel 5.5 Transaksi Use Case Tambah Data User

<p>Transaksi 1:</p> <ol style="list-style-type: none"> 1. Admin memilih menu Data <i>User</i> lalu sistem menampilkan data <i>user</i> dan operasi detail, tambah, ubah dan hapus 2. Admin memilih operasi tambah lalu sistem menampilkan menu untuk menambah data <i>user</i> 3. Admin memasukkan data <i>user</i> {mengisi data user} 4. Sistem menampilkan informasi bahwa data berhasil ditambahkan <p>Transaksi 2:</p>
--



a. Data yang dimasukkan tidak lengkap

Pada {mengisi data user} di *basic flow*, apabila admin tidak mengisi salah satu data yang harus terisi maka sistem akan menampilkan informasi bahwa terdapat data yang belum terisi.

Total Transaksi : 2 transaksi

Kategori : Simple

Tabel 5.6 Transaksi Use Case Ubah Data User

Transaksi 1:

1. Admin memilih menu Data User lalu sistem menampilkan data user dan operasi detail, tambah, ubah dan hapus
2. Admin memilih operasi ubah lalu sistem menampilkan data user yang akan dilakukan perubahan
3. Admin mengubah data user pada sistem dan menyimpan hasil perubahan {mengisi data user}
4. Sistem menampilkan informasi bahwa data user berhasil diubah

Transaksi 2:

a. Data yang dimasukkan tidak lengkap

Pada {mengisi data user} di *basic flow*, apabila admin tidak mengisi salah satu data yang harus terisi maka sistem akan menampilkan informasi bahwa terdapat data yang belum terisi.

Total Transaksi : 2 transaksi

Kategori : Simple

Tabel 5.7 Transaksi Use Case Hapus Data User

Transaksi 1:

1. Admin memilih menu Data User lalu sistem menampilkan data user dan operasi detail, tambah, ubah dan hapus
2. Admin memilih operasi hapus lalu sistem melakukan operasi dan verifikasi penghapusan data user
3. Sistem menampilkan informasi bahwa data user berhasil dihapus

Total Transaksi : 1 transaksi

Kategori : Simple

Tabel 5.8 Transaksi Use Case Baca Pesan Masuk

Transaksi 1:

1. Admin memilih sub menu pesan masuk dan sistem menampilkan data pesan masuk dan operasi detail dan hapus
2. Admin memilih detail pada salah satu pesan masuk lalu sistem menampilkan detail pesan masuk

Total Transaksi : 1 transaksi

Kategori : Simple

Tabel 5.9 Transaksi *Use Case* Hapus Pesan Masuk**Transaksi 1:**

1. Admin memilih sub menu pesan masuk dan sistem menampilkan data pesan masuk dan operasi detail dan hapus
2. Admin memilih operasi hapus lalu sistem melakukan operasi dan verifikasi penghapusan data pesan masuk
3. Sistem menampilkan informasi bahwa data pesan masuk berhasil dihapus

Total Transaksi : 1 transaksi**Kategori : Simple**Tabel 5.10 Transaksi *Use Case* Baca Pesan Keluar**Transaksi 1:**

1. Admin memilih sub menu pesan keluar dan sistem menampilkan data pesan keluar dan operasi detail dan hapus
2. Admin memilih detail pada salah satu pesan keluar lalu sistem menampilkan detail pesan keluar

Total Transaksi : 1 transaksi**Kategori : Simple**Tabel 5.11 Transaksi *Use Case* Hapus Pesan Keluar**Transaksi 1:**

1. Admin memilih sub menu pesan keluar dan sistem menampilkan data pesan keluar dan operasi detail dan hapus
2. Admin memilih operasi hapus lalu sistem melakukan operasi dan verifikasi penghapusan data pesan keluar
3. Sistem menampilkan informasi bahwa data pesan keluar berhasil dihapus

Total Transaksi : 1 transaksi**Kategori : Simple**Tabel 5.12 Transaksi *Use Case* Lihat Kuota SMS**Transaksi 1:**

1. Admin memilih sub menu kuota SMS lalu sistem menampilkan jumlah kuota SMS dan tombol ubah

Total Transaksi : 1 transaksi**Kategori : Simple**Tabel 5.13 Transaksi *Use Case* Ubah Kuota SMS**Transaksi 1:**

1. Admin memilih sub menu Kuota SMS lalu sistem menampilkan jumlah kuota SMS dan operasi ubah
2. Admin memilih operasi ubah lalu admin mengubah jumlah kuota SMS pada sistem dan menyimpan hasil perubahan {mengisi jumlah kuota SMS}



3. Sistem menampilkan informasi bahwa jumlah kuota SMS berhasil diperbarui

Transaksi 2:

- a. Data yang dimasukkan tidak sesuai dengan format

Pada {mengisi jumlah kuota SMS} pada *sub flow*, apabila admin mengisi selain format angka maka sistem akan menampilkan informasi bahwa data yang dimasukkan salah

Total Transaksi : 2 transaksi

Kategori : Simple

Tabel 5.14 Transaksi Use Case Lihat Data Antrian

Transaksi 1:

1. Admin Counter memilih menu Data Antrian lalu sistem menampilkan data antrian dan operasi panggil ulang pasien dan panggil pasien selanjutnya

Total Transaksi : 1 transaksi

Kategori : Simple

Tabel 5.15 Transaksi Use Case Panggil Ulang Pasien

Transaksi 1:

1. Admin Counter memilih menu Data Antrian lalu sistem menampilkan data antrian dan operasi panggil ulang pasien dan panggil pasien selanjutnya
2. Admin memilih operasi panggil ulang pasien lalu sistem melakukan pemanggilan ulang pasien
3. Admin memasukkan status pelayanan {mengisi status pelayanan}
4. Sistem menampilkan informasi pemanggilan ulang berhasil

Total Transaksi : 1 transaksi

Kategori : Simple

Tabel 5.16 Transaksi Use Case Panggil Pasien Selanjutnya

Transaksi 1:

1. Admin Counter memilih menu Data Antrian lalu sistem menampilkan data antrian dan operasi panggil ulang pasien dan panggil pasien selanjutnya
2. Admin memilih operasi panggil pasien selanjutnya lalu sistem melakukan pemanggilan pasien selanjutnya
3. Sistem menampilkan informasi pemanggilan ulang berhasil

Total Transaksi : 1 transaksi

Kategori : Simple

Tabel 5.17 Transaksi Use Case Tambah Data Video

Transaksi 1:

1. Admin memilih menu Monitor lalu sistem menampilkan data video dan operasi tambah dan hapus
2. Admin memilih operasi tambah lalu sistem akan menampilkan menu untuk mengunggah *file* video
3. Sistem akan menampilkan informasi bahwa data video berhasil ditambahkan

**Total Transaksi : 1 transaksi****Kategori : Simple****Tabel 5.18 Transaksi Use Case Hapus Data Video****Transaksi 1:**

1. Admin memilih menu Monitor lalu sistem menampilkan data video dan operasi tambah dan hapus
2. Admin memilih operasi hapus lalu sistem melakukan operasi hapus data video
3. Sistem menampilkan informasi bahwa data video berhasil dihapus

Total Transaksi : 1 transaksi**Kategori : Simple****Tabel 5.19 Transaksi Use Case Tambah Data Running Text****Transaksi 1:**

1. Admin memilih menu Monitor lalu sistem menampilkan data *running text* dan operasi tambah dan hapus
2. Admin memilih operasi tambah lalu sistem akan menampilkan menu untuk mengunggah *file running text*
3. Sistem akan menampilkan informasi bahwa data *running text* berhasil ditambahkan

Total Transaksi : 1 transaksi**Kategori : Simple****Tabel 5.20 Transaksi Use Case Hapus Data Running Text****Transaksi 1:**

1. Admin memilih menu Monitor lalu sistem menampilkan data *running text* dan operasi tambah dan hapus
2. Admin memilih operasi hapus lalu sistem melakukan operasi hapus data *running text*
3. Sistem menampilkan informasi bahwa data *running text* berhasil dihapus

Total Transaksi : 1 transaksi**Kategori : Simple****Tabel 5.21 Transaksi Use Case Mengambil Nomor Antrian****Transaksi 1:**

1. Pasien memilih memilih operasi antri kemudian sistem menampilkan informasi pengambilan nomor antrian berhasil

Total Transaksi : 1 transaksi**Kategori : Simple****Tabel 5.22 Transaksi Use Case Mengecek Nomor Antrian****Transaksi 1:**

1. Pasien memilih operasi cek antrian lalu sistem menampilkan nomor antrian pasien dan antrian yang dipanggil saat ini



Total Transaksi : 1 transaksi

Kategori : Simple

Tabel 5.23 Transaksi Use Case Melakukan Registrasi

Transaksi 1:

1. Sistem menampilkan menu registrasi untuk masuk ke dalam sistem
2. Pasien memasukkan nomor {mengisi nomor HP}
3. Sistem secara otomatis mengirimkan *password* ke nomor HP pasien
4. Sistem menampilkan informasi bahwa *password* dikirim ke nomor HP pasien

Transaksi 2:

a. Pasien tidak menerima *password*

Pada {mengisi nomor HP} jika pasien memasukkan nomor yang tidak aktif, maka pasien gagal menerima *password* sehingga pasien harus memulai *basic flow* dari awal

Total Transaksi : 2 transaksi

Kategori : Simple

Tabel 5.24 Transaksi Use Case Lihat Data Admin

Transaksi 1:

1. Admin memilih menu Data Admin lalu sistem menampilkan data admin dan operasi detail, tambah, ubah dan hapus
2. Admin memilih operasi detail lalu sistem menampilkan detail data admin

Total Transaksi : 1 transaksi

Kategori : Simple

Tabel 5.25 Transaksi Use Case Tambah Data Admin

Transaksi 1:

1. Admin memilih menu Data Admin lalu sistem menampilkan data admin dan operasi detail, tambah, ubah dan hapus
2. Admin memilih operasi tambah lalu sistem menampilkan menu untuk menambah data admin
3. Admin memasukkan data admin {mengisi data admin}
4. Sistem menampilkan informasi bahwa data berhasil ditambahkan

Transaksi 2:

a. Data yang dimasukkan tidak lengkap

Pada {mengisi data admin} di *basic flow*, apabila admin tidak mengisi salah satu data yang harus terisi maka sistem akan menampilkan informasi bahwa terdapat data yang belum terisi.

Total Transaksi : 2 transaksi

Kategori : Simple

Tabel 5.26 Transaksi *Use Case* Ubah Data Admin

Transaksi 1:
<ol style="list-style-type: none"> Admin memilih menu Data Admin lalu sistem menampilkan data admin dan operasi detail, tambah, ubah dan hapus Admin memilih operasi ubah lalu sistem menampilkan data admin yang akan dilakukan perubahan Admin mengubah data admin pada sistem dan menyimpan hasil perubahan {mengisi data admin} Sistem menampilkan informasi bahwa data admin berhasil diubah
Transaksi 2:
<ol style="list-style-type: none"> Data yang dimasukkan tidak lengkap <p>Pada {mengisi data admin} di <i>basic flow</i>, apabila admin tidak mengisi salah satu data yang harus terisi maka sistem akan menampilkan informasi bahwa terdapat data yang belum terisi.</p>
Total Transaksi : 2 transaksi
Kategori : Simple

Tabel 5.27 Transaksi *Use Case* Hapus Data Admin

Transaksi 1:
<ol style="list-style-type: none"> Admin memilih menu Data Admin lalu sistem menampilkan data admin dan operasi detail, tambah, ubah dan hapus Admin memilih operasi hapus lalu sistem melakukan operasi dan verifikasi penghapusan data admin Sistem menampilkan informasi bahwa data admin berhasil dihapus
Total Transaksi : 1 transaksi
Kategori : Simple

Setelah mengelompokkan skenario menjadi sejumlah transaksi dan dapat diketahui tipe *use casenya*, maka langkah selanjutnya melakukan perhitungan *Unadjusted Use Case Weight* (UUCP) dengan mengalikan bobot dengan jumlah *use case* sesuai dengan tipe yang akan disajikan pada Tabel 5.15.

Tabel 5.28 Perhitungan Total *Unadjusted Use Case Weight* (UUCW)

Type	Bobot	Jumlah <i>Use Case</i>	Bobot x Jumlah <i>Use Case</i>
Simple	5	25	125
Average	10	0	0
Complex	15	0	0
Total UUCW			125

Kemudian setelah mendapatkan nilai *Unadjusted Actor Weight* (UAW) dan nilai *Unadjusted Use Case Weight* (UUCW) langkah selanjutnya melakukan perhitungan nilai *Unadjusted Use Case Point* (UUCP) berdasarkan persamaan yang terlampir pada Bab 2.

$$\begin{aligned} \text{UUCP} &= \text{UAW} + \text{UUCW} \\ &= 9 + 125 = 134 \end{aligned}$$



Nilai **Unadjusted Use Case Point (UUCP)** sebesar **134** didapat dari hasil penjumlahan nilai *Unadjusted Actor Weight (UAW)* dan nilai *Unadjusted Use Case Weight (UUCW)*. Nilai UAW didapat dari mengkategorikan aktor-aktor pada *Use Case Diagram* sedangkan nilai UUCW didapat dari mengkategorikan *use case* yang ada pada *Use Case Diagram*.

5.2.2 Perhitungan *Technical and Environment Complexity Factor*

5.2.2.1 *Technical Complexity Factor (TCF)*

Pada tahap ini dilakukan penilaian melalui penyebaran lembar penilaian yang telah divalidasi oleh *expert judgement*, perhitungan ini menilai skor yang diisi oleh tim pengembang pada setiap faktor teknis. Dalam pelaksanaan penyebaran lembar penilaian ini melibatkan responden dari tim pengembang Sistem Antrian RSUD Dr. Soetrasno Rembang yang tergabung dalam CV Pabrik Teknologi.

Rekap hasil dan perhitungan pada masing-masing faktor teknis yang disajikan pada Tabel 5.16. Perhitungan skor untuk setiap faktor didapat dari hasil pengisian lembar penilaian oleh tim pengembang kemudian dilakukan penjumlahan dan mencari rata-rata dari setiap faktor. Untuk rekap seluruh hasil pengisian lembar penilaian oleh tim pengembang terlampir pada Lampiran D.

Tabel 5.29 Perhitungan *Technical Complexity Factor*

	Technical Complexity Factor	Bobot	Skor (0-5)	B x S
1	<i>Distributed System Required</i>	2	4	8
2	<i>Response Time</i>	2	5	10
3	<i>End User Efficiency</i>	1	5	5
4	<i>Complex Internal Processing Required</i>	1	3	3
5	<i>Reusable Code</i>	1	3	3
6	<i>Easy to Install</i>	0.5	3	1.5
7	<i>Easy to Use</i>	0.5	4	2
8	<i>Portable</i>	2	5	10
9	<i>Easy to Change</i>	1	4	4
10	<i>Concurrent</i>	1	4	4
11	<i>Security Features</i>	1	3	3
12	<i>Access for Third Parties</i>	1	3	3
13	<i>Special Training Required</i>	1	1	1
			Total TF	57.5
TOTAL TCF		$0.6 + (0.01 * 57.5) = 1,175$		

Nilai TCF pada Tabel 5.15 sebesar **1,175** didapat dari rumus perhitungan $TCF = 0.6 + (0.01 * TF)$. Dimana TF adalah hasil penjumlahan dari seluruh perkalian skor dan bobot pada 13 faktor.

5.2.2.2 *Environment Complexity Factor (ECF)*

Pada tahap ini juga dilakukan penilaian melalui penyebaran lembar penilaian yang telah divalidasi oleh *expert judgement*, perhitungan ini menilai skor yang diisi oleh tim pengembang pada setiap faktor lingkungan. Dalam pelaksanaan penyebaran lembar

penilaian ini melibatkan responden dari tim pengembang Sistem Antrian RSUD Dr. Soetrasno Rembang yang tergabung dalam CV Pabrik Teknologi.

Rekap hasil dan perhitungan pada masing-masing faktor lingkungan yang disajikan pada Tabel 5.17. Perhitungan skor untuk setiap faktor didapat dari hasil pengisian lembar penilaian oleh tim pengembang kemudian di lakukan penjumlahan dan mencari rata-rata dari setiap faktor. Untuk rekap seluruh hasil pengisian lembar penilaian oleh tim pengembang terlampir pada Lampiran D.

Tabel 5.30 Perhitungan *Environment Complexity Factor*

<i>Environment Complexity Factor</i>		Bobot	Skor (0-5)	B x S
1	<i>Familiarity With The Project</i>	1.5	3	4.5
2	<i>Application Experience</i>	0.5	3	1.5
3	<i>OO Programming Experience</i>	1	4	4
4	<i>Lead Analyst Capability</i>	0.5	3	1.5
5	<i>Motivation</i>	1	3	3
6	<i>Stable Requirements</i>	2	4	8
7	<i>Part Time Staff</i>	-1	4	-4
8	<i>Difficult Programming Language</i>	-1	2	-2
			Total EF	16,5
TOTAL ECF		1.4 + (- 0.03 * 16.5) = 0,905		

Nilai ECF pada Tabel 5.17 sebesar **0,905** didapat dari rumus perhitungan $ECF = 1.4 + (-0.03 * EF)$. Dimana EF adalah penjumlahan dari seluruh perkalian skor dan bobot pada 8 faktor.

5.2.3 Perhitungan Nilai *Use Case Point* (UCP)

Sebelum dilakukan perhitungan *hours of effort*, harus dihitung terlebih dahulu nilai *Use Case Point* (UCP) yang didapatkan dari perkalian antara nilai UUCP, nilai TCF dan nilai ECF sesuai dengan persamaan berikut ini.

$$\begin{aligned}
 \text{Use Case Point (UCP)} &= \text{UUCP} * \text{TCF} * \text{ECF} \\
 &= 134 * 1,175 * 0,905 \\
 &= \mathbf{142,49}
 \end{aligned}$$

Nilai UCP sebesar **142,49** didapat dari hasil perkalian antara nilai UUCP, nilai TCF dan nilai ECF.

5.2.4 Perhitungan *Hours of Effort*

Tabel 5.31 Perhitungan *Use Case Point* (UCP)

Perhitungan	Rumus	Hasil
<i>Unadjusted Use Case Weight</i> (UUCW)		125
<i>Unadjusted Actor Weight</i> (UAW)		9
<i>Unadjusted Use Case Point</i> (UUCP)	UUCW + UAW	134

Tabel 5.31 Perhitungan *Use Case Point* (UCP) (Lanjutan)

Perhitungan	Rumus	Hasil
<i>Technical Complexity Factor</i>	$0.6 + (0.01 * TF)$	1.175
<i>Environment Complexity Factor</i>	$1.4 + (- 0.03 * EF)$	0,905
<i>Use Case Point</i> (UCP)	$UUCP * TCF * ECF$	142,49

Selanjutnya mengubah nilai *Use Case Point* (UCP) yang telah didapat menjadi nilai *effort* yaitu *Hours of Effort*, maka nilai UCP harus dikalikan dengan nilai *staff-hour per Use Case Point* berdasarkan teori Karner (1993). Perhitungan *effort* yang dibutuhkan untuk pengerjaan Sistem Antrian RSUD Dr. Soetrasno Rembang dapat diketahui dengan menggunakan persamaan berikut ini.

$$\begin{aligned} \text{Hours of Effort} &= 142,49 \times 20 \\ &= \mathbf{2.850 \text{ (jam)}} \end{aligned}$$

Berdasarkan persamaan diatas, nilai **142,49** merupakan nilai *Use Case Point* (UCP) yang didapat dari hasil penjumlahan antara nilai *Unadjusted Use Case Point* (UUCP) dan nilai *Technical and Environment Complexity Factor* (TCF dan ECF).

Hasil dari nilai *staff-hour per Use Case Point* berdasarkan teori Karner yaitu **20 staff hours**, sehingga dalam proyek ini memiliki waktu kerja sebanyak **2.850 jam**.

5.2.5 Perhitungan Biaya Total

Mengacu pada *guideline* Shaleh (2011), maka nilai *Hours of Effort* yang telah didapat pada langkah sebelumnya, dibagi menjadi tiga kelompok aktivitas yang disajikan pada Tabel 5.32.

Tabel 5.32 *Hours of Effort* ke Dalam 3 Aktivitas

	Kelompok Aktivitas	% Effort	Hours of Effort
1	Software Development		
A	Requirement	7,5%	214
B	Spesifications	7,5%	214
C	Design	10%	285
D	Implemention	10%	285
E	Acceptance & Deployment	7,5%	214
	Total	42,5%	1211
2	On Going Activity		
A	Project Management	8,34%	238
B	Configuration Management	4,16%	119
C	Documentation	4,16%	119
D	Training & Support	4,16%	119
	Total	20,82%	593
3	Quality and Testing		
A	Integration Testing	7,5%	214
B	Quality Assurance	8,34%	238
C	Evaluation & Testing	20,84%	594
	Total	36,68%	1045



Selanjutnya, menentukan standar gaji yang diperlukan untuk setiap fase dengan mengacu pada Indonesia *Salary Guide* 2006 yang diterbitkan oleh *Kelly Service*. Dalam penelitian ini, penentuan posisi yang dibutuhkan setiap fase diambil berdasarkan teori yang terlampir pada Bab 2. Kemudian menentukan standar gaji per jam dengan melakukan pembagian antara standar gaji per bulan dengan **160** jam kerja dan hasilnya disajikan pada Tabel 5.33.

Tabel 5.33 Penetapan Standar Gaji

Posisi Dalam <i>Salary Guide</i>	Standar Gaji (per-bulan) (Rp)	Standar Gaji (per-jam)
<i>Project Manager</i>	Rp 20.000.000,00	Rp 125.000,00
<i>System Analyst</i>	Rp 7.000.000,00	Rp 43.750,00
<i>Software Engineer</i>	Rp 5.000.000,00	Rp 31.250,00
<i>Test Analyst</i>	Rp 8.000.000,00	Rp 50.000,00

Pada langkah terakhir, setelah menentukan posisi dalam setiap fase berdasarkan deskripsi yang terlampir pada Bab 2 yang mengacu dengan Indonesia *Salary Guide* 2006 yang diterbitkan oleh *Kelly Servic*, maka selanjutnya menghitung biaya per fase dengan melakukan perkalian antara standar gaji per jam dengan *hours of effort* yang telah ditentukan pada Tabel 5.32. Kemudian melakukan penjumlahan seluruh fase yang menghasilkan estimasi biaya total yang disajikan pada Tabel 5.34.

Tabel 5.34 Perhitungan Biaya Total

No	Kelompok Aktivitas	Posisi dalam <i>Salary Guide</i>	% Effort	Hours of Effort	Standar Gaji (per-jam)	Total
1	<i>Software Development</i>					
A	<i>Requirement</i>	<i>System Analyst</i>	7,5%	214 jam	Rp 43.750,00	Rp 9.362.500
B	<i>Spesification</i>	<i>System Analyst</i>	7,5%	214 jam	Rp 43.750,00	Rp 9.362.500
C	<i>Design</i>	<i>System Analyst</i>	10%	285 jam	Rp 43.750,00	Rp 12.468.750
D	<i>Implementation</i>	<i>Software Engineer</i>	10%	285 jam	Rp 31.250,00	Rp 8.906.250
E	<i>Acceptance & Deployment</i>	<i>Software Engineer</i>	7,5%	214 jam	Rp 31.250,00	Rp 6.687.500
	Total		42,5%	1211 jam		Rp 46.787.500
2	<i>Ongoing Activity</i>					
A	<i>Project Management</i>	<i>Project Manager</i>	8,34%	238 jam	Rp 125.000,00	Rp 29.750.000
B	<i>Configuration Management</i>	<i>Software Engineer</i>	4,16%	119 jam	Rp 31.250,00	Rp 3.718.750
C	<i>Documentation</i>	<i>Software Engineer</i>	4,16%	119 jam	Rp 31.250,00	Rp 3.718.750
D	<i>Training & Support</i>	<i>Software Engineer</i>	4,16%	119 jam	Rp 31.250,00	Rp 3.718.750
	Total		20,82%	593 jam		Rp 40.906.250
3	<i>Quality & Testing</i>					
A	<i>Integration Testing</i>	<i>Test Analyst</i>	7,5%	214 jam	Rp 50.000,00	Rp 10.700.000
B	<i>Quality Assurance</i>	<i>Project Manager</i>	8,34%	238 jam	Rp 125.000,00	Rp 29.750.000
C	<i>Evaluation & Testing</i>	<i>Test Analyst</i>	20,84%	594 jam	Rp 50.000,00	Rp 29.700.000
	Total		36,68%	1045 jam		Rp 70.150.000
					Total	Rp 157.843.750

Jadi total estimasi biaya pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang yaitu sebesar **Rp 157.843.750,00**.

5.3 Perhitungan Biaya dengan Metode *Function Point*

Metode *Function Point* merupakan metode perhitungan estimasi biaya dengan memberi bobot pada *input* dan *output*, *logical file*, *external interface file* dan *external inquiry* pada aplikasi untuk mencerminkan tingkat kompleksitas. Perhitungan total untuk semua jenis fungsi disebut perhitungan fungsi, kemudian dimodifikasi dengan menggunakan penilaian dari 14 faktor teknik yang menggambarkan persyaratan sistem dan lingkungan pengembangan sistem (Dalal et al, 2015).

Merujuk pada *Context Diagram* yang terlampir pada Bab 4, menunjukkan *Context Diagram* dari Sistem Antrian RSUD Dr. Soetrasno Rembang. Mendapatkan aliran data sebuah informasi yang lebih lengkap maka dibutuhkan proses dekomposisi dari level *Context Diagram* menjadi *Data Flow Diagram level 1* yang terlampir pada Bab 4 yang kemudian di dekomposisi lagi dari *Data Flow Diagram level 1* ke *Data Flow Diagram level 2*.

5.3.1 Menentukan Tipe Fungsi Pengguna

Langkah pertama yang dilakukan yaitu mengklasifikasikan tipe fungsi pengguna berdasarkan karakteristik kompleksitas yang dimiliki dari hasil analisis terhadap *Data Flow Diagram* pada Sistem Antrian RSUD Dr. Soetrasno Rembang yang terlampir pada Bab 4. Tipe fungsi pengguna yang akan dianalisis yaitu *External Input* (EI), *External Output* (EO), *External Inquiry* (EQ), *Internal Logical File* (ILF) dan *External Interface File* (EIF) (Marthaler, 2005). Tipe fungsi pengguna ada 5 jenis yaitu: 1) *External Input* (EI) merupakan sebuah proses dasar yang melakukan pengolahan data dan mengotrol informasi yang berasal dari luar system; 2) *External Outputs* (EO) merupakan sebuah proses yang melakukan pengiriman data atau informasi yang terlebih dahulu melalui *process logic* atau proses matematik; 3) *External Inquiry* (EQ) merupakan sebuah proses yang melakukan pengiriman data atau informasi diluar dari batas sistem (menampilkan data dilayar *user*); 4) *Internal Logical File* (ILF) merupakan kumpulan data atau informasi yang digunakan dalam aplikasi dan 5) *External Interface Files* (EIF) merupakan berkas yang dibagikan diantara sistem perangkat lunak yang dihitung sebagai tipe berkas antarmuka eksternal dalam setiap perangkat lunak.

Berdasarkan hasil analisis pada *Data Flow Diagram* dari Sistem Antrian RSUD Dr. Soetrasno Rembang, maka menghasilkan 3 fungsi pengguna yaitu *Internal Logical File* (ILF), *External Input* (EI) dan *External Outputs* (EO). Dalam analisis ini menggunakan aturan perhitungan kompleksitas yang terlampir pada Bab 2 dengan hasil tidak ditemukannya *External Inquiry* (EQ) dan *External Interface Files* (EIF). Hasil analisis tipe fungsi pengguna akan disajikan pada Tabel 5.35.

Tabel 5.35 Tipe Fungsi Pengguna Pada Sistem Antrian RSUD Dr. Soetrasno Rembang

Data Function		
<i>Internal Logical File (ILF)</i>	<i>External Input (EI)</i>	<i>External Output (EO)</i>
1. Admin	1. Melakukan Login	1. Autentifikasi User
2. Antrian	2. Tambah Data Admin	2. Lihat Data Admin
3. User	3. Tambah Data User	3. Lihat Data User
4. Inbox	4. Ambil Nomor Antrian	4. Baca Pesan Masuk



5. Outbox 6. Kalimat 7. Video 8. Kuota_SMS	5. Tambah Data Video 6. Tambah Data Running Text	5. Baca Pesan Keluar
---	---	----------------------

5.3.2 Menentukan Bobot Kompleksitas *Function Point*

Setelah menentukan fungsi pengguna pada Sistem Antrian RSUD Dr. Soetrasno Rembang, selanjutnya yaitu menentukan bobot kompleksitas *Function Point* yaitu menentukan FTR, DET dan RET serta kompleksitas *Low*, *Average* atau *High*. FTR, DET dan RET ditentukan berdasarkan hasil analisis pada tipe fungsi pengguna yang disajikan pada Tabel 5.36. Penentuan bobot kompleksitas *Function Point* terdiri dari: 1) *Data Element Type* (DET) yaitu *field* yang tidak berulang dalam sebuah ILF (*Internal Logical File*) dan didefinisikan unik; 2) *Record Element Type* (RET) yaitu sub kelompok data yang berada di dalam sebuah *Logical File*; 3) *File Type Reference* (FTR) yaitu tipe ILF (*Internal Logical File*) atau EIF (*External Logical File*). Penentuan bobot kompleksitas pada Sistem Antrian RSUD Dr. Soetrasno Rembang yang disajikan pada Tabel 5.36.

Tabel 5.36 Penentuan FTR, DET dan RET Sistem Antrian RSUD Dr. Soetrasno Rembang

Fungsi	<i>Data Element Type</i> (DET)	<i>Record Element Type</i> (RET)	<i>File Type References</i> (FTR)	Kompleksitas Fungsional
ILF	ID_admin Nama Password Jabatan Counter ID_antrian_counter ID_antrian No_antrian ID_user No_telepon No_aktivasi DestinationNumber ID_pesan TextDecode SenderNumber ID_Kuota ID_Kalimat ID_Video	ID_admin Nama Jabatan StatusPelayanan Password Counter ID_antrian_counter ID_antrian No_antrian Tipe_antrian Tanggal Status_panggilan Jenis_antrian ID_user No_telepon No_aktivasi Destination_number TextDecode ID_pesan SenderNumber TextDecode ReceivingDateTime ID_pesan ID_Kuota Jumlah_Kuota Tanggal ID_Kalimat Tanggal	-	AVERAGE



Tabel 5.36 Penentuan FTR, DET dan RET Sistem Antrian RSUD Dr. Soetrasno Rembang (Lanjutan)

Fungsi	Data Element Type (DET)	Record Element Type (RET)	File Type References (FTR)	Kompleksitas Fungsional
		Penulis ID_Video Tanggal		
	Total DET 18	Total RET 31	Total FTR 0	
EI	Username, password Data Admin Data User Data Antrian Data Video Data Running Text		Admin User Antrian Video Kalimat	HIGH
	Total DET 6	Total RET 0	Total FTR 5	
EO	Username, password Data Admin Data User Data Pesan Masuk Data Pesan Keluar		Admin User Inbox Outbox	AVERAGE
	Total DET 5	Total RET 0	Total FTR 4	

5.3.3 Perhitungan *Unadjusted Function Point* (UFP)

Pada langkah ini, dilakukan perhitungan nilai *Unadjusted Function Point* (UFP) berdasarkan penentuan bobot kompleksitas pada Tabel 5.21. Tipe fungsi EI mendapatkan bobot kompleksitas *high*, tipe fungsi EO mendapatkan bobot kompleksitas *average* dan tipe fungsi ILF mendapatkan bobot kompleksitas *average*. Penentuan bobot kompleksitas berdasarkan aturan yang terlampir pada Bab 2 pada Tabel 2.7, 2.8 dan 2.9. Perhitungan UFP disajikan pada Tabel 5.37.

Tabel 5.37 Perhitungan *Unadjusted Function Point* (UFP)

Tipe Fungsi Pengguna	Jumlah Tipe Fungsi Pengguna	Perhitungan Bobot Kompleksitas UFP			Bobot Kompleksitas Fungsi Pengguna	Nilai Kompleksitas Fungsi Pengguna	UFP
		DET	RET	FTR			
EI	6	6	-	5	<i>HIGH</i>	6	36
EO	5	5	-	4	<i>AVERAGE</i>	5	25
ILF	8	18	31	-	<i>AVERAGE</i>	10	80
EIF	-	-	-	-	-	-	0
EQ	-	-	-	-	-	-	0
Total UFP							141

Nilai UFP sebesar **141** didapat dari perhitungan total hasil perkalian antara nilai kompleksitas dengan jumlah tipe pengguna.



5.3.4 Perhitungan *Technical Complexity Factor* (TCF)

Pada tahap ini dilakukan penilaian melalui penyebaran lembar penilaian yang telah divalidasi oleh *expert judgement*, perhitungan ini menilai skor yang diisi oleh tim pengembang pada setiap faktor. Dalam pelaksanaan penyebaran lembar penilaian ini melibatkan responden dari tim pengembang Sistem Antrian RSUD Dr. Soetrasno Rembang yang tergabung dalam CV Pabrik Teknologi.

Rekap hasil dan perhitungan pada masing-masing faktor lingkungan yang disajikan pada Tabel 5.38. Perhitungan skor untuk setiap faktor didapat dari hasil pengisian lembar penilaian oleh tim pengembang kemudian dilakukan penjumlahan dan mencari rata-rata dari setiap faktor. Untuk rekap seluruh hasil pengisian lembar penilaian oleh tim pengembang terlampir pada Lampiran D.

Tabel 5.38 Perhitungan *Technical Complexity Factor*

	Faktor	Skor
1	<i>Data Communications</i>	5
2	<i>Distributed Data Processing</i>	4
3	<i>Performance</i>	3
4	<i>Heavily Used Configuration</i>	1
5	<i>Transaction Rate</i>	4
6	<i>Online Data Entry</i>	3
7	<i>End-User Efficiency</i>	4
8	<i>Online Update</i>	3
9	<i>Complex Processing</i>	1
10	<i>Reusability</i>	4
11	<i>Installation Ease</i>	3
12	<i>Operation Ease</i>	1
13	<i>Multiple Sites</i>	3
14	<i>Facilitate Change</i>	3
Total RCAF		42

Selanjutnya, untuk mendapatkan nilai TCF, maka menggunakan persamaan $TCF = 0,65 + 0,01 \times RCAF$. Dimana RCAF merupakan jumlah keseluruhan skor di 14 faktor. Berdasarkan Tabel 5.23, nilai RCAF sebesar **42** maka langkah selanjutnya dapat melakukan perhitungan TCF.

$$\begin{aligned} TCF &= 0,65 + (0,01 \times RCAF) \\ &= 0,65 + (0,01 \times 42) \\ &= \mathbf{1,07} \end{aligned}$$

Setelah melakukan perhitungan dengan persamaan diatas, maka nilai TCF sebesar **1,07**.

5.3.5 Perhitungan Nilai *Function Point* (FP)

Setelah mendapatkan nilai TCF, langkah selanjutnya yaitu menghitung nilai *Function Point* (FP) dengan mengalikan nilai *Unadjusted Function Point* (UFP) dan *Technical Complexity Factor* (TCF). Nilai *Function Point* sebesar **151** yang disajikan pada persamaan berikut ini.



$$\begin{aligned} \text{Function Point} &= \text{UFP} * \text{TCF} \\ &= 141 * 1,07 = 151 \end{aligned}$$

5.3.6 Perhitungan Effort

Setelah mendapatkan nilai *Function Point* (FP), langkah selanjutnya yaitu menghitung *effort* (usaha). Mengetahui *effort* maka menggunakan fungsi estimasi eksponensial oleh Jone's dengan menanyakan langsung kepada *Project Manager* mengenai jenis *software* dan kemampuan atau keterampilan tim pengembang Sistem Antrian RSUD Dr. Soetrasno Rembang yang disajikan pada Tabel 5.39. Berdasarkan Lampiran A, *Project Manager* mengatakan jenis *software* yang dikembangkan termasuk ke dalam *Business* karena *software* dimiliki oleh sebuah instansi yaitu RSUD Dr. Soetrasno Rembang dan kemampuan atau keterampilan tim pengembang termasuk dalam *Average* sehingga nilai *j* yang didapat adalah **0,43**.

Tabel 5.39 Eksponen Jones

Kind of Software	Organization's Skill/Abilities		
	Best In Class	Average	Worst In Class
System	0.43	0.45	0.48
Business	0.41	0.43	0.46
Shrink-wrap	0.39	0.42	0.45

$$\begin{aligned} \text{effort} &= \text{FP}^{3*j} / 27 \\ &= 151^{3*0,43} / 27 \\ &= 24 \text{ person-month} \end{aligned}$$

Kemudian menghitung lama durasi pengerjaan *software* maka menggunakan persamaan f^j . Nilai *s* sebesar **9 bulan**.

$$\begin{aligned} s &= f^j = 151^{0,43} \\ &= 9 \text{ months} \end{aligned}$$

5.3.7 Perhitungan Biaya Total

Kemudian pada langkah selanjutnya melakukan konversi estimasi *effort* dari satuan bulan menjadi satuan hari. Satu bulan terhitung 20 hari kerja, sehingga dilakukan perkalian 9 bulan dengan 20 hari yang menghasilkan **180 hari**. Mengacu pada penelitian Kassem Saleh (2011) maka *effort* (usaha) dibagi menjadi 3 kelompok aktivitas yang disajikan pada Tabel 5.40. Kemudian menghitung nilai *person* pada tiap fase dengan melakukan perkalian antara % *effort* dengan nilai *effort* dan menghitung *days* pada tiap fase dengan melakukan perkalian antara 180 hari dengan % *effort*.

Tabel 5.40 Pembagian Estimasi Effort

	Kelompok Aktivitas	% Effort	Person	Days
1	Software Development			
A	Requirement	7,5%	2 orang	14 hari
B	Spesification	7,5%	2 orang	14 hari
C	Design	10%	2 orang	18 hari

Tabel 5.40 Pembagian Estimasi *Effort* (Lanjutan)

	Kelompok Aktivitas	% <i>Effort</i>	<i>Person</i>	<i>Days</i>
D	<i>Implementation</i>	10%	2 orang	18 hari
E	<i>Acceptance & Deployment</i>	7,5%	2 orang	14 hari
	Total	42,5%	10 orang	77 hari
2	<i>Ongoing Activity</i>			
A	<i>Project Management</i>	8,34%	2 orang	15 hari
B	<i>Configuration Management</i>	4,16%	1 orang	7 hari
C	<i>Documentation</i>	4,16%	1 orang	7 hari
D	<i>Training & Support</i>	4,16%	1 orang	7 hari
	Total	20,82%	5 orang	37 hari
3	<i>Quality & Testing</i>			
A	<i>Integration Testing</i>	7,5%	2 orang	14 hari
B	<i>Quality Assurance</i>	8,34%	2 orang	15 hari
C	<i>Evaluation & Testing</i>	20,84%	5 orang	38 hari
	Total	36,68%	9 orang	66 hari

Langkah selanjutnya yaitu menentukan standar gaji yang diperlukan untuk setiap fase dengan mengacu pada Indonesia *Salary Guide* 2006 yang diterbitkan oleh *Kelly Service*. Dalam penelitian ini, penentuan posisi yang dibutuhkan setiap fase diambil berdasarkan teori yang terlampir pada Bab 2. Kemudian menentukan standar gaji per hari dengan melakukan pembagian antara standar gaji per bulan dengan 20 hari dan hasilnya disajikan pada Tabel 5.41.

Tabel 5.41 Penetapan Standar Gaji

Posisi Dalam <i>Salary Guide</i>	Standar Gaji (per-bulan)	Standar Gaji (per-hari)
<i>Project Manager</i>	Rp 20.000.000,00	Rp 1.000.000,00
<i>System Analyst</i>	Rp 7.000.000,00	Rp 350.000,00
<i>Software Engineer</i>	Rp 5.000.000,00	Rp 250.000,00
<i>Test Analyst</i>	Rp 8.000.000,00	Rp 400.000,00

Pada langkah terakhir, setelah menentukan posisi dalam setiap fase berdasarkan deskripsi yang terlampir pada Bab 2 yang mengacu dengan Indonesia *Salary Guide* 2006 yang diterbitkan oleh *Kelly Service*, maka selanjutnya menghitung biaya per fase dengan melakukan perkalian antara standar gaji per hari dengan *person* dan *days* yang telah ditentukan pada Tabel 5.40. Kemudian melakukan penjumlahan seluruh fase yang menghasilkan estimasi biaya total yang disajikan pada Tabel 5.42.

Tabel 5.42 Perhitungan Total Estimasi Biaya

No	Kelompok Aktivitas	Posisi dalam Salary Guide	Person	Days	Standar Gaji (per-hari) (Rp)	Total (Rp)
1	Software Development					
A	Requirement	System Analyst	2 orang	14 hari	Rp 350.000,00	Rp 9.800.000,00
B	Spesification	System Analyst	2 orang	14 hari	Rp 350.000,00	Rp 9.800.000,00
C	Design	System Analyst	2 orang	18 hari	Rp 350.000,00	Rp 12.600.000,00
D	Implementation	Software Engineer	2 orang	18 hari	Rp 250.000,00	Rp9.000.000,00
E	Acceptance & Deployment	Software Engineer	2 orang	14 hari	Rp 250.000,00	Rp 7.000.000,00
	Total		10 orang	77 hari		Rp 48.200.000,00
2	Ongoing Activity					
A	Project Management	Project Manager	2 orang	15 hari	Rp 1.000.000,00	Rp 30.000.000,00
B	Configuration Management	Software Engineer	1 orang	7 hari	Rp 250.000,00	Rp 1.750.000,00
C	Documentation	Software Engineer	1 orang	7 hari	Rp 250.000,00	Rp 1.750.000,00
D	Training & Support	Software Engineer	1 orang	7 hari	Rp 250.000,00	Rp 1.750.000,00
	Total		5 orang	36 hari		Rp 35.250.000,00
3	Quality & Testing					
A	Integration Testing	Test Analyst	2 orang	14 hari	Rp 400.000,00	Rp 11.200.000,00
B	Quality Assurance	Project Manager	2 orang	15 hari	Rp 1.000.000,00	Rp 30.000.000,00
C	Evaluation & Testing	Test Analyst	5 orang	38 hari	Rp 400.000,00	Rp 76.000.000,00
	Total		9 orang	67 hari		Rp 117.200.000,00
					Total Biaya	Rp 200.650.000,00

Jadi, hasil akhir estimasi biaya pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang yaitu sebesar **Rp 200.650.000,00**.

5.4 Analisis Perhitungan dengan Guesstimate

Analisis perhitungan menggunakan metode *Guesstimate* ini mendeskripsikan durasi pengerjaan, jumlah tim pengembang dan estimasi biaya pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang. Berdasarkan Lampiran A, penentuan harga pengembangan *software* ini menggunakan metode *Guesstimate* atau angka perkiraan yang dirumuskan oleh



Project Manager. Sedangkan untuk perkiraan gaji anggota tim bergantung pada seberapa besar harga proyek yang akan dikembangkan serta berdasarkan berapa *task* yang ada pada setiap fase pengembangan.

Selain itu berdasarkan dokumen *Excel* dari CV Pabrik Teknologi, terdapat alokasi SDM dan *timeline* pengerjaan yang dibutuhkan untuk pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang. Informasi yang diperoleh dari wawancara dengan narasumber dari CV Pabrik Teknologi yaitu mengenai durasi pengerjaan Sistem Antrian RSUD Dr. Soetrasno Rembang, alokasi SDM dan estimasi biaya.

1. Durasi pengembangan Sistem Antrian Dr. Soetrasno Rembang sekitar 2 bulan dimulai pada bulan April 2017 hingga Mei 2017.
2. Anggota tim pengembang *website* Sistem Antrian RSUD Dr. Soetrasno Rembang berjumlah 5 orang.
 - 1) **Project Manager** : Firman Jati Pamungkas, S.Kom
Farid Angga Pribadi, S.Kom
 - 2) **Programmer** : Masrur Anwar
 - 3) **System Analyst** : Komang Candra Brata, S.Kom., M.T., M.Sc.
Lutfi Fanani, S.Kom., M.T., M.Sc

Deskripsi tugas untuk masing-masing anggota tim sebagai berikut.

- 1) **Project Manager**, bertugas melakukan pengawasan terhadap pekerjaan seluruh anggota tim pengembang, mempersiapkan kebutuhan operasional anggota tim pengembang, membuat *report* progres pekerjaan dan menjembatani kepentingan management dan tim pengembang.
 - 2) **Programmer**, melakukan pengembangan *front-end* dan *back-end* sistem serta membuat dokumentasi program.
 - 3) **System Analyst**, menganalisis kebutuhan sistem yang akan dikembangkan dengan mewawancarai *stakeholder*, membuat *flowchart*, *ER Diagram* dan mengawasi serta mengarahkan kerja *programmer*.
3. Estimasi biaya yang dibutuhkan untuk pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang membutuhkan biaya berkisar antara Rp 40.000.000,00 sampai Rp 60.000.000,00 dengan anggota tim berjumlah 5 orang selama 2 bulan.

5.5 Analisis Hasil *Use Case Point*, *Function Point* dan *Guesstimate*

Hasil analisis dari penelitian ini yaitu perbandingan antara metode *Use Case Point*, *Function Point* dan *Guesstimate*. Perbandingan ini berisi data berupa durasi pengerjaan dan biaya total yang dibutuhkan untuk pengembangan Sistem Antrian RSUD Dr. Soetrasno Rembang.

Tabel 5.43 Analisis Hasil *Use Case Point*, *Function Point* dan *Guesstimate*

	<i>Use Case Point</i>	<i>Function Point</i>	<i>Guesstimate</i>
Durasi Pengerjaan	2.850 jam atau 18 bulan	9 bulan	2 bulan
Biaya Total	Rp 157.843.750,00	Rp 200.650.000,00	Rp 40.000.000 sampai Rp 60.000.000



Pada Tabel 5.43 menyajikan perbandingan antara metode *Use Case Point*, *Function Point* dan *Guesstimate* serta terlihat adanya perbedaan durasi pengerjaan dan perbedaan biaya total. Faktor-faktor penyebabnya akan disajikan pada Tabel 5.44 dan Tabel 5.45.

Tabel 5.44 Analisis Perbandingan Durasi Pengerjaan

Metode	Deskripsi
<i>Guesstimate</i> dan <i>Use Case Point</i>	Pada metode <i>Guesstimate</i> membutuhkan durasi pengerjaan lebih cepat yaitu selama 2 bulan sedangkan metode <i>Use Case Point</i> membutuhkan durasi pengerjaan lebih lama yaitu selama 2.850 jam kerja atau setara 18 bulan. Hal ini disebabkan karena <i>Guesstimate</i> tidak membatasi jam kerja pada setiap fase, lain halnya dengan metode <i>Use Case Point</i> membatasi jam kerja hanya 8 jam per hari.
<i>Guesstimate</i> dan <i>Function Point</i>	Pada metode <i>Guesstimate</i> membutuhkan durasi pengerjaan lebih cepat yaitu selama 2 bulan sedangkan metode <i>Function Point</i> membutuhkan durasi pengerjaan lebih lama yaitu selama 9 bulan. Hal ini disebabkan karena <i>Guesstimate</i> memiliki <i>deadline</i> pengerjaan dan tidak ada pertimbangan jenis <i>software</i> dan pengalaman tim pengembang sedangkan pada metode <i>Function Point</i> mempertimbangkan jenis <i>software</i> dan seberapa lama pengalaman tim berdasarkan teori <i>Jone's First Order</i> yaitu Sistem Antrian Dr. Soetrasno Rembang termasuk <i>Business Software</i> karena dimiliki oleh instansi dengan pengalaman tim <i>average</i> . Penilaian tersebut berdasarkan hasil wawancara dengan <i>Project Manager</i> yang terlampir pada Lampiran A.
<i>Use Case Point</i> dan <i>Function Point</i>	Pada metode <i>Function Point</i> membutuhkan durasi pengerjaan lebih cepat yaitu selama 9 bulan sedangkan metode <i>Use Case Point</i> membutuhkan durasi pengerjaan selama 2.850 jam atau setara 18 bulan. Hal ini disebabkan karena metode <i>Use Case Point</i> menggunakan dua jenis lembar penilaian yang memiliki indikator bernilai negatif sehingga mempengaruhi durasi pengerjaan seperti pada indikator <i>Part Time Staff</i> , tim pengembang memberikan skor tinggi sehingga nilai negatif semakin tinggi sedangkan metode <i>Function Point</i> hanya menggunakan satu jenis lembar penilaian.

Tabel 5.45 Analisis Perbandingan Estimasi Biaya

Metode	Deskripsi
<i>Guesstimate</i> dan <i>Use Case Point</i>	Pada metode <i>Guesstimate</i> membutuhkan estimasi biaya dengan nominal yang lebih kecil yaitu berkisar antara Rp 40.000.000,00 sampai Rp 60.000.000,00 sedangkan metode <i>Use Case Point</i> membutuhkan estimasi biaya dengan nominal yang lebih besar yaitu sebesar Rp 157.843.750,00. Hal ini disebabkan karena pada <i>Guesstimate</i> melakukan penentuan gaji per orang berdasarkan nilai proyek, misalnya Programmer sebesar Rp 3.000.000,00 sedangkan <i>Function Point</i> menggunakan standar gaji dari Indonesia <i>Salary Guide</i> yang diterbitkan oleh <i>Kelly Service</i> yaitu sebesar



Tabel 5.45 Analisis Perbandingan Estimasi Biaya (Lanjutan)

Metode	Deskripsi
	5.000.000,00 per orang.
<i>Guesstimate</i> dan <i>Function Point</i>	Pada metode <i>Guesstimate</i> membutuhkan estimasi biaya dengan nominal yang lebih kecil yaitu berkisar antara Rp 40.000.000,00 sampai Rp 60.000.000,00 sedangkan metode <i>Function Point</i> membutuhkan estimasi biaya dengan nominal yang lebih besar yaitu sebesar Rp 200.650.000,00. Hal ini disebabkan karena pada metode <i>Guesstimate</i> membutuhkan SDM yang lebih sedikit dibanding metode <i>Function Point</i> . Bisa dilihat pada fase <i>Implementation</i> , <i>Guesstimate</i> membutuhkan SDM sebanyak 1 orang sedangkan <i>Function Point</i> membutuhkan 2 orang. Selain itu, <i>Guesstimate</i> melakukan penentuan gaji per orang berdasarkan nilai proyek misalnya Programmer sebesar Rp 3.000.000,00, sedangkan <i>Function Point</i> menggunakan standar gaji dari Indonesia <i>Salary Guide</i> yang diterbitkan oleh <i>Kelly Service</i> yaitu sebesar Rp 5.000.000,00 per orang.
<i>Use Case Point</i> dan <i>Function Point</i>	Pada metode <i>Use Case Point</i> membutuhkan estimasi biaya dengan nominal yang lebih kecil yaitu sebesar Rp 157.843.750,00 sedangkan metode <i>Function Point</i> membutuhkan estimasi dengan nominal yang lebih besar yaitu sebesar Rp 200.650.000,00. Hal ini disebabkan karena metode <i>Function Point</i> dapat menentukan kebutuhan SDM pada setiap fase sehingga jika di total maka menghasilkan estimasi biaya yang lebih besar. Lain halnya dengan metode <i>Use Case Point</i> yang diasumsikan bahwa di setiap fase membutuhkan 1 orang.

BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian ini, dalam manajemen proyek Sistem Antrian RSUD Dr. Soetrasno Rembang maka dapat ditarik kesimpulan yaitu:

1. Tahapan pengerjaan Sistem Antrian RSUD Dr. Soetrasno Rembang dengan pendekatan *Work Breakdown Structure* (WBS) menghasilkan dua WBS yaitu WBS *Guesstimate* dan WBS *Use Case Point* dan *Function Point*. WBS *Guesstimate* memiliki empat level, level pertama Sistem Antrian RSUD Dr. Soetrasno Rembang, level dua terdiri dari fase *Define Project Goal*, *Plan Project* dan *Execute Project Plan*, level ketiga terdiri dari fase *Software Development* dan level keempat terdiri dari fase *Planning*, *Analysis*, *Design*, *Implementation*, *Maintenance and Support*. Sedangkan WBS *Use Case Point* dan *Function Point* memiliki empat level, level pertama Sistem Antrian RSUD Dr. Soetrasno Rembang, level dua terdiri dari fase *Define Project Goal*, *Plan Project* dan *Execute Project Plan*, level ketiga terdiri dari *Software Development*, *Ongoing Activity*, *Quality and Testing*, level keempat terdiri dari fase *Requirement*, *Spesification*, *Design*, *Implementation*, *Acceptance and Deployment*, *Project Management*, *Configuration Management*, *Documentation*, *Training and Support*, *Integration Testing*, *Quality Assurance*, *Evaluation and Testing*.
2. Hasil penjadwal pengerjaan proyek Sistem Antrian RSUD Dr. Soetrasno Rembang menggunakan *Gantt Chart* berdasarkan *Work Breakdown Structure* (WBS) menghasilkan *Gantt Chart* untuk *Guestimate* dengan tiga fase: 1) *Define Project Goal*; 2) *Plan Project*; 3) *Execute Project Plan*. Pada fase *Execute Project Plan* dibagi menjadi satu fase pekerjaan yaitu *Software Development* membutuhkan 5 orang, 71 hari dan biaya berkisar antara Rp 40.000.000,00 sampai Rp 60.000.000,00. Sedangkan *Gantt Chart* untuk metode *Use Case Point* dengan tiga fase: 1) *Define Project Goal*; 2) *Plan Project*; 3) *Execute Project Plan*. Pada fase *Execute Project Plan* dibagi menjadi tiga fase yaitu *Software Development* membutuhkan *System Analyst* dan *Software Engineer*, 151 hari pengerjaan dan biaya Rp 46.787.500, *Ongoing Activity* membutuhkan *Project Manager* dan *Software Engineer*, 74 hari pengerjaan dan biaya Rp 40.906.250 dan *Quality and Testing* membutuhkan *Test Analyst* dan *Project Manager*, 132 hari pengerjaan dan biaya Rp 70.150.000. Sedangkan *Gantt Chart* untuk metode *Function Point* dengan tiga fase: 1) *Define Project Goal*; 2) *Plan Project*; 3) *Execute Project Plan*. Pada fase *Execute Project Plan* dibagi menjadi tiga fase yaitu *Software Development* membutuhkan 10 orang, 77 hari pengerjaan dan biaya Rp 48.200.000, *Ongoing Activity* membutuhkan lima orang, 36 hari pengerjaan dan biaya Rp 35.250.000 dan *Quality and Testing* membutuhkan sembilan orang, 67 hari pengerjaan dan biaya Rp 117.200.000.
3. Hasil penerapan metode *Use Case Point* menghasilkan keluaran berupa *hours of effort* selama 8.950 jam kerja dan biaya total sebesar Rp 157.843.750,00.
4. Hasil penerapan metode *Function Point* menghasilkan keluaran berupa estimasi *effort* (usaha) yaitu 24 orang dengan durasi pengerjaan selama 9 bulan dan biaya total sebesar Rp 200.650.000,00.
5. Pada durasi pengerjaan menggunakan metode *Guesstimate* memerlukan waktu yang lebih singkat yaitu selama 3 bulan, sedangkan metode *Function Point* memerlukan



waktu selama 9 bulan dan metode *Use Case Point* memerlukan waktu paling lama yaitu selama 2.850 jam kerja atau setara 18 bulan. Pada estimasi biaya menggunakan metode *Guesstimate* membutuhkan biaya dengan nominal yang lebih kecil yaitu berkisar antara Rp 40.000.000,00 sampai Rp 60.000.000,00. Sedangkan metode *Use Case Point* menghasilkan biaya sebesar Rp 157.843.750,00 serta metode *Function Point* menghasilkan biaya dengan nominal yang paling besar yaitu Rp 200.650.000,00.

6.2 Saran

1. Pengembangan perhitungan estimasi biaya menggunakan *Function Point* selanjutnya, diharapkan menggunakan submodel selain *post architecture* agar lebih memahami metode *Function Point* dan apabila menggunakan submodel *post architecture* diharapkan dapat menggunakan perhitungan jumlah dari *Line of Code*.
2. Pada penelitian estimasi biaya menggunakan metode *Use Case Point* dan/atau metode *Function Point* selanjutnya, diharapkan menggunakan standar gaji lain selain *Kelly Service*.
3. Pada penelitian estimasi biaya menggunakan metode *Use Case Point* dan/atau metode *Function Point* selanjutnya, diharapkan menggunakan *Guideline* Estimasi Biaya yang lain.

DAFTAR PUSTAKA

- Azwar, S., 2015. *Reliabilitas dan Validitas* (Edisi Empat). Yogyakarta: Pustaka Pelajar.
- Bittner, et al., 2002. *Use Case Modelling*. Addison Wesley.
- Bredemeyer, et al., 2001. *Functional Requirements and Use Cases*. Bredemeyer Consulting.
- Connel, Mc., 1996. *Rapid Development*. Washington: Microsoft Press.
- Connel, Mc., 2006. *Software Estimation: Demystifying Black Art*. Washington: Microsoft Press.
- Dalal, et al., 2015. *Analysis of Software Estimation Method: Function Point and Use Case Point*. India: Shrinthji Institute of Technology and Engineering.
- Daniari, Imania, 2013. *Perkiraan Biaya Pembuatan Enterprise Resource Planning (ERP) Untuk Unit Bisnis Pabrik Gula Pada PT Perkebunan XYZ Dengan Metode Function Point*. S1. Institut Teknologi Sepuluh Nopember.
- Fan, et al., 2009. *Software Cost Estimation*. Hongkong: The Hongkong Polytechnic University.
- Farr, 2007. *Computer and Technical Careers*. JIST Publishing, Inc.
- Johri, et al., *Use Case Point Estimation Technique in Software Development*. India: Amity University Uttar Pradesh.
- Kelly Service, 2016. *Indonesia Employment Salary Outlook and Salary Guide*. Indonesia: Kelly Service inc.
- Kendall, et al., 2014. *Systems Analysis and Design* (Vol. 19, p. 02). Year Prentice Hall.
- Ribu, K., 2001. *Estimating Object Oriented Software Project with Use Case Point*. Thesis. University Osloensis.
- Kusrini, et al., 2015. *Implementasi Metode Function Point untuk Mengukur Volume Software*. Yogyakarta: STMIK AMIKOM Yogyakarta.
- Leung, et al., 2002. *Software Cost Estimation*. Hong Kong: The Hong Kong Polytechnic University.
- Li, et al., 2009. *A Study of Project Selection and Feature Weighting for Analogy Based Software Cost Estimation*. Singapore: National University of Singapore.
- Longstreet, D., 2005. *Function Point Analysis Training Course*. Longstreet Consulting Inc.
- Marchewka, J., 2003. *Information Technology Project Management*.
- Marthaler, et al., 2005. *Function Point Counting Practice Manual*. The International Function Point Users Group.
- Momoh et al., 2008. *A Work Breakdown Structure for Implementing and Costing an ERP Project*.
- Noerlina, 2008. *Perencanaan Manajemen Proyek Sistem Informasi dan Teknologi Informasi Online Bisnis*. Jakarta Barat: Universitas Bina Nusantara.
- Ochodek, et al., 2011. *Improving the Reability of Transaction Identification in Use Case*. Poland: Poznan University of Technology.



Prassida, Frieska G., 2015. *Estimasi Biaya Pembuatan Modul Enterprise Resource Planning (ERP) Untuk Unit Bisnis Pabrik Gula di PT Perkebunan XYZ Dengan Metode Use Case Point*. S1. Institut Teknologi Sepuluh Nopember.

Pressman, S., 2010. *Software Engineering: A Practitioner's Approach (7th edition)*. New York: McGraw-Hill.

Recker, et al., 2013. *Scientific Research in Information Systems*. Springer-Verlag.

Rev, 2003. *Work Breakdown Structure (Office of Engineering and Construction Management ed)*. United State of America.

Saleh, K. 2011., *Effort and Cost Allocation in Medium to Large Software Development Projects*. Internasional Journal of Computers(I).

Sarwono, Jonathan., 2006. *Metode Penelitian Kuantitatif Dan Kualitatif (Edisi Pertama)*. Yogyakarta: Graha Ilmu.

Schwalbe, Kathy, 2014. *Information Technology Project Management (7th edition)*. United State of America: Augburgs College.

Shalahuddin, et al., 2014. *Rekayasa Perangkat Lunak Terstruktur dan berorientasi objek*. Edisi 2. Bandung: Informatika

Sholih, et al., 2013. *Effort Distribution to Estimate Cost in Small to Medium Software Development Project with Use Case Point*. Surabaya: Institut Teknologi Sepuluh Nopember.

Sommerville, Ian. 2011. *Software Engineering (9th edition ed.)*. [pdf] Addison-Wesley.

Thomas, et al., 2015. *Data Flow Diagramming by Example*. BA Experts.

Yunautama, 2008. *Proyek Pengembangan Sistem Informasi Kepegawaian PT Multi Area Conindo*. Bandung: OMG IT & Management Consultants.

Zawari, A., 2011. *Project Estimation with Use Case Point*.