

**PERHITUNGAN BIAYA PROYEK SISTEM INFORMASI REKAM
MEDIS DENGAN MENGGUNAKAN METODE COCOMO II
(Studi Kasus: 7Treesdigital)**

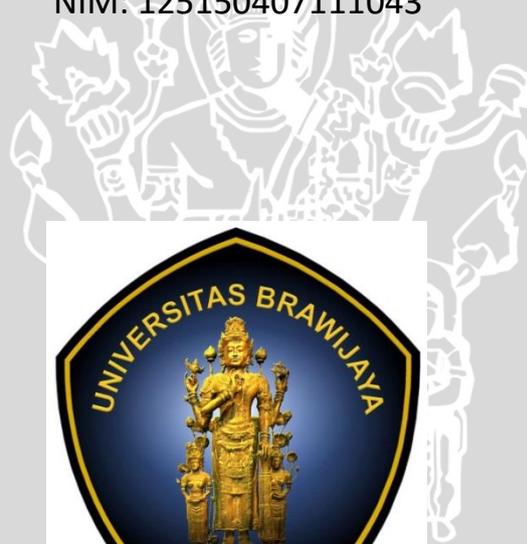
SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Aldi Yudha Maswinandar

NIM: 125150407111043



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

PERHITUNGAN BIAYA PROYEK SISTEM INFORMASI REKAM MEDIS DENGAN
MENGUNAKAN METODE COCOMO II (Studi kasus: 7Treesdigital)

SKRIPSI

Untuk memenuhi sebagian persyaratan
mencapai gelar Sarjana Komputer

Disusun Oleh :

ALDI YUDHA MASWINANDAR
125150407111043

Skripsi ini telah diuji dan dinyatakan lulus pada
10 Agustus 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

M. Chandra Saputra, S.Kom., M.Eng

Niken Hendrakusma W, S.Kom. M.Kom

NIK: -

NIK: 2016069006212001

Mengetahui
Ketua Jurusan Sistem Informasi

Herman Tolle, Dr. Eng., S.T, M.T

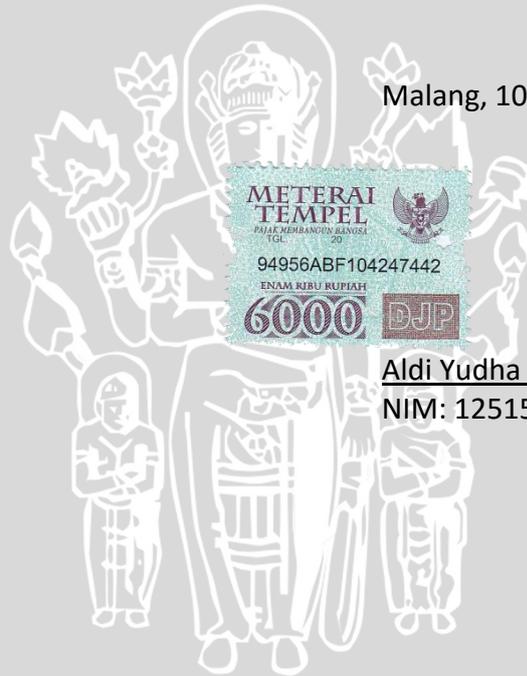
NIP: 19740823 200012 1 00

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, 10 Agustus 2016



Aldi Yudha Maswinandar
NIM: 125150407111043

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, yang telah memberikan segala rahmat dan hidayah-Nya, serta atas junjungan Nabi Muhammad SAW, sehingga penulis diberi kemudahan, kesabaran, ketabahan, kekuatan serta mendapatkan hikmah terbaik dalam menyelesaikan skripsi yang berjudul “PERHITUNGAN BIAYA PROYEK SISTEM INFORMASI REKAM MEDIS DENGAN MENGGUNAKAN METODE COCOMO II (Studi Kasus: 7Tressdigital)” dengan baik dan benar, yang disusun guna memenuhi salah satu syarat untuk menyelesaikan tugas akhir pada program studi Sistem Informasi dan mencapai gelar Sarjana Komputer di Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

Laporan skripsi ini disusun setelah melakukan survey beberapa software house yang berada di kota Malang. terselesaikannya laporan skripsi ini tidak lepas dari dukungan banyak pihak yang telah membantu. Oleh karena itu, pada kesempatan ini dengan segala kerendahan hati, perkenankanlah penulis mengucapkan terima kasih kepada:

1. Orang tua penulis, Bapak Darmadi Ariyadi dan Ibu Wiwid Widyawati, adik penulis Ervio Adhewiyaksa dan Adelia Ramadhanti, serta keluarga besar penulis yang tercinta, yang telah memberikan dorongan baik secara moril maupun materil, menjadi sumber inspirasi bagi penulis untuk terus semangat dalam mengejar cita-cita, serta doa yang selalu dipanjatkan untuk penulis disetiap sujudnya.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D., Bapak Heru Nurwarsito, Ir., M.Kom., Bapak Marji, Drs., M.T., dan Bapak Edy Santoso, S.Si, M.Kom. selaku Dekan, Wakil Dekan I, Wakil Dekan II, dan Wakil Dekan III, Fakultas Ilmu Komputer, Universitas Brawijaya.
3. Bapak Suprpto, S.T, M.T. dan Bapak Ismiarta Aknuranda, ST., M.Sc., Ph.D selaku Ketua dan Sekretaris Program Studi Sistem Informasi Fakultas Ilmu Komputer, Universitas Brawijaya.
4. Bapak M. Chandra Saputra, S.Kom., M.Eng selaku Dosen Pembimbing I dan Ibu Niken Hendrakusuma Wardhani S.kom., M.Kom selaku Dosen Pembimbing II skripsi yang telah bersedia banyak meluangkan waktu dan memberikan bimbingan, saran dan nasihat maupun motivasi sehingga penulisan skripsi ini dapat terselesaikan dengan baik dan tepat waktu.
5. Seluruh Civitas Akademika Fakultas Ilmu Komputer, Universitas Brawijaya, Malang yang telah berkenan menyalurkan semua ilmu pengetahuan saat penulis aktif mengikuti perkuliahan, sehingga dengan modal ilmu pengetahuan tersebut penulis dapat mengerjakan skripsi ini dan siap terjun ke masyarakat.
6. Seluruh pemain PB. Filkom, Bapak Ismiarta Aknuranda, Bapak Adharul Muttaqin, Mas Joko, Mas Efendi, Mas Dhika Wirya, Mas Andito, Mas Putra yang selalu memberikan saran dan motivasi kepada penulis saat berada dilapangan.
7. Bapak Yugo Yudansyah, selaku CEO dari software house 7Tressdigital yang bersedia membantu penulis dalam pengumpulan data skripsi.

8. Rahma Sischa Imansari, selaku teman hidup yang selalu memberikan support kepada penulis serta memberikan motivasi-motivasi.
9. Keluarga kecil dikota perantauan yaitu keluarga JNCKS yang penulis sayangi, M. Bagus Setiawan, Prasetya M. Aditya, Tommy Krisna Permadi, Adi Nugroho, Ardhi Hanifa Rahmansyah, Andre Firza Setyananda, Ishak Novianus H, Abraham Dwi Asmoro, Fadly Yoga Pratama, Afandi Ahmad Syaikhu, Dimas Hamdan, Agung Baskoro Wibisono, Rahma Sischa Imansari, Komang Ayu Tresna Pratiwi, Dyah Ayu Widyastuti, Dewi Nuraini W, Astri Dyahaloka, Ernita Oktaviana Safitri dan Istidyah yang telah memberikan bantuan dan dukungan selama proses pengerjaan skripsi ini serta telah menjadi sahabat dan keluarga seperjuangan.
10. Keluarga BIOS Extend yaitu Mas Wahyu Sugih Prabowo, Mas Mirza Redhi, Mas Jhon, Mas Abi, Mas Viel, Mbak Betty Anggoro, Mbak Elliya, Mbak Yasmin, Mbak Elok yang telah mengajarkan penulis tentang berorganisasi didalam kampus dan bersedia menjadi kakak dari penulis yang selalu memberikan motivasi-motivasi.
11. Keluarga BIOS Cloud, Danu Akbar, Yuyon alif, Argo Wahyu, Cicha Utami, Rezqi Andini, Febry A.Indaswari, Nancy, Qoirul Kotimah, Sifasani dan staff-staff BIOS Cloud yang telah memberikan support dan do'a untuk penulis.
12. Keluarga Besar Mahasiswa Sistem Informasi (KBMSI), khususnya angkatan 2012, terimakasih atas semua kerjasamanya, bantuannya, pengalamannya, dukungannya, dan doanya, selama pengerjaan skripsi ini maupun selama penulis menempuh bangku perkuliahan.
13. Teman-teman Fakultas Ilmu Komputer dan Universitas Brawijaya, serta semua yang telah memberi ilmu, dukungan dan motivasi kepada penulis.
14. Serta semua pihak yang telah membantu secara langsung maupun tidak langsung, dan semua pihak yang telah berjasa dalam penyelesaian laporan skripsi ini yang penulis hormati dan banggakan.

Demikianlah yang dapat penulis sampaikan, dengan doa dan harapan semoga segala kebaikan yang telah diberikan kepada penulis mendapatkan balasan yang terbaik oleh Allah SWT. Penulis menyadari sepenuhnya dalam penyusunan laporan skripsi ini, masih jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran serta penilaian yang bersifat membangun dari semua pihak guna menyempurnakan hasil laporan skripsi ini. Semoga laporan skripsi ini dapat bermanfaat bagi pembaca secara umum dan penulis secara khusus. Akhir kata penulis mengucapkan banyak terima kasih.

Malang, 10 Agustus 2016

Aldi Yudha Maswinandar
Yudha554@gmail.com

ABSTRAK

Komponen manajemen proyek antara lain ruang lingkup (*scope*), waktu (*time*), dan biaya (*cost*), 3 aspek tersebut yang menjadi faktor dalam keseimbangan suatu proyek. Dalam setiap pengerjaan sebuah proyek alangkah baiknya menggunakan pengerjaan yang terstruktur dengan mengacu pada dokumen *Work Breakdown Structure* (WBS). Agar proses inialisasi sebuah proyek dapat berjalan hingga akhir dengan maksimal harus adanya pembagian disetiap fase pengerjaannya yang terkelola dengan baik. Adapun Masalah yang sering terjadi pada *software house* adalah dalam pengembangan suatu proyek sebuah organisasi tidak mementingkan manajemen proyek dengan benar, padahal dengan adanya manajemen proyek yang benar dapat membantu organisasi tersebut dalam hal estimasi SDM, biaya proyek hingga durasi proyek.

Dengan adanya masalah seperti yang dijelaskan, penelitian ini mengangkat judul tentang Perencanaan Biaya Proyek Sistem Informasi Dengan Metode COCOMO II (Studi Kasus: 7Treesdigital) yang nantinya akan membantu meminimalisir dari masalah tersebut. Dari judul tersebut penelitian ini akan menjabarkan dari setiap pengerjaan proyek dengan menggunakan dokumen WBS agar nantinya setiap pengerjaan proyek dapat terkontrol dengan baik. Kemudian menggunakan metode COCOMO II untuk menghitung usaha pengembangan perangkat lunak sebagai fungsi ukuran program dan serangkaian pengendali biaya yang menyangkut penilaian subyektif terhadap produk, perangkat keras, personil dan atribut proyek. Perhitungan estimasi dalam metode COCOMO II menggunakan perhitungan nilai *unadjusted function point* dari sistem rekam medis (SIRIS).

Analisis bobot kompleksitas dari sistem SIRIS didapatkan dari analisis *context diagram* dari proyek tersebut. Untuk menghitung besarnya usaha menggunakan proses perhitungan dari analisis *function point* dari *data flow diagram* dari sistem. Sedangkan, perhitungan *scale factor* dan *effort adjustment multipliers* menggunakan kuesioner yang disebarkan kepada tim pengembang proyek rekam medis tersebut. *person-month* yang didapatkan nantinya akan dikalikan nilai gaji yang mengacu pada standar gaji UMR kabupaten malang. Hasil dari penelitian ini berupa rekomendasi alokasi sumber daya, biaya proyek, serta durasi waktu pengerjaan suatu proyek. Penelitian ini menghasilkan perbandingan antara metode COCOMO II dan *guesstimate* yang meliputi estimasi sdm, waktu proyek dan biaya proyek.

Kata kunci: *Work Breakdown structure, COCOMO II, Unadjusted function points, Context diagram, Data flow diagram, Effort adjustment multipliers, Scale Factor, Person Month, UMR kabupaten Malang.*

ABSTRACT

Components of the project management, among others, the scope (scope), time (time), and cost (cost), the third aspect is a factor in the balance of a project. In each work on a project it is advisable to use workmanship structured with reference to the document Work Breakdown Structure (WBS). In order for a project initialization process can run until the end with the division must have at every phase of the process is managed. The problem that often occurs at the software house is in the development of a project of an organization is not concerned with project management correctly, but with the right project management can help these organizations in terms of human resource estimation, project costs up to the duration of the project. With the problem as described, this study raised the title of Project Cost Planning Information System with COCOMO II Method (Case Study: 7Treesdigital) which will help minimize these problems.

The title of this study will describe each project with WBS menggunakan document so that later each project can be well controlled. Then using the method for calculating the COCOMO II software development effort as a function of the size of the program and a series of charge controllers concerning the subjective assessment of the product, hardware, personnel and project attributes. Calculation of estimates in COCOMO II method using calculation of unadjusted function point of medical records system (SIRIS).

Analysis weights complexity of SIRIS system context diagram obtained from the analysis of the project. To calculate the size of the business using the process of calculation of function point analysis of the data flow diagram of the system. Meanwhile, scale factor calculation and adjustment effort multipliers using a questionnaire distributed to the project development team of the medical record. person-month obtained will be multiplied by the value of the salary which refers to the unfortunate city minimum wage standards. The results of this research is on resource allocation, project costs, as well as the duration of time spent on a project. This research resulted in the comparison between the methods COCOMO II and guesstimate that includes estimates of tablespoons, project time and cost of the project.

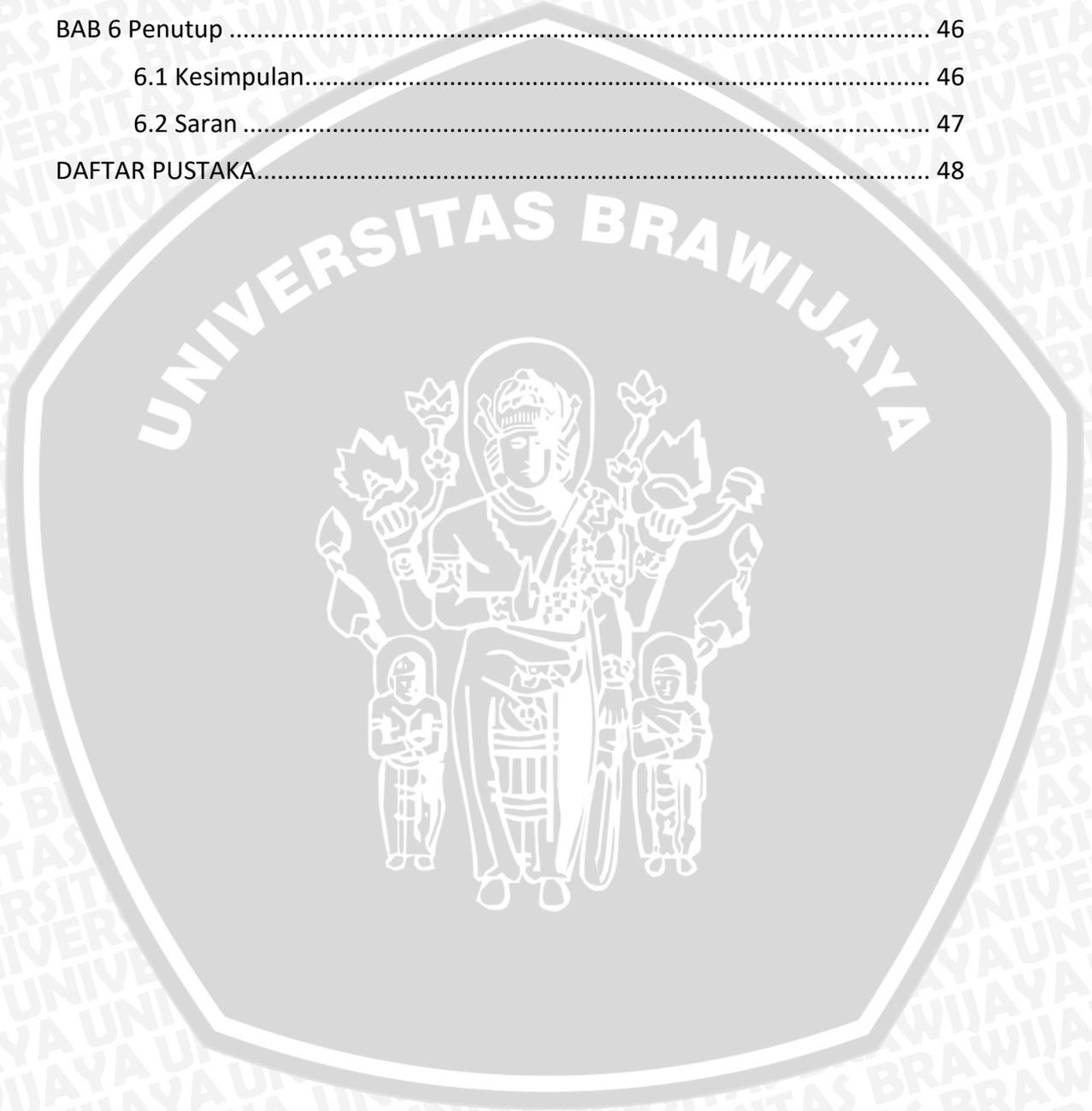
Keywords: work breakdown structure, COCOMO II, unadjusted function points, Context diagrams, data flow diagrams, Effort adjustment multipliers, Scale Factor, Person Month, UMR Kabupaten Malang.

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	viii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 kajian Pustaka	5
2.2 Profil 7Tressdigital	6
2.2.1 Visi 7Tressdigital.....	6
2.2.2 Misi 7Tressdigital.....	6
2.3 Manajemen Proyek	6
2.3.1 Pengertian Manajemen.....	6
2.3.2 Pengertian Proyek.....	6
2.3.3 Pengertian Manajemen Proyek	6
2.4 Work Breakdown Structure.....	7
2.5 Metode COCOMO II.....	7
2.5.1 Metode COCOMO II.....	7
2.5.2 Submodel COCOMO II	9
2.5.3 Unadjusted Function Point.....	10

2.5.3.1 External Input (EI)	11
2.5.3.2 External Output (EO).....	11
2.5.3.3 Internal Logical Files (ILF).....	11
2.5.3.4 External Interface Files (EIF)	12
2.5.4 Scale Factor.....	13
2.5.5 Effort Adjustment Multiplier	15
2.5.6 Estimasi Usaha (Person-Month).....	18
2.5.7 Estimasi Biaya	18
2.5.8 Data Flow Diagram	18
2.6 Guesstimate.....	21
BAB 3 METODOLOGI	22
3.1 Studi Pustaka.....	22
3.2 Pengumpulan Data	23
3.2.1 Dokumen Perancangan	23
3.2.2 Wawancara.....	23
3.2.3 Kuesioner.....	23
3.3 Analisis Hasil.....	24
3.4 Kesimpulan.....	25
BAB 4 PENGUMPULAN DATA	26
4.1 Mekanisme Observasi.....	26
4.2 Hasil Obsesi.....	26
4.3 Mekanisme Kuesioner	28
4.4 Hasil Kuesioner	28
4.4 Dokumen Perancangan.....	28
BAB 5 ANALISIS DAN PERHITUNGAN	34
5.1 Perhitungan <i>Function Point</i>	34
5.1.1 Analisis Perhitungan Total <i>Unadjusted Function Point</i> SIRIS.....	34
5.1.2 Konversi Hasil <i>Unadjusted Function Point</i> menjadi <i>Kilo Line of Code</i>	38
5.2 Perhitungan <i>Scale Factor</i>	39
5.3 Perhitungan <i>Effort Adjusment Multiplier</i>	39
5.4 Perhitungan Estimasi Usaha (<i>Person-Month</i>).....	41

5.5 Perhitungan Estimasi Biaya.....	41
5.6 Analisis Perhitungan dengan <i>Guesstimate</i>	42
5.7 Analisis Perhitungan dengan COCOMO II.....	43
5.8 Analisis Hasil COCOMO II dan <i>Guesstimate</i>	44
BAB 6 Penutup	46
6.1 Kesimpulan.....	46
6.2 Saran	47
DAFTAR PUSTAKA.....	48

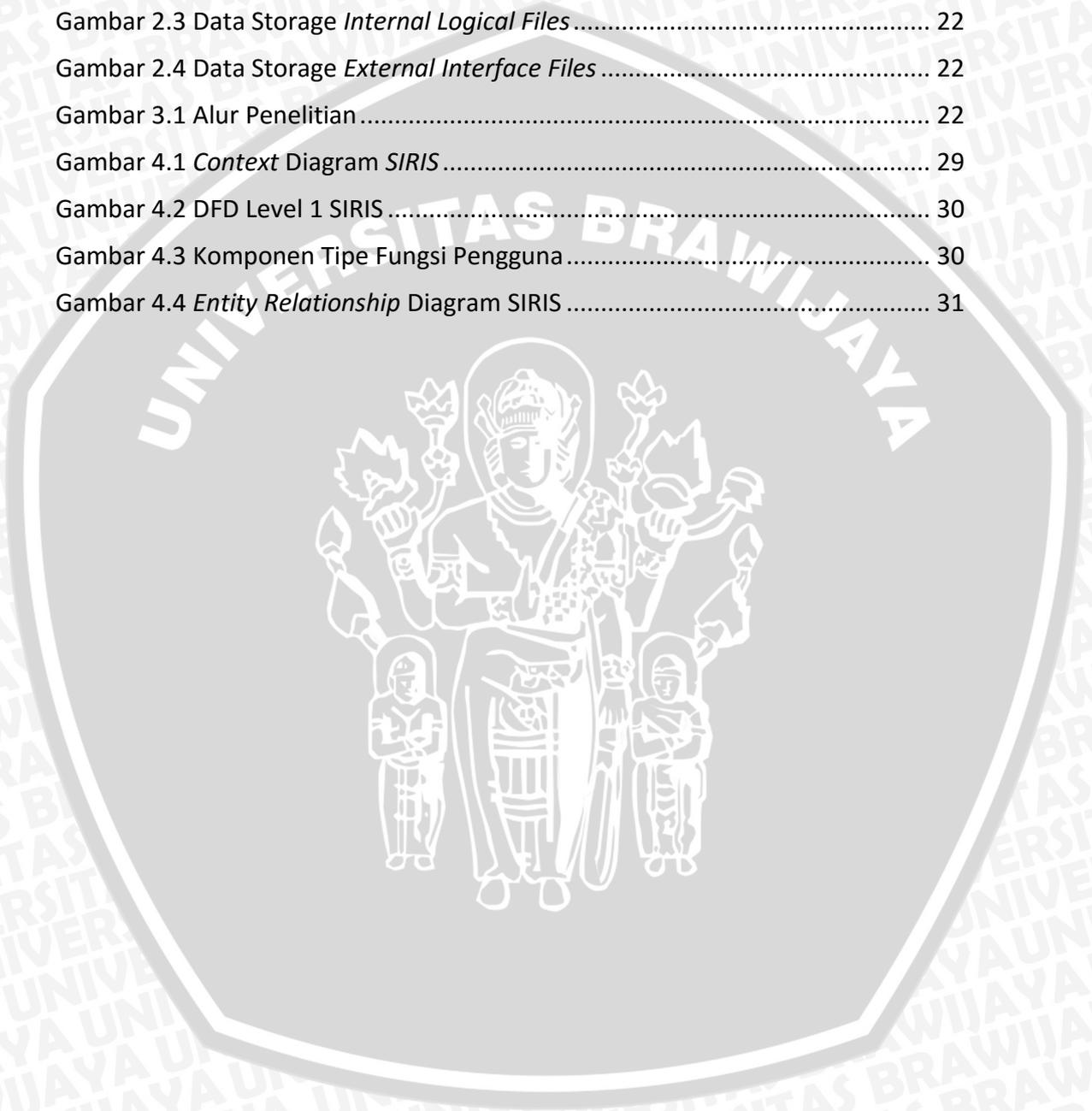


DAFTAR TABEL

Tabel 2.1 Tipe Fungsi Pengguna.....	10
Tabel 2.2 Bobot Kompleksitas Fungsi Pengguna	12
Tabel 2.3 DET, RET dan FTR.....	12
Tabel 2.4 Bobot Kompleksitas <i>Unadjusted Function Point</i> (UFP)	13
Tabel 2.5 QSM SLOC.....	14
Tabel 2.6 Definisi Scale Factor	15
Tabel 2.7 Bobot <i>Scale Factor</i> COCOMO II submodel <i>post architecture</i>	16
Tabel 2.8 Pengelompokan Faktor Biaya.....	16
Tabel 2.9 <i>Effort Adjustment Multipliers</i>	16
Tabel 2.10 Bobot <i>Effort Multipliers</i> COCOMO II <i>Post Architecture</i>	18
Tabel 2.11 Notasi <i>Data Flow Diagram</i>	20
Tabel 2.12 Notasi <i>Entity Relationship Diagram</i>	21
Tabel 3.1 Alur Perhitungan COCOMO II	25
Tabel 4.1 Pengguna Pasien	27
Tabel 4.2 Pengguna Dokter.....	27
Tabel 4.3 Pengguna Admin	28
Tabel 4.4 Pengguna Dokter dan Admin	28
Tabel 4.5 Pengguna Dokter dan Pasien	28
Tabel 4.6 Pengguna Dokter, Pasien dan Admin	28
Tabel 4.7 Pengguna Dokter, Pasien, Admin, dan Pengunjung.....	29
Tabel 5.1 Pengelompokan Tipe Fungsi Pengguna (ILF).....	36
Tabel 5.2 Pengelompokan Tipe Fungsi Pengguna (EI)	37
Tabel 5.3 Pengelompokan Tipe Fungsi Pengguna (EO).....	38
Tabel 5.4 Penentuan bobot RET, DET dan FTR pada UFP	29
Tabel 5.5 Perhitungan <i>Unadjusted Function Point</i> (UFP).....	42
Tabel 5.6 Hasil Perhitungan <i>Function Point</i>	42
Tabel 5.7 Hasil Penilaian <i>Scale Factor</i>	43
Tabel 5.8 Hasil Penilaian <i>Effort Adjustment Multiplier</i>	43
Tabel 5.9 Analisis Hasil COCOMO II dan Guesstimate	48

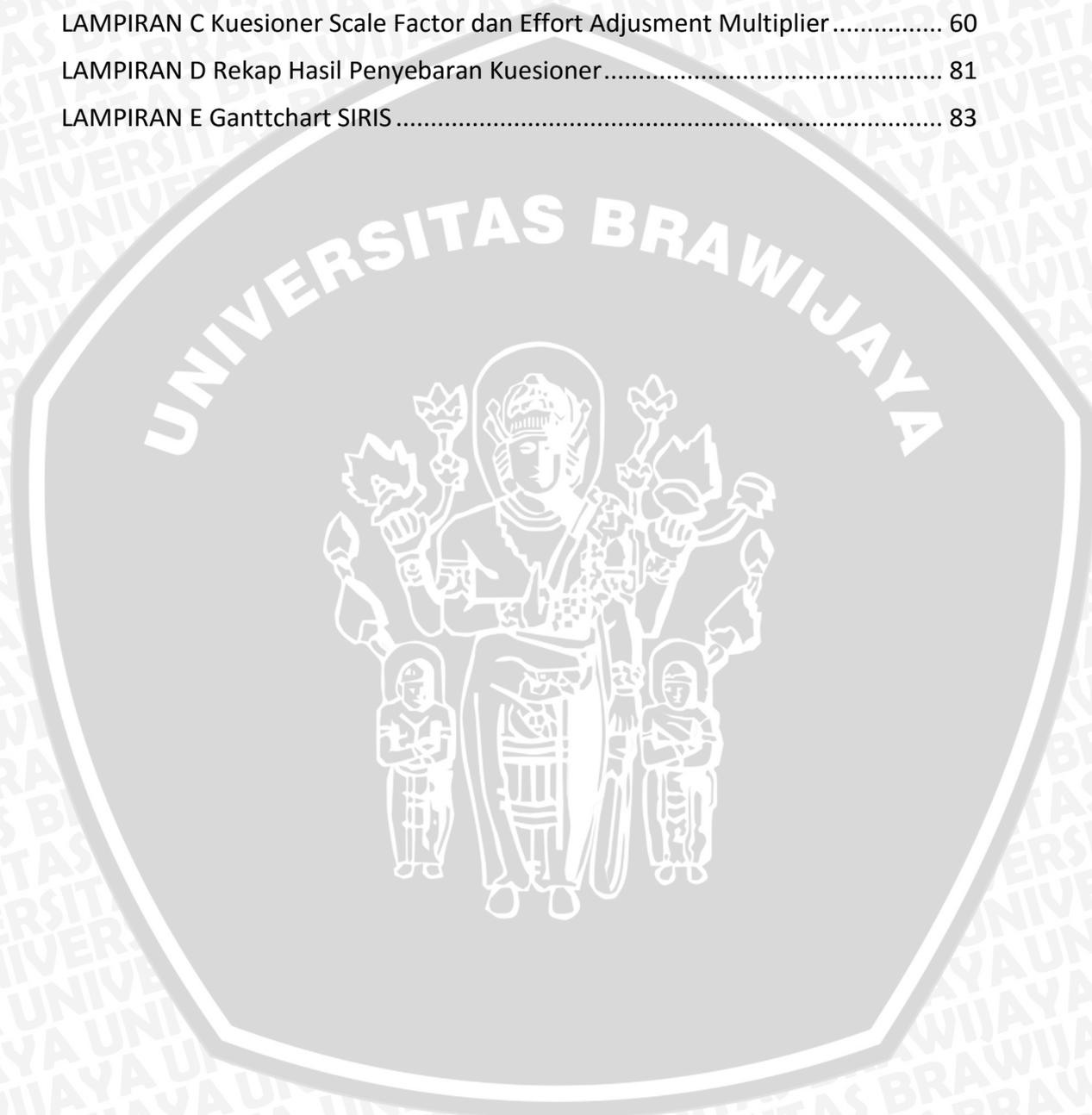
DAFTAR GAMBAR

Gambar 2.1 Data Storage <i>External Input</i>	22
Gambar 2.2 Data Storage <i>External Output</i>	22
Gambar 2.3 Data Storage <i>Internal Logical Files</i>	22
Gambar 2.4 Data Storage <i>External Interface Files</i>	22
Gambar 3.1 Alur Penelitian	22
Gambar 4.1 <i>Context Diagram SIRIS</i>	29
Gambar 4.2 DFD Level 1 SIRIS	30
Gambar 4.3 Komponen Tipe Fungsi Pengguna	30
Gambar 4.4 <i>Entity Relationship Diagram SIRIS</i>	31



DAFTAR LAMPIRAN

LAMPIRAN A Penjelasan Tentang Manajemen Proyek.....	54
LAMPIRAN B Pengerjaan Sistem SIRIS	57
LAMPIRAN C Kuesioner Scale Factor dan Effort Adjusment Multiplier.....	60
LAMPIRAN D Rekap Hasil Penyebaran Kuesioner.....	81
LAMPIRAN E Ganttchart SIRIS.....	83



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Dalam mengembangkan suatu proyek sistem informasi yang berkualitas perlu adanya manajemen proyek yang baik untuk mencapai tujuan suatu proyek. Menurut buku manajemen proyek yang ditulis oleh Budi Santosa, manajemen proyek adalah kegiatan dalam merencanakan, mengorganisasikan, mengarahkan dan mengendalikan sumber daya organisasi perusahaan untuk mencapai tujuan tertentu dalam waktu tertentu dengan sumber daya tertentu. Pengelolaan proyek tidak hanya ditinjau dari segi biaya yang dikeluarkan akan tetapi juga mengelola sumber daya yang dibutuhkan dalam menyelesaikan suatu proyek tersebut agar mendapatkan hasil yang efisien dan efektif. Salah satu cara agar perkembangan perangkat lunak dapat berjalan dengan baik dan lancar, yaitu menggunakan metode manajemen proyek. Manajemen proyek yang baik turut menentukan keberhasilan perusahaan dalam merencanakan, melaksanakan, dan mengakhiri proyek (Santosa, 2003). Agar proyek bisa dikatakan sukses, manajemen proyek harus mempertimbangkan 3 aspek yaitu ruang lingkup (*scope*), waktu (*time*), dan biaya (*cost*) (Schwalbe, 2014).

Dalam setiap pengerjaan sebuah proyek alangkah baiknya menggunakan pengerjaan yang terstruktur dengan mengacu pada dokumen *Work Breakdown Structure* (WBS). Agar proses inialisasi sebuah proyek dapat berjalan hingga akhir dengan baik harus adanya pembagian disetiap fase pengerjaannya yang terkelola dengan baik, didalam dokumen WBS terdapat pengembangan penjadwalan yaitu menggunakan *gantchart* yang berguna untuk mendetailkan disetiap pengerjaan dalam suatu proyek. Dokumen WBS dapat menjadi dasar perencanaan sebuah proyek agar berjalan efektif karena pada dokumen WBS dapat diidentifikasi, diperkirakan, dijadwalkan, dan dianggarkan sesuai kebutuhan proyek. Dalam hal ini dokumen WBS dapat menghubungkan antara ruang lingkup (*scope*), penjadwalan (*schedule*) dan biaya (*cost*) sebuah proyek (Rev, 2003).

Penelitian yang berjudul analisis alokasi sumber daya manusia dalam manajemen proyek sistem informasi website travelcar.co.id pada PT. Jagad Indonesia menyebutkan bahwa masalah yang sering dihadapi dalam pengerjaan proyek adalah sulitnya mengalokasikan sumber daya manusia, durasi dan biaya suatu proyek sistem informasi (Lestari, 2015). Kesulitan untuk menentukan jumlah estimasi sumber daya manusia yang tepat, juga didukung hasil wawancara (Lampiran A). Pihak *software house* mengatakan bahwa adanya kesulitan dalam merencanakan estimasi SDM, biaya proyek dan durasi sebuah proyek. Pengalokasian biaya dan SDM yang sedikit dapat menimbulkan waktu yang lebih lama dalam pengerjaan suatu proyek, dalam pengerjaan sebuah proyek harus adanya manajemen proyek untuk mendetailkan segala estimasi yang dibutuhkan oleh sebuah proyek. Menanggapi hal tersebut dalam penelitian ini menggunakan metode *constructive cost model II* (COCOMO II) untuk menghitung estimasi SDM, biaya dan durasi suatu proyek agar tidak melenceng dari perencanaan proyek

diawal, metode ini dikembangkan pada tahun 1981 oleh Barry Boehm yang bertujuan untuk memperkenalkan hirarki model estimasi atau perkiraan *Person-Months* dengan nama COCOMO 81 untuk mengembangkan sebuah perangkat lunak. Dengan berjalannya waktu pada tahun 2000 Barry Boehm memperkenalkan bukunya yang berjudul *software cost estimation with COCOMO II*, dalam buku tersebut Barry Boehm mengembangkan model COCOMO 81 menjadi model yang lebih komprehensif yang dinamain COCOMO II. Dalam pengembangan sebuah proyek dapat menggunakan metode COCOMO II untuk membantu dalam memperhitungkan estimasi biaya dan waktu yang dibutuhkan sebuah proyek agar berjalan dengan baik (Boehm, 2000).

Penelitian yang berjudul "*Improving Estimation Accuracy of the COCOMO II Using an Adaptive Fuzzy Logic Model*" (Siew et al, 2011) menunjukkan jumlah kegagalan proyek-proyek diselesaikan atas biaya yang banyak dan melebihi jadwal telah menjadi masalah yang signifikan bagi manajer proyek perangkat lunak. Perkiraan ini tidak hanya menyebabkan proyek melebihi anggaran, meskipun antara 30% dan 40% dari proyek perangkat lunak yang akhirnya terselesaikan, akan tetapi tidak sedikit proyek yang melebihi dari anggaran atau jadwal atau dibatalkan bahkan gagal. Estimasi biaya perangkat lunak adalah seperangkat teknik dan prosedur yang digunakan organisasi untuk sampai pada perkiraan untuk suatu penawaran, perencanaan proyek, dan perkiraan. Dengan demikian, akurasi estimasi adalah masalah yang sangat signifikan bagi para manajer, staf teknis, dan terutama praktisi yang melakukan atau mengandalkan estimasi biaya (Siew et al., 2011).

Masalah yang sering terjadi pada *software house* adalah dalam pengembangan suatu proyek kebanyakan sebuah organisasi tidak mementingkan manajemen proyek dengan benar, dengan adanya manajemen proyek yang benar dapat membantu organisasi tersebut dalam hal estimasi SDM, biaya proyek hingga durasi proyek. Menurut hasil wawancara (Lampiran A) pada pihak *software house* 7treesdigital, dalam pembuatan perencanaan suatu proyek diharuskan memiliki dokumen perancangan agar dapat berjalan dengan baik. Pada *software house* 7treesdigital menggunakan perhitungan biaya proyek dengan cara perkiraan atau bisa disebut dengan metode *guesstimate*. Metode *guesstimate* yang digunakan pada *software house* hanya memperkirakan biaya proyek dengan cara perkiraan (Marom, 2012).

Penelitian ini menggunakan proyek sistem informasi yang telah diselesaikan oleh *software house* terhitung selesai pada tahun 2014 yaitu sistem informasi rekam medis (SIRIS). Penelitian ini menjabarkan setiap pengerjaan proyek dengan menggunakan penjadwalan yang ada pada dokumen WBS menggunakan *gantchart* yang nantinya akan membantu mendetailkan setiap pengerjaan proyek agar dapat dikontrol. Setelah dijabarkan dari setiap permasalahan pengerjaan proyek, penelitian ini menggunakan metode COCOMO II untuk menghitung estimasi waktu, biaya proyek dan jumlah sumber daya manusia yang dibutuhkan dalam pengerjaan proyek tersebut. Metode matematika yang digunakan pada

COCOMO II akan menghasilkan estimasi biaya variabel sebagai fungsi *cost factor* yang utama. *Cost factor* adalah komponen-komponen biaya yang diperhitungkan dalam cost estimation model pada COCOMO II (Martini, 2012).

Penelitian ini, menentukan estimasi biaya proyek dengan menggunakan metode COCOMO II yang dianalisis dari estimasi biaya (*direct cost*). *Direct cost* adalah suatu biaya langsung yang berhubungan dengan produk atau jasa, misalnya seperti biaya gaji, pembelian hardware atau software (Schwalbe, 2014). Implementasi metode COCOMO II ada perhitungan estimasi usaha (*Person-Month*) yang secara langsung tergolong dalam estimasi biaya *direct cost*. Oleh karena itu penelitian ini mengangkat tema penelitian dengan judul “Perhitungan Biaya Proyek Sistem Informasi Rekam Medis Dengan Menggunakan Metode Cocomo II (Studi Kasus: 7Treesdigital)” yang nantinya akan membantu meminimalisir masalah tersebut.

1.2 Rumusan Masalah

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

1. Bagaimana implementasi penggunaan metode COCOMO II untuk melakukan analisis perhitungan estimasi biaya *direct cost* suatu proyek?
2. Bagaimana relasi antara penjadwalan pada dokumen WBS menggunakan *gantchart* dan penggunaan metode COCOMO II dalam perhitungan biaya suatu proyek?
3. Bagaimana hasil perbandingan perhitungan biaya proyek antara metode *guesstimate* dan metode COCOMO II?

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah:

1. Mengimplementasikan penggunaan COCOMO II untuk melakukan analisis perhitungan biaya *direct cost* dalam suatu proyek.
2. Menjelaskan peranan penggunaan dokumen WBS dan metode COCOMO II dalam pengerjaan suatu proyek.
3. Mengetahui hasil analisis perbandingan metode COCOMO II dan *Guesstimate*.

1.4 Manfaat

Manfaat yang dapat diperoleh dari penelitian ini antara lain:

1. Memberikan masukan bagi organisasi sebagai pertimbangan dalam mengalokasikan sumber daya dalam pengerjaan, baik itu waktu serta biaya yang harus dikeluarkan dalam pembuatan suatu proyek.
2. Mengurangi resiko terhadap terjadinya pembengkakan waktu dan biaya suatu proyek, serta dapat mengontrol pengerjaan proyek dengan penjadwalan yang ada pada dokumen WBS (*Work Breakdown Structure*) menggunakan *gantchart*.

1.5 Batasan Masalah

Agar masalah yang diteliti tidak keluar dan menyimpang, serta lebih terarah dan dapat dipahami dengan mudah maka perlu adanya batasan masalah. Adapun batasan masalah ini antara lain:

1. Ruang lingkup hanya dibatasi pada analisis perhitungan estimasi sumber daya manusia, biaya pengerjaan (*direct cost*), dan waktu suatu proyek.
2. Analisis perhitungan dilakukan dengan menggunakan metode COCOMO II untuk submodel *post architecture*.
3. Pada penelitian ini menggunakan sebuah sistem yang nantinya akan dianalisis, yaitu sistem informasi rekam medis (SIRIS).

1.6 Sistematika Pembahasan

Dalam penyusunan skripsi ini, sistematika penulisan terdiri dari 6 (enam) bab. Adapun uraian masing-masing bab tersebut adalah sebagai berikut:

Bab I: Pendahuluan

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

Bab II: Kajian Pustaka dan Dasar Teori

Mendeskripsikan dasar teori yang akan digunakan dan mendukung penelitian ini.

Bab III: Metodologi Penelitian

Menguraikan tentang metode yang akan digunakan dalam penelitian ini, serta menguraikan tentang metode pengambilan data, dan analisis hasil.

Bab IV: Pengumpulan Data

Berisi tentang pengumpulan data berupa wawancara, survey dan penyebaran kuesioner yang bertujuan untuk mendapatkan informasi terkait sistem. Dalam pengumpulan data ini respondennya bertujuan kepada tim pengembang dari sistem tersebut.

Bab V: Analisis dan Perhitungan

Menjelaskan tentang analisis dan perhitungan menggunakan metode COCOMO II dan *Guesstimate*. Analisis tersebut meliputi estimasi penerapan sumber daya dan waktu serta biaya dalam pengerjaan suatu proyek.

Bab VI: Penutup

Menjelaskan tentang kesimpulan dan saran hasil penelitian yang menjawab rumusan masalah yang telah diuraikan pada bab pendahuluan dan berisi saran yang dapat digunakan untuk memperbaiki sistem yang dijadikan objek penelitian.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Untuk mendukung penelitian ini mengambil dari beberapa referensi dari penelitian yang telah ada sebelumnya. Penelitian yang telah diambil dan dilakukan pada perusahaan dalam membuat estimasi sumber daya manusia (SDM) yang dibutuhkan, waktu yang dibutuhkan dan penjabaran setiap pengerjaannya dalam perancangan proyek. Referensi yang pertama mengacu pada pembuatan dokumen WBS dengan judul “*A Work Breakdown Structure for Implementing and Costing an ERP Project*”. Dalam Jurnal tersebut membahas penggunaan dokumen WBS untuk mengatur proyek *enterprise resource planning*, karena dalam pengerjaan proyek ERP tersebut penuh berbagai masalah terkait perencanaan proyek dan biaya proyek. Dalam hal ini dokumen WBS sangat berperan penting dalam memberikan detail segala pekerjaan yang harus dikerjakan sehingga menghasilkan perencanaan yang baik dengan menggunakan dokumen WBS (Momoh et al., 2008).

Setelah mendetailkan pekerjaan yang akan ditangani dalam pengerjaan proyek, kemudian perlu adanya menghitung estimasi waktu, jumlah pekerja yang akan bekerja didalam proyek dan biaya suatu proyek. Penelitian ini juga menggunakan refensi yang kedua yang mengacu pada penghitungan estimasi usaha dan jadwal pengerjaan proyek, jurnal yang digunakan ini berjudul “Estimasi Biaya Pembuatan Perangkat Lunak Menggunakan Metode Cocomo II Pada Sistem Informasi Pelaporan Kegiatan Pembangunan”. Dalam jurnal tersebut peneliti membuat estimasi usaha dan estimasi jadwal proyek perangkat lunak sistem informasi pelaporan kegiatan pembangunan dimana inputan yang dibutuhkan dalam jurnal tersebut berupa nilai *source line of code* yang kemudian dihitung menggunakan rumus *scale driver*, *effort multiplier*, faktor skala, penggerak biaya dan faktor fungsi pengguna kemudian menghasilkan perbandingan estimasi menggunakan *online freeware* dan menggunakan software modifikasi. Dimana software modifikasi lebih menunjukkan hasil rata-rata yang lebih rendah dibandingkan dengan *online freeware* (Primaraka, 2008).

Dari penjelasan dari referensi-referensi tersebut, membantu peneliti dalam menghitung estimasi waktu pengerjaan dan jumlah SDM yang dibutuhkan disetiap proyeknya. Adapun perbedaan dengan penelitian sebelumnya adalah adanya penambahan menggunakan gantt chart pada dokumen WBS untuk menentukan penjadwalan pengerjaan dan mendatailkan suatu pengerjaan disuatu proyek.

2.2 Profil 7Treesdigital

7Treesdigital dibawah legalitas PT. Dikasi Satu Digital, Malang. Memiliki sebuah tim kreatif yang telah berpengalaman secara individu di bidangnya, 7Treesdigital telah mengerjakan berbagai proyek berbasis *website* dan *mobile* baik yang berjalan secara *offline* maupun yang membutuhkan *web server*. Saat ini 7Treesdigital telah bekerjasama dengan beberapa perusahaan dari berbagai

negara, seperti negara-negara di Timur Tengah, yang membutuhkan jasa pembuatan *website* maupun *mobile apps* secara profesional. Berbagai aplikasi *website* dan *mobile* telah dikembangkan dengan baik.

Tim yang berkemampuan tinggi, didukung oleh kekompakan dan kerjasama yang solid, mampu menghasilkan produk yang berkualitas serta mampu bersaing dengan pengembang lainnya.

2.2.1 Visi 7Treesdigital

Menjadi perusahaan IT nomor satu di Indonesia yang mampu mengembangkan produk dan jasa IT berkelas dunia dengan selalu mengedepankan kerjasama tim, profesionalisme, dan semangat kerja yang tinggi.

2.2.2 Misi 7Treesdigital

1. Mengembangkan produk IT yang berkualitas dunia.
2. Menghadirkan jasa produk IT yang berkualitas dan memuaskan pelanggan.
3. Mengembangkan kerjasama dan kemitraan yang saling menguntungkan secara berkesinambungan.
4. Menciptakan jejaring kerjasama untuk menumbuhkan industri digital di Indonesia.
5. Memberikan *benefit* dan *value* bagi mitra pelanggan serta *stakeholder*.

2.3 Manajemen Proyek

2.3.1 Pengertian Manajemen

Menurut Iman Soeharto (2001) dalam buku manajemen proyek dari konseptual sampai operasional, manajemen merupakan proses merencanakan, mengorganisasikan, memimpin, dan mengendalikan kegiatan anggota serta sumber daya yang lain untuk mencapai sasaran organisasi (perusahaan) yang telah ditentukan (Soeharto, 2001).

2.3.2 Pengertian Proyek

Menurut Budi Santosa (2003), secara garis besar proyek memiliki empat tahapan berikut. Pertama, Tahap Konsepsi. Tahap menyusun dan merumuskan gagasan, menganalisis pendahuluan, dan melakukan studi kelayakan. Kedua, Tahap Pendefinisian. Tahap kegiatan penyiapan rencana proyek secara detail dan penentuan spesifikasi proyek secara rinci. Ketiga, Tahap Akuisisi. Tahap kegiatan yang terdiri dari desain, pengadaan fasilitas pendukung maupun material untuk tahap selanjutnya, produksi, dan implementasi. Keempat, Tahap Operasi. Tahap akhir suatu proyek dan proyek diserahkan kepada *user*. Tahap itu terjadi tergantung pada jenis proyek (Santosa, 2003).

2.3.3 Pengertian Manajemen Proyek

Menurut Schwalbe (2014), Ada sembilan area pengetahuan manajemen memiliki fungsi yang saling terkait satu sama lainnya di dalam area pengetahuan manajemen tersebut. Manajer proyek harus mempunyai pengetahuan dan keahlian di dalam sembilan area ini. Empat inti area pengetahuan manajemen

proyek meliputi manajemen lingkup proyek, waktu, biaya, dan manajemen kualitas. Pengetahuan manajemen itu dapat membantu manajer proyek untuk menentukan sumber daya manusia, komunikasi, risiko, dan manajemen pengadaan proyek. Dari inti ke empat area manajemen proyek tersebut saling berhubungan dengan sembilan area manajemen yaitu manajemen ruang lingkup proyek, manajemen waktu, manajemen biaya, manajemen kualitas proyek, manajemen sumber daya manusia, manajemen komunikasi, manajemen risiko, manajemen pengadaan, dan manajemen integrasi proyek (Schwalbe, 2014).

2.4 Work Breakdown Structure (WBS)

Sebuah WBS adalah landasan dari perencanaan proyek yang efektif dalam melaksanakan, mengendalikan dan pelaporan. Semua pekerjaan terkandung dalam WBS untuk diidentifikasi, diperkirakan, dijadwalkan, dan dianggarkan. WBS adalah struktur dan kode yang mengintegrasikan dan menghubungkan semua pekerjaan proyek berupa lingkup, jadwal, dan biaya. Oleh karena itu, WBS berisi proyek lingkup dasar yang diperlukan untuk mencapai tujuan teknis sebuah proyek. WBS digunakan sebagai alat manajemen di seluruh siklus hidup dari proyek untuk mengidentifikasi, menetapkan, dan melacak total lingkup kerjanya. Ketika pendanaan proyek awal yang diterima, Direktur Proyek (PD) mengembangkan WBS yang mengidentifikasi dana yang diperlukan sesuai dengan jadwal dan kebutuhan tugas dalam elemen WBS. WBS umumnya adalah kerangka multi-level yang mengatur dan menampilkan grafis elemen yang mewakili pekerjaan yang harus dicapai (Rev, 2003). Untuk struktur pekerjaan proyek menjadi elemen-elemen WBS yang harus diketahui:

1. Didefinisikan, dapat dijelaskan dan mudah dipahami oleh peserta proyek.
2. Dikelola, unit yang berarti suatu pekerjaan dimana tanggung jawab dan wewenang tertentu dapat ditugaskan kepada seorang pekerja.
3. Perkiraan, durasi dapat diperkirakan dalam waktu yang dibutuhkan untuk menyelesaikan dan biaya dapat diperkirakan dari sumber daya manusia yang dibutuhkan dalam menyelesaikan sebuah proyek.
4. Bebas, tidak ketergantungan pada elemen lainnya yang sedang berlangsung.
5. Terintegrasi, terintegrasi dengan elemen-elemen pekerjaan proyek dan dengan perkiraan biaya dan jadwal untuk menyertakan seluruh proyek.
6. Terukur, dapat digunakan untuk mengukur sebuah kemajuan proyek.
7. Adaptasi, fleksibel sehingga apabila ada penambahan atau penghapusan lingkup pekerjaan dapat segera ditampung dalam rangka WBS.

2.5 Metode COCOMO II

2.5.1 Metode COCOMO II

Menurut Barry Boehm (1981) mendesain COCOMO untuk memberikan estimasi/perkiraan jumlah *person-months* untuk mengembangkan suatu produk perangkat lunak. Referensi pada model ini dikenal dengan nama COCOMO 81. Model estimasi COCOMO telah digunakan oleh ribuan manajer proyek suatu

proyek perangkat lunak, dan berdasar pada pengalaman dari ratusan proyek sebelumnya. Dalam buku klasik tentang “*Software Engineering Economics*”, Barry Boehm memperkenalkan hierarki model estimasi perangkat lunak yang memiliki nama COCOMO, singkatan dari **Constructive Cost Model** atau Model Biaya Konstruktif (Boehm, 1981). Model COCOMO yang asli menjadi salah satu model estimasi biaya perangkat lunak yang paling banyak digunakan dan dibahas dalam industri. Model tersebut telah berkembang menjadi model estimasi yang lebih komprehensif, dinamai COCOMO II. Model ini menghitung usaha pengembangan perangkat lunak sebagai fungsi ukuran program dan serangkaian pengendali biaya yang menyangkut penilaian subyektif terhadap produk, perangkat keras, personil dan atribut proyek. Dalam bukunya Barry Boehm juga menjelaskan tentang model-model COCOMO II. Terdapat 4 model-model dari COCOMO II menurut Barry Boehm, yaitu:

1. Dasar Cocomo

Dengan menggunakan estimasi parameter persamaan (dibedakan menurut tipe sistem yang berbeda) upaya pengembangan dan pembangunan durasi dihitung berdasarkan perkiraan. Dengan rincian untuk fase ini diwujudkan dalam persentase. Model COCOMO dapat diaplikasikan dalam tiga tingkatan kelas (Boehm, 2000):

- a. Proyek organik (*organic mode*) Adalah proyek dengan ukuran relatif kecil, dengan anggota tim yang sudah berpengalaman, dan mampu bekerja pada permintaan yang relatif fleksibel.
- b. Proyek sedang (*semi-detached mode*) Merupakan proyek yang memiliki ukuran dan tingkat kerumitan yang sedang, dan tiap anggota tim memiliki tingkat keahlian yang berbeda
- c. Proyek terintegrasi (*embedded mode*) Proyek yang dibangun dengan spesifikasi dan operasi yang ketat.

Model COCOMO dasar dapat dihitung menggunakan persamaan-persamaan berikut ini:

$$E = A_b (KLOC)^{b_b} \quad (\text{Persamaan 2.1})$$

$$D = C_b (E)^{d_b} \quad (\text{Persamaan 2.2})$$

$$P = E \times D \quad (\text{Persamaan 2.3})$$

Keterangan:

E: Besarnya Usaha (orang perbulan)

D: Lama Waktu Pengerjaan (bulan)

KLOC: Estimasi Jumlah Baris Kode (ribuan)

P: Jumlah Orang Yang Diperlukan

A_b, b_b, C_b, d_b : Koefisien

2. Menengah Cocomo

Persamaan estimasi sekarang mempertimbangkan 15 pengaruh faktor-faktor ini adalah atribut produk (seperti kehandalan perangkat lunak, ukuran *database*, kompleksitas), komputer atribut-atribut (seperti pembatasan waktu komputasi, pembatasan memori utama), personil atribut (seperti aplikasi pemrograman dan pengalaman, pengetahuan tentang bahasa pemrograman), dan proyek atribut (seperti lingkungan pengembangan perangkat lunak, tekanan waktu pengembangan). Tingkat pengaruh yang dapat diklasifikasikan sebagai sangat rendah, rendah, normal, tinggi, sangat tinggi, ekstra tinggi.

3. Detail Cocomo

Dalam hal ini adalah rincian untuk tahapan yang tidak diwujudkan dalam persentase, tetapi dengan cara faktor-faktor pengaruh dialokasikan. Pada saat yang sama, maka dibedakan menurut tiga tingkatan hirarki produk (modul, subsistem, sistem), produk yang berhubungan dengan faktor-faktor pengaruh sekarang dipertimbangkan dalam persamaan estimasi yang sesuai. Selain itu detail COCOMO dapat menghubungkan semua karakteristik versi *intermediate* dengan penilaian terhadap pengaruh pengendali biaya pada setiap langkah (analisis, perancangan, dll) dari proses rekayasa perangkat lunak.

4. Aplikasi Teknis

Untuk proyek perangkat lunak mengembangkan aplikasi komersial, biasanya COCOMO muncul dengan nilai-nilai estimasi usaha, karena itu COCOMO diterapkan untuk pengembangan perangkat lunak secara teknis. Keadaan ini disebabkan oleh hasil implementasi dalam persamaan estimasi COCOMO sesuai dengan tingkat produktivitas dalam pembangunan secara teknis yang berkaitan dengan pengembangan perangkat lunak komersial yang lebih tinggi dalam tingkat produktivitas. Dengan menggunakan dasar estimasi instruksi (DSI) itu berusaha untuk mengurangi ke tidak pastian dan masalah besar sehubungan dengan basis estimasi tradisional *line of code* (LOC). Namun, beberapa masalah tetap tidak ada kepastian dari estimasi DSI dan untuk usaha pengembangan yang DSI yang berdasarkan metode rekayasa perangkat lunak modern tidak lagi sangat penting karena semakin terjadi kegiatan usaha diawal dan DSI hanya akan efektif menjelang akhir proses pembangunan, DSI serta LOC tergantung pada bahasa pemrograman yang dipilih.

2.5.2 Submodel COCOMO II

Dalam jurnal yang berjudul “Estimasi Biaya Pembuatan Perangkat Lunak Menggunakan metode COCOMO II Pada Sistem Informasi Pelaporan Kegiatan Pembangunan” menjelaskan juga tentang sub-model dari Metode COCOMO II yaitu sebagai berikut (Primaraka, 2008):

1. *Application Composition*

Model ini cocok untuk proyek yang dibangun dengan menggunakan sistem yang menerapkan GUI.

2. *Early Design*

Model ini digunakan untuk membuat perkiraan kasar dari biaya proyek dan durasi sebelum seluruh arsitektur ditentukan. Model ini menggunakan

sebuah *cost drivers* dan persamaan estimasi.

3. *Post Architecture*

Model ini adalah Model yang paling rinci dari COCOMO II. Model ini digunakan setelah arsitektur keseluruhan proyek dikembangkan. Model ini memiliki perhitungan biaya baru, aturan penghitungan baris baru, dan persamaan yang baru.

Pada penelitian ini, tahap estimasi perancangan sistem menggunakan sub model *post architecture* karena sudah adanya sistem yang telah dikembangkan jadi peneliti akan menghitung estimasi waktu sebuah proyek dan menentukan jumlah pekerja yang dibutuhkan dalam pengerjaan sebuah proyek tersebut. Setelah didapatkan perhitungan tersebut menggunakan metode COCOMO II, nantinya akan didukung oleh pembuatan penjadwalan yang ada pada dokumen WBS dengan menggunakan *gantchart* dalam perincian setiap pengerjaannya.

2.5.3 *Unadjusted Function Points (UFP)*

UFP adalah untuk mengukur proyek perangkat lunak dengan mengkuantisasi kegunaan pemrosesan informasi yang berhubungan dengan tipe berkas, keluaran, masukan data atau kontrol eksternal. Salah satu penentuan tipe fungsi pengguna dapat menggunakan pendekatan berdasar pada data *flow diagram* dan *database* program (ERD), dimana ERD ini merepresentasikan penggunaan *database* beserta tabel dan *field* atau atribut. Data ini digunakan untuk menentukan bobot kompleksitas berdasar aturan *Data Elemen Type*, *File Type References*, dan *Record Element Type* (Boehm, 2000).

Menghitung UFP harus mengidentifikasi lima tipe fungsi pengguna terlebih dahulu, pada tabel 2.1 akan menjelaskan kelima tipe fungsi pengguna tersebut dalam estimasi berdasarkan *unadjusted function point*.

Tabel 2.1 Tipe Fungsi Pengguna

1	<i>External Input</i> (EI)	Jumlah setiap tipe data atau masukan kontrol pengguna unik yang memasuki batas luar dari sistem perangkat lunak yang sedang diukur dan menambah atau mengubah data dalam berkas <i>internal logik</i> .
2	<i>External Output</i> (EO)	Jumlah setiap tipe data atau keluaran kontrol unik yang meniggalkan batas luar dari sistem perangkat lunak yang sedang diukur.
3	<i>Internal Logical File</i> (ILF)	Jumlah setiap grup <i>logik</i> utama dari data atau informasi kontrol pengguna dalam sistem perangkat lunak sebagai tipe berkas <i>internal logik</i> . Meliputi setiap berkas <i>logik</i> yang dibuat, digunakan atau dirawat oleh sistem perangkat lunak.
4	<i>External Interface Files</i> (EIF)	Berkas yang dilewatkan atau dibagikan diantara sistem perangkat lunak seharusnya dihitung sebagai tipe berkas antarmuka eksternal dalam setiap sistem.

Sumber: Pranatha et al (2012)

Tabel 2.1 Tipe Fungsi Pengguna (Lanjutan)

5	<i>External Inquiry (EQ)</i>	Jumlah setiap kombinasi masukan keluaran, dimana masukan menyebabkan dan menimbulkan keluaran yang seketika, sebagai tipe <i>inquiry</i> eksternal.
---	------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------

Sumber: Pranatha et al (2012)

2.5.3.1 External Input (EI)

Jumlah setiap tipe data atau masukan kontrol pengguna unik yang memasuki batas luar dari sistem perangkat lunak yang sedang diukur dan menambah atau mengubah data dalam berkas *internal logik*. Contoh: sebuah *database* yang mendapat masukan dari pengguna. Berikut adalah gambar 2.1 lambang DFD yang digunakan pada *external input* (EI) (Litvinenko, 2010):



Gambar 2.1 Data Storage External Input

2.5.3.2 External Output (EO)

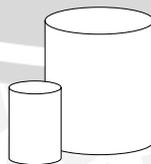
Jumlah setiap tipe data atau keluaran kontrol unik yang meninggalkan batas luar dari sistem perangkat lunak yang sedang diukur. Contoh: sebuah *database* yang menghasilkan keluaran untuk pengguna. Berikut adalah gambar 2.2 lambang DFD yang digunakan pada *external output* (EO) (Litvinenko, 2010):



Gambar 2.2 Data Storage External Output

2.5.3.3 Internal Logical File (ILF)

Jumlah setiap grup *logik* utama dari data atau informasi kontrol pengguna dalam sistem perangkat lunak sebagai tipe berkas *internal logik*. Meliputi setiap berkas *logik* yang dibuat, digunakan atau dirawat oleh sistem perangkat lunak. Contoh: sebuah *database* yang dapat memelihara data secara internal. Berikut adalah gambar 2.3 lambang DFD yang digunakan pada *internal logical file* (ILF) (Litvinenko, 2010):

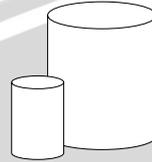


Gambar 2.3 Data Storage Internal Logical File



2.5.3.4 External Interface Files (EIF)

Berkas yang dilewatkan atau dibagikan diantara sistem perangkat lunak seharusnya dihitung sebagai tipe berkas antarmuka eksternal dalam setiap sistem. Contoh: sebuah *database* yang memelihara data secara external. Berikut adalah gambar 2.4 lambang DFD yang digunakan pada *external interface files* (EIF) (Litvinenko, 2010):



Gambar 2.4 Data Storage External Interface Files

Setelah menganalisis nilai tipe fungsi pengguna seperti tabel 2.1 diatas, setiap kategori memiliki nilai perhitungan yang nantinya akan dikalikan dengan bobot kompleksitas fungsi pengguna yang ditunjukkan pada tabel 2.2 untuk menghasilkan nilai *unadjusted function points* (UFP). berikut tabel bobot kompleksitas fungsi pengguna berdasarkan kriteria dari setiap kategorinya (Pranatha et al., 2012).

Tabel 2.2 Bobot Kompleksitas Fungsi Pengguna

Tipe Fungsi	Bobot Kompleksitas		
	Low	Average	High
Internal Logical Files	7	10	15
External Interface Files	5	7	10
External Input	3	4	6
External Outputs	4	5	7
External Inquiry	3	4	6

Sumber: Pranatha et al (2012)

Dalam menentukan bobot kompleksitas berdasarkan pada jumlah tipe elemen data tiap fungsi dan jumlah referensi tipe *file*. Ada 3 kategori dalam menentukan bobot kompleksitas, yaitu DET, RET dan FTR. Berikut akan dijelaskan bobot kompleksitas DET, RET dan FTR pada tabel 2.3 sebagai berikut:

Tabel 2.3 DET, RET dan FTR

Kategori	Keterangan
Data Element Type (DET)	<i>Field</i> yang tak berulang dan diidentifikasi user sebagai <i>field</i> yang unik.
Record Element Type (RET)	Subgroup dari data <i>element type</i> yang berada dalam <i>Internal Logical File</i> (ILF) atau <i>Eksternal Internface Files</i> (EIF).

Sumber: Pranatha et al (2012)

Tabel 2.3 DET, RET dan FTR (Lanjutan)

File Type Reference (FTR)	Sebuah jenis file yang dibaca oleh fungsi transaksional. Fungsi transaksional merepresentasikan fungsionalitas yang disediakan <i>user</i> untuk melakukan pemrosesan data menggunakan aplikasi.
----------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Sumber: Pranatha et al (2012)

Setelah menganalisis dari tiap kategori DET, RET dan FTR, Kemudian akan dihitung nilai dari DET, RET dan FTR tersebut kedalam tingkat bobot kompleksitas UFP seperti tabel 2.4 berikut:

Tabel 2.4 Bobot Kompleksitas *Unadjusted Function Point* (UFP)

Untuk Internal Logical Files dan External Interface Files			
<i>Data Element Tipe</i>			
<i>Record Element Type</i>	1-19	20-50	51+
1	Low	Low	Avg.
2-5	Low	Avg.	High
6+	Avg.	High	High
Untuk External Output dan External inquiry			
<i>Data Element Tipe</i>			
<i>File Type Reference</i>	1-5	6-19	20+
0 atau 1	Low	Low	Avg.
2-3	Low	Avg.	High
4+	Avg.	High	High
Untuk External Input			
<i>Data Element Tipe</i>			
<i>File Type Reference</i>	1-4	5-15	16+
0 atau 1	Low	Low	Avg.
2-3	Low	Avg.	High
3+	Avg.	High	High

Sumber: Pranatha et al (2012)

Tingkat kerumitan dalam menentukan bobot yang akan digunakan pada bobot kompleksitas fungsi pengguna untuk menentukan *unadjusted function points* (UFP). UFP yang telah diperoleh kemudian didengan nilai QSM yang digunakan, setelah itu dikonversikan kedalam satuan KLOC dengan cara dibagikan dengan 1000 agar dapat disubstitusikan ke dalam persamaan estimasi usaha (Sarno et al., 2002). Pada tabel 2.5 dibawah ini adalah untuk bahasa pemrograman yang ada berdasarkan pada *Quantitative Software Management* (QSM) *Function Point Language*. Pada jurnal penelitian (Pranatha et al., 2012) yang berjudul “Analisis Perkiraan Biaya Pembuatan *Enterprise Resource Planning* Modul Pabrik Gula Diperusahaan Perkebunan Dengan Metode COCOMO II” menjelaskan hasil



perhitungan UFP dikalikan dengan bahasa pengembangan yang digunakan dalam sistem. Dalam penggunaan QSM ini menggunakan nilai median karena nilai median menjadi indikator yang lebih akurat dari tendensi sentral (Ridge, 2016).

Tabel 2.5 QSM SLOC

BAHASA	QSM SLOC			
	Avg	Median	Low	High
ABAP (SAP) *	28	18	16	60
ASP*	51	54	15	69
Assembler *	119	98	25	320
C *	97	99	39	333
C# *	50	53	25	80
HTML *	34	40	14	48
Java *	53	53	14	134
JavaScript *	47	53	31	63
VB.NET *	52	60	26	60
Visual Basic *	42	44	20	60

Sumber: QSM (2016)

2.5.4 Scale Factor

Untuk mengetahui *scale factor*, ada beberapa parameter pengukuran bobot. Namun, dalam pemberian nilai disetiap bobotnya terhadap masing-masing parameter melalui cara penyebaran kuesioner, yang nantinya akan diberikan kepada responden (tim pengembang). Setiap skala rating mempunyai rentang dari *very low* hingga *extra high*. Setiap tingkat rating memiliki bobot yang disebut *W*, nilai spesifik disebut *scale factor*. *Scale factor* proyek akan dijumlah semua, kemudian digunakan untuk menentukan eksponen skala B. Perhitungan *scale factor* menggunakan persamaan 2.4 berikut (Boehm, 2000):

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j \quad (\text{Persamaan 2.4})$$

Keterangan:

B: 0.91 (for COCOMO II)

SF: *scale factor*

$\sum_{j=1}^5 SF_j$: bobot *scale factor*

Menurut Boehm (2000), terdapat 5 *scale factor* sebagai parameter yaitu *Precedentedness* (PREC), *Development Flexibility* (FLEX), *Risk Resolution* (RESL), *Team Cohesion* (TEAM) dan *Project Maturity* (PMAT). Berikut tabel 2.6 yang menjelaskan *scale factor*, dan tabel 2.7 menjelaskan nilai bobot dari *scale factor*:

Tabel 2.6 Definisi Scale Factor

FAKTOR SKALA	DESKRIPSI
<i>Precedentedness</i>	Faktor skala ini merupakan faktor skala yang menggambarkan pengalaman masa lalu organisasi dengan proyek-proyek sejenis.
<i>Development</i> <i>Fleksibility</i>	Faktor skala ini merupakan faktor skala yang menggambarkan kemampuan klien dalam menentukan tujuan dan mengkomunikasikan kebutuhan perangkat lunak kepada tim pengembang.
<i>Risk Resolution</i>	Faktor skala ini merupakan faktor skala yang menggambarkan proses identifikasi risiko yang terkait dengan proyek, untuk menghasilkan pengukuran jumlah risiko yang terkait dengan proyek.
<i>Team Cohesion</i>	Faktor skala ini merupakan faktor skala yang menggambarkan seberapa baik tim pengembangan bekerja sama.
<i>Process</i> <i>Maturity</i>	<p>Faktor skala ini merupakan faktor skala yang menggambarkan kematangan proses pengembangan perangkat lunak dalam organisasi. Hal ini didasarkan pada Model Kematangan Kemampuan Rekayasa Perangkat Lunak <i>Institute</i> (CMM) yang didefinisikan berdasarkan lima tingkat kematangan untuk proses pengembangan perangkat lunak. Berikut adalah deskripsi dari tiap tingkatan kematangan kemampuan rekayasa perangkat lunak:</p> <ul style="list-style-type: none"> i. <i>Initial Level</i> Pada tingkat ini, organisasi tidak memiliki prosedur manajemen yang efektif atau rencana proyek. Prosedur formal untuk pengendalian proyek ada dan tidak ada mekanisme organisasi untuk memastikan bahwa prosedur tersebut digunakan secara konsisten. Di level ini, organisasi berhasil mengembangkan perangkat lunak namun karakteristik perangkat lunak (kualitas dll) dan proses (anggaran, jadwal dll) akan tak terduga. ii. <i>Repeatable Level</i> Pada tingkat ini, proses-proses dasar manajemen proyek sudah dikembangkan untuk melacak biaya, jadwal, dan fungsionalitas perangkat lunak. Disebut tingkat berulang karena organisasi sudah mendokumentasikan proses-proses yang perlu agar dapat berhasil mengulangi proyek dari jenis yang sama. iii. <i>Defined Level</i> Pada tingkat ini, organisasi memiliki proses dan kegiatan rekayasa perangkat lunak (definisi kebutuhan, <i>development</i>, <i>maintenance</i>) sudah didokumentasikan dan diintegrasikan dalam proses pengembangan perangkat lunak yang standar untuk digunakan dalam organisasi. Keseluruhan proyek dikembangkan dan dipelihara menggunakan proses yang standar dan resmi diakui dalam organisasi. iv. <i>Managed Level</i> Sebuah organisasi pada level 4 memiliki ukuran yang rinci tentang proses pengembangan perangkat lunak dan kualitas produk yang telah dikumpulkan dan didokumentasikan, baik proses/kualitas produk secara kuantitatif yang dapat dimengerti dan dikontrol. Proses dan pengukuran produk dikumpulkan dan dimasukkan ke dalam aktivitas perbaikan proses. v. <i>Optimising Level</i> Pada tingkat ini, sebuah organisasi berkomitmen untuk perbaikan proses yang berkesinambungan. Proses perbaikan yang dianggarkan dan direncanakan dan merupakan bagian integral dari proses organisasi.

Sumber: Boehm (2000)



Tabel 2.7 Bobot Scale Factor COCOMO II submodel *post architecture*

Faktor skala	Very Low	Low	Nominal	High	Very High	Extra High
<i>Precedentedness</i>	6.20	4.96	3.72	2.48	1.24	0.00
<i>Flexibility</i>	5.07	4.05	3.04	2.03	1.01	0.00
<i>Risk Resolution</i>	7.07	5.65	4.24	2.83	1.41	0.00
<i>Team Cohesion</i>	5.48	4.38	3.29	2.19	1.10	0.00
<i>Process Maturity</i>	7.80	6.24	4.68	3.12	1.56	0.00

Sumber: Boehm (2000)

2.5.5 Effort Adjustment Multipliers (EM)

Menurut Boehm (2000), terdapat 17 *point effort adjustment multipliers* yang digunakan dalam submodel *post architecture* COCOMO II untuk menentukan usaha nominal yang menggambarkan suatu proyek perangkat lunak yang sedang dikembangkan. Dalam 17 poin tersebut akan dibagi menjadi 6 kelas mulai dari *very low* hingga *extra high*. Kemudian dibagi menjadi 4 kelompok seperti tabel 2.8 (Boehm, 2000):

Tabel 2.8 Pengelompokkan Faktor Biaya

Kelompok	Effort Adjustment Multipliers
<i>Product</i>	RELY, DATA, CPLX, RUSE, DOCU
<i>Platform</i>	TIME, STOR, PVOL
<i>Personnel</i>	ACAP, PCAP, PCON, AEXP, PEXP, LTEX
<i>Project</i>	TOOL, SITE, SCED

Sumber: Boehm (2000)

Setelah itu, pada tabel 2.9 akan menjelaskan terkait 17 poin dari *effort adjustment multipliers* dan pada tabel 2.10 menjelaskan detail nilai dari bobot yang ada pada *effort adjustment multipliers*:

Tabel 2.9 Effort Adjustment Multipliers

EFFORT ADJUSTMENT MULTIPLIERS	DESKRIPSI
RELY	RELY (Keandalan perangkat lunak yg disyaratkan): Penilaian <i>cost driver</i> terkait sejauh mana perangkat lunak menjalankan aplikasi sesuai fungsinya selama periode waktu.
DATA	DATA (Ukuran Database): Penilaian <i>cost driver</i> terkait ukuran database yang digunakan. Ukuran dapat diperoleh dengan menghitung D/P.

Sumber: Boehm (2000)



Tabel 2.9 Effort Adjustment Multipliers (Lanjutan)

CPLX	CPLX (Kompleksitas Produk): Penilaian <i>cost driver</i> terkait perangkat lunak dan perangkat keras dalam melakukan tugasnya, seperti <i>platform</i> (arsitektur, sistem operasi, bahasa pemrograman dan antarmuka yang terkait), sistem manajemen <i>database</i> , <i>browser</i> yang sesuai digunakan dalam menjalankan aplikasi ini.
DOCU	DOCU (Dokumentasi sesuai dengan kebutuhan siklus hidup perangkat lunak): Penilaian <i>cost driver</i> dalam hal kesesuaian dokumentasi proyek terhadap kebutuhan siklus hidup perangkat lunak.
ACAP	ACAP (Kemampuan Analisis Sistem): Penilaian <i>cost driver</i> terkait kemampuan personel dalam analisis dan desain, efisiensi dan ketelitian, serta kemampuan untuk berkomunikasi dan bekerja sama. Dalam hal ini, dapat dinilai dari sertifikasi yang sudah didapatkan personel atau pengalaman kerja tim dalam suatu proyek.
PCAP	PCAP (Kemampuan Programmer): Penilaian <i>cost driver</i> terkait kemampuan programmer dalam efisiensi penulisan kode program, ketelitian, dan kemampuan untuk berkomunikasi dan bekerja sama sebagai sebuah tim. Dengan kata lain sudah berapa banyak proyek di mana <i>programmer</i> tersebut ikut terlibat didalamnya.
PCON	PCON (Kontinuitas Personil): Penilaian <i>cost driver</i> terkait tingkat pergantian personel tiap tahun pada proyek.
RUSE	RUSE (Pengembangan kode yang diarahkan untuk penggunaan kembali): Penilaian <i>cost driver</i> terkait tingkat upaya yang diperlukan untuk mengembangkan komponen yang dimaksudkan untuk digunakan kembali pada proyek-proyek yang sedang berjalan atau proyek di masa mendatang.
TIME	TIME (Kendala waktu eksekusi): Penilaian <i>cost driver</i> terkait persentase kendala waktu eksekusi yang diharapkan dapat digunakan pada sistem perangkat lunak.
STOR	STOR (Kendala Penyimpanan Utama): Penilaian <i>cost driver</i> terkait persentase tingkat kendala penyimpanan utama yang dikenakan pada sistem perangkat lunak.
PVOL	PVOL (Kompleksitas dari hardware dan software): Penilaian <i>cost driver</i> terkait perubahan yang terjadi pada <i>hardware</i> dan <i>software</i> (OS, DBMS) pada kurun waktu tertentu.
AEXP	AEXP (Pengalaman tim dalam mengembangkan aplikasi): Penilaian <i>cost driver</i> terkait tingkat pengalaman kerja tim proyek pada suatu proyek pengembangan aplikasi sistem perangkat lunak atau subsistem.
PEXP	PEXP (Pengalaman tim dalam pengembangan terkait platform aplikasi): Penilaian <i>cost driver</i> terkait pemahaman tim dalam menggunakan platform, interface, database, jaringan & middleware.

Sumber: Boehm (2000)

Tabel 2.9 Effort Adjustment Multipliers (Lanjutan)

LTEX	LTEX: Penilaian <i>cost driver</i> terkait pengalaman dalam pemrograman dengan bahasa tertentu dan pemanfaatan <i>CASE tools</i> dari tim proyek dalam mengembangkan perangkat lunak.
TOOL	TOOL (Penggunaan <i>software</i> bantuan): Penilaian <i>cost driver</i> terkait penggunaan <i>CASE tool</i> dalam pengembangan perangkat lunak pada proyek, seperti dari mengubah kode yang sederhana menjadi terintegrasi.
SITE	SITE: Penilaian <i>cost driver</i> terkait cara distribusi yang digunakan dalam pengembangan perangkat lunak pada proyek, apakah masih via telepon, fax, email, video <i>conference</i> , atau bahkan sudah menggunakan <i>interactive</i> multimedia.
SCED	SCED: Penilaian <i>cost driver</i> terkait tingkat persentase dari percepatan atau kemunduran jadwal terhadap jadwal suatu proyek yang telah ditetapkan sebelumnya.

Sumber: Boehm (2000)

Tabel 2.10 Bobot Effort Multipliers COCOMO II Post Architecture

Faktor skala	Very Low	Low	Nominal	High	Very High	Extra High
RELY	0.82	0.92	1.00	1.10	1.26	-
DATA	-	0.90	1.00	1.14	1.28	-
CPLX	0.73	0.87	1.00	1.17	1.34	1.74
RUSE	-	0.95	1.00	1.07	1.15	1.24
DOCU	0.81	0.91	1.00	1.11	1.23	-
TIME	-	-	1.00	1.11	1.29	1.63
STOR	-	-	1.00	1.05	1.17	1.46
PVOL	-	0.87	1.00	1.15	1.30	-
ACAP	1.42	1.19	1.00	0.85	0.71	-
PCAP	1.34	1.15	1.00	0.88	0.76	-
PCON	1.29	1.12	1.00	0.90	0.81	-
AEXP	1.22	1.10	1.00	0.88	0.81	-
PEXP	1.19	1.09	1.00	0.91	0.85	-
LTEX	1.20	1.09	1.00	0.91	0.84	-
TOOL	1.17	1.09	1.00	0.90	0.78	-
SITE	1.22	1.09	1.00	0.93	0.86	0.80
SCED	1.43	1.14	1.00	1.00	1.00	-

Sumber: Boehm (2000)

Setelah dilakukan penyebaran kuesioner dengan responden dari tim pengembang, maka akan terkumpul informasi berupa data-data tentang *scale factor*, *effort adjustment multiplier* dan fungsi pengguna *unadjusted function points*.

2.5.6 Estimasi Usaha (*Person-Month*)

Dalam perhitungan *Person Month* (PM) submodel *post architecture* dalam buku Barry Boehm menggunakan persamaan 2.5 sebagai berikut (Boehm, 2000):

$$PM = A \times (\text{size})^E \times \prod_{i=1}^{17} EM_i \quad (\text{Persamaan 2.5})$$

Keterangan:

PM: *Person-Month*

A: 2.94 (*for COCOMO II.2000*)

E: nilai *scale factor*

Size: nilai KLOC

$\prod_{i=1}^{17} EM_i$: nilai *effort adjustment multiplier*

Persamaan diatas menggunakan input *Size* dari pengembangan perangkat lunak. Ukuran yang digunakan adalah dalam *source line of code* (SLOC). SLOC diperoleh dari perkiraan ukuran modul perangkat lunak yang menyusun program aplikasi. Ukuran tersebut dapat juga diperkirakan dari *unadjusted function points* (UFP), kemudian diubah kedalam KLOC dengan cara dibagi 1000.

2.5.7 Estimasi Biaya

COCOMO II memberikan sebuah kemampuan untuk estimasi perhitungan biaya yang digunakan untuk menghitung estimasi pengerjaan suatu proyek. Perhitungan estimasi biaya menggunakan persamaan 2.6 sebagai berikut (Boehm, 2000):

$$TDEV = [3.67 \times (PM)^{(0.28+0.2x(e-0.91))}] \quad (\text{Persamaan 2.6})$$

Keterangan:

TDEV = *Time To Develop* (waktu dalam bulan)

PM = *Person-Month*

E = nilai *scale factor*

Setelah didapatkan nilai dari perhitungan TDEV dan *Average Staffing* (P), kemudian menghitung biaya total dengan persamaan 2.7 berikut (Pranatha et al., 2012):

$$\text{Biaya Total} = \text{Average Staffing (P)} \times \text{Average Cost Labor} \quad (\text{Persamaan 2.7})$$

Keterangan:

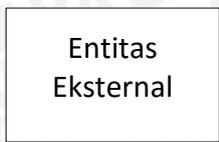
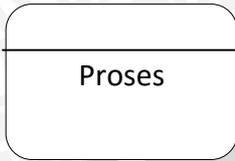
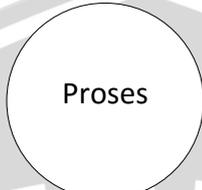
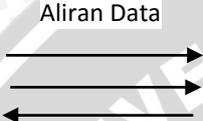
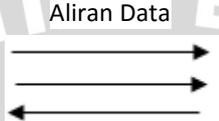
Average Staffing (P) = jumlah pekerja

Average Cost Labor = rata-rata biaya pekerja

2.5.8 Data Flow Diagram (DFD)

Suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data pada suatu sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. Berikut notasi-notasi yang ada pada DFD:

Tabel 2.11 Notasi Data Flow Diagram

Gane/Sarson	Yourdon/DeMarco	Keterangan
		Entitas eksternal dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi berada diluar sistem.
		Orang / unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak terdefiniskan.
		Aliran data dengan arah khusus dari sumber ke tujuan.
		Penyimpanan data atau tempat data dilihat oleh proses.

Sumber: Roger (2012)

Data Flow Diagram dibagi menjadi beberapa tingkatan level yaitu:

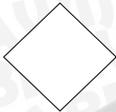
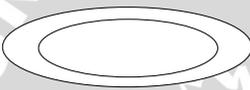
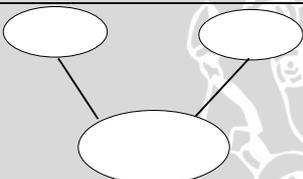
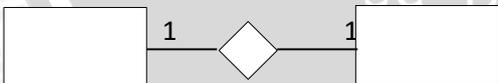
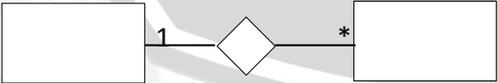
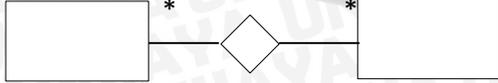
1. Level 0 atau yang disebut *context diagram* merupakan level tertinggi dari suatu *data flow diagram* (DFD).
2. Level 1 merupakan hasil dekomposisi dari *data flow diagram* (DFD) Level 0.
3. Level 2 merupakan hasil dekomposisi dari proses-proses yang ada pada *data flow diagram* (DFD) Level 1.
4. Dan seterusnya.

Untuk penggunaan *data flow diagram* pada metode COCOMO II menggunakan DFD pada level 0 dan level 1 sebagai acuan dalam menentukan tipe fungsi pengguna (Pranatha et al., 2012). Dalam menentukan analisis pada DFD dengan cara mengidentifikasi pada tipe fungsi pengguna.

2.5.9 Entity Relationship Diagram (ERD)

Menurut Brady dan Loonam (2010), *Entity Relationship Diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *system analysts* dalam tahap analisis persyaratan proyek pengembangan system (Brady, 2010). Notasi dalam ERD akan dijelaskan pada tabel 2.12 berikut:

Tabel 2.12 Notasi *Entity Relationship Diagram*

Notasi	Keterangan
	Entity merupakan basis data yaitu suatu obyek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data
	Relationship adalah hubungan antara dua jenis entitas dan direpresentasikan sebagai garis lurus yang menghubungkan dua entitas.
	Atribut memberikan informasi lebih rinci tentang jenis entitas. Atribut memiliki struktur internal berupa tipe data.
	Atribut Key adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (<i>Row/Record</i>) dalam tabel secara unik.
	Atribut Multivalued adalah nilai dari suatu atribut yang mempunyai lebih dari satu (<i>multivalued</i>) nilai dari atribut yang bersangkutan.
	Atribut Composite adalah suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu yang masih bisah dipecah lagi atau mempunyai sub atribut.
	Atribut Derivatif adalah atribut yang tidak harus disimpan dalam database atau atribut yang dihasilkan dari atribut lain atau dari suatu <i>relationship</i> .
	One to One adalah setiap anggota entitas hanya boleh berhubungan dengan satu anggota entitas lainnya.
	One to Many adalah setiap anggota entitas dapat berhubungan dengan lebih dari satu anggota entitas lainnya.
	Many to Many adalah setiap entitas dapat berhubungan dengan banyak entitas.

Sumber: Brady (2010)

Penggunaan ERD pada metode COCOMO II berguna untuk merepresentasikan penggunaan tabel dan *field-field* atau atribut dalam *database* (Pranatha et al.,



2012). Field-field ERD yang digunakan untuk menganalisis DET, RET dan FTR untuk menentukan bobot pada tabel DET, RET dan FTR.

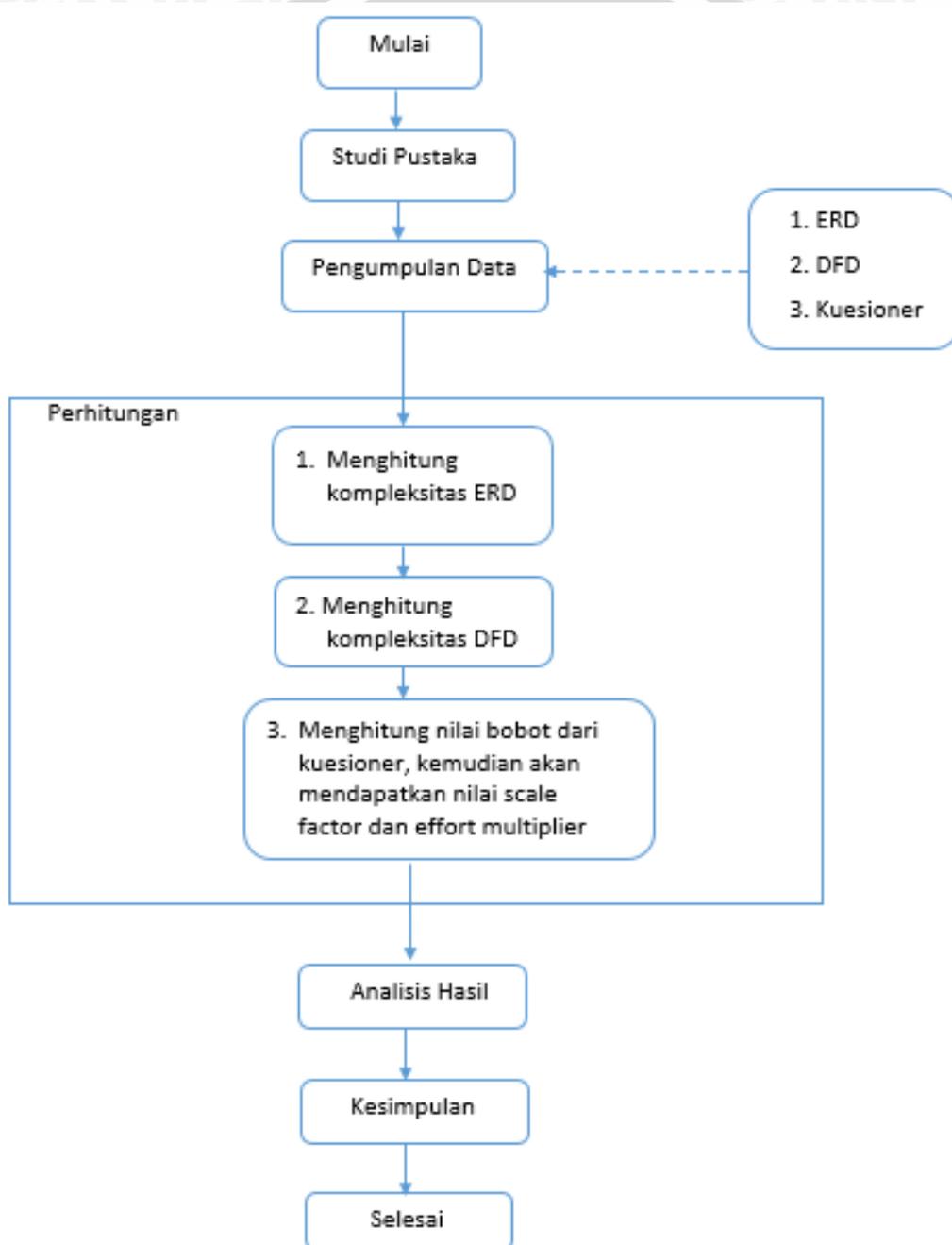
2.6 Metode *Guesstimate* / Perkiraan Harga Proyek

Metode *guesstimate* adalah metode perhitungan perkiraan yang diterapkan pada *software house* 7Tressdigital. Metode *guesstimate* digunakan untuk memberikan estimasi SDM, durasi pengerjaan dan biaya proyek dengan cara perkiraan dari *expert judgment* yang ada pada *software house* (Marom, 2012). *Expert judgment* ditentukan berdasarkan pengalaman dalam mengerjakan proyek dan memiliki jabatan minimal sebagai *project manager*.



BAB 3 METODOLOGI

Metode penelitian yang dilakukan ini dimulai dari studi pustaka, metode pengumpulan data, dilanjutkan dengan implementasi, yaitu metode perhitungan dengan COCOMO II, untuk kemudian dilakukan analisis perbandingan antara hasil perhitungan COCOMO II dengan perkiraan biaya asli dari *software house* 7Treesdigital.



Gambar 3.1 Alur penelitian



3.1 STUDI PUSTAKA

Studi pustaka dilakukan untuk mengumpulkan informasi yang bersumber dari buku, naskah penelitian, dan informasi dari internet. Studi pustaka dilakukan untuk mendapatkan informasi tentang:

1. Manajemen proyek sistem informasi meliputi penjelasan umum manajemen proyek.
2. Metode COCOMO II penjelasan umum tentang metode COCOMO II, sub-model COCOMO II dan perhitungan COCOMO II.
3. Metode WBS terkait penjabaran sebuah pekerjaan.

3.2 PENGUMPULAN DATA

Pengumpulan data merupakan tahapan penting untuk mengumpulkan data - data yang akan digunakan sebagai sumber informasi yang selanjutnya akan dilakukan analisis dari data tersebut. Data yang dibutuhkan tersebut berupa dokumen perancangan, kemudian melalui wawancara dan kuesioner yang memerlukan responden dari tim pengembang. Semua data tersebut didapatkan langsung dari *software house 7Treesdigital*.

Dari pengumpulan data tersebut yang bertujuan untuk menanyakan banyaknya sumber daya manusia, biaya proyek dan waktu pengerjaan proyek tersebut kemudian data-data tersebut diperkuat dengan data kuesioner yang memerlukan responden oleh tim pengembang. Semua data tersebut didapatkan langsung dari *software house 7Treesdigital*.

3.2.1 Dokumen Perancangan

Tahap pengumpulan data dari dokumen perancangan berupa dokumen *software requirement sistem*, dalam penelitian ini menggunakan data dari dokumen perancangan berupa ERD dan DFD. Penelitian ini memerlukan DFD *context* (level 0) dan DFD level 1 untuk kemudian dianalisis mendapatkan tipe fungsi pengguna, yang nantinya menghasilkan penentuan bobot kompleksitas dari *unadjusted functions point*.

3.2.2 Wawancara

Tahap pengumpulan data melalui cara wawancara, dalam penelitian ini membutuhkan ketersediaan dari *project manager* untuk dimintai keterangan tentang proyek yang telah dikerjakan. Wawancara ini menanyakan beberapa hal tentang proyek tersebut seperti sumber daya manusia yang dibutuhkan dalam pengerjaannya, biaya proyek dan waktu pengerjaan proyek tersebut. Bertujuan untuk mendapatkan pemahaman yang lebih baik terkait perencanaan proyek yang ada pada *software house*.

3.2.3 Kuesioner

Tahap pra-kuesioner, digunakan untuk membuat kuesioner yang nantinya akan disebarkan kepada responden yaitu tim pengembang. Tujuan dari proses pra-kuesioner ini adalah melakukan *validasi* atau uji ahli terkait konten kuesioner yang dilakukan oleh seorang *expert judgment*. Kegiatan ini dilakukan untuk mereview hasil dari kuesioner yang nantinya akan diberikan masukan untuk

perbaikan dari *expert judgment* (Schwalbe, 2014). *Expert judgment* mengidentifikasi dari berbagai poin-poin yang ada pada kuesioner kemudian memastikan kuesioner tersebut telah sesuai dengan metode COCOMO II.

Setelah tahap pra-kuesioner telah selesai, kemudian dilakukan penyebaran kuesioner kepada responden untuk mendapatkan informasi terkait perhitungan dari *scale factor* dan *effort adjustment multiplier*.

Dalam buku Barry Boehm (2000) yg berjudul *Software Cost Estimations With Cocomo II* menjelaskan terkait pengumpulan data, pengumpulan data diberikan untuk mereka (tim pengembang) yang terlibat dalam pengumpulan data agar dapat terdefinisi dengan jelas dan sesuai prosedur (Boehm, 2000). Setelah kuesioner yang telah *divalidasi* oleh *expert judgment*, kemudian kuesioner disebarakan kepada responden. Responden dalam penelitian ini adalah tim pengembang dari *software house* yang telah membuat sistem SIRIS.

3.3 ANALISIS HASIL

Analisis hasil penelitian ini dilakukan dengan metode perhitungan dengan COCOMO II. Tahapan metode perhitungan dengan COCOMO II ini adalah sebagai berikut:

Tabel 3.1 Alur Perhitungan COCOMO II

Tahap	Input	Proses	Output
1.	ERD dan DFD	1. Menganalisis kesesuaian data penyimpanan (storage) dari ERD dan DFD. 2. Identifikasi data penyimpanan tersebut kedalam lima fungsi pengguna EI, EO, ILF, EIF, EI. 3. Menentukan bobot kompleksitas berupa DET, FTR, dan RET.	Nilai untuk perhitungan Unadjusted function point
2.	Nilai UFP	Nilai UFP dikonversi kedalam SLOC (Source Line Of Code).	Nilai Estimasi Usaha
3.	Kuesioner Scale Factor	Perhitungan dari hasil kuesioner tentang Precedentedness, Development Fleksibility, Risk Resolution, Team Cohesion, Process Maturity.	Nilai Scale Factor
4.	Kuesioner Effort Adjustment Multiplier	Perhitungan dari hasil kuesioner tentang keandalan perangkat lunak (RELY), ukuran database (DATA), kompleksitas produk (CLPX), pengembangan kode yang diarahkan dalam penggunaan kembali (RUSE), dokumentasi sesuai perangkat lunak (DOCU), keandalan waktu eksekusi (TIME), keandalan penyimpanan (STOR), kompleksitas dari software dan hardware (PVOL), kemampuan analisis sistem (ACAP), kemampuan programmer (PCAP), kontinuitas personil (PCON), pengalaman tim dalam mengembangkan aplikasi (AEXP), pengalaman	Nilai Effort Adjustment Multiplier

Sumber: Diadaptasi dari Pranatha et al., (2000)



Tabel 3.1 Alur Perhitungan COCOMO II (Lanjutan)

4.	Kuesioner <i>Effort Adjustment Multiplier</i>	tim dalam pengembangan terkait platform aplikasi (<i>PEXP</i>), penilaian <i>cost driver</i> terkait <i>programmer (LTEX)</i> , penggunaan <i>software bantuan (TOOL)</i> , penilaian <i>cost driver</i> terkait perangkat lunak (<i>SITE</i>), penilaian <i>cost driver</i> terkait pengembangan proyek (<i>SCED</i>).	Nilai <i>Effort Adjustment Multiplier</i>
5.	<i>Person-Month</i>	Perhitungan <i>Person-Month</i> berupa estimasi jadwal pengerjaan dan estimasi biaya proyek.	Nilai <i>Person-Month</i>

Sumber: Diadaptasi dari Pranatha et al., (2000)

Berikut penjelasan tahapan-tahapan pada tabel 3.1:

1. Mendapatkan dokumen perancangan berupa *context* diagram beserta DFD level 1 sesuai dengan proses bisnis yang ada pada sistem SIRIS.
2. Setelah menganalisis DFD dan ERD mendapatkan nilai *Unadjusted Function Point*. kemudian menentukan bobot kompleksitas berdasar aturan *data element type (DET)*, *file type references (FTR)*, dan *record element type (RET)*. Sebelum itu, ada lima tipe fungsi pengguna perlu untuk diidentifikasi terlebih dahulu. Lima tipe fungsi dari pengguna, yaitu *External Input (EI)*, *External Outputs (EO)*, *Internal Logical File (ILF)*, *External Interface Files (EIF)*, *External Inquiry (EI)*. UFP yang telah diperoleh harus diubah kedalam SLOC supaya dapat *diconversikan* ke dalam persamaan estimasi usaha.
3. Kemudian langkah berikutnya adalah mendapat nilai *Scale Factor*. Untuk mengetahui faktor skala, telah ada beberapa parameter pengukuran disertai bobotnya yang telah dijelaskan pada bab 2. Untuk pemberian nilai terhadap masing-masing parameter tersebut membutuhkan penilaian obyektif yang didapatkan melalui penyebaran kuesioner terhadap beberapa responden, responden disini adalah tim pengembang sistem SIRIS.
4. Selanjutnya melakukan penilaian dan pemberian bobot secara obyektif terhadap parameter-parameter yang menyusun variabel *Effort Adjustment Multipliers*. Hal ini dilakukan dengan pendekatan yang sama untuk mendapatkan nilai faktor skala, yaitu melalui kuesioner yang disebar ke tim pengembang sistem SIRIS.
5. Langkah terakhir adalah perkiraan usaha/effort (*person/month*) dapat diketahui dengan perhitungan menggunakan rumus persamaan. Keseluruhan tahapan dalam metode ini akan menghasilkan perhitungan berupa besarnya usaha, biaya, dan alokasi sumber daya manusia pada sistem SIRIS.

3.4 KESIMPULAN

Kesimpulan dari penulisan ini adalah, dalam perencanaan sebuah proyek organisasi disarankan untuk merancang dokumen *work breakdown structure* terlebih dahulu berguna untuk mendetailkan setiap pengerjaan proyek. Setelah selesai pembuatan dokumen WBS, setiap organisasi dapat menerapkan metode perhitungan COCOMO II untuk menghitung estimasi SDM, biaya pengerjaan proyek serta durasi pengerjaan proyek.

BAB 4 PENGUMPULAN DATA

4.1 Mekanisme Observasi

Langkah pertama dalam melakukan penelitian ini adalah dengan melakukan observasi. Observasi dilakukan dengan melihat secara langsung tentang kondisi IT yang ada pada 7Treesdigital. Dalam pelaksanaan observasi peneliti disini bertujuan melihat cara kerja sistem yang akan digunakan dalam object penelitian tersebut, peneliti memperhatikan segala proses yang bekerja dalam Sistem Informasi Rekam Medis (SIRIS).

4.2 Hasil Observasi

Hasil dari melakukan observasi pada SIRIS akan dijelaskan sebagai berikut:

1. Berikut definisi fitur-fitur yang ada pada SIRIS, mengacu pada dokumen perancangan sebagai berikut:
 - a) Halaman Home
 - b) Halaman Dokter
 - c) Halaman Pasien
 - d) Kategori Penyakit
 - e) Artikel
 - f) Prediksi (Demam)
2. Mengetahui kebutuhan-kebutuhan fungsional pada sistem SIRIS, mengacu pada dokumen perancangan sebagai berikut:
 - a) Pengguna: Pasien

Tabel 4.1 Pengguna Pasien

No	Nama Fungsi	Deskripsi
1	Lihat Profil Pasien Sendiri	Pasien melihat data diri pribadi berupa: Username, Nama, Riwayat Kesehatan, Alamat, Telepon Rumah / HP, Catatan dari Dokter, Golongan darah, Jenis Kelamin, Berat Badan, dan Tinggi Badan.
2	Lihat Detail Penyakit Pasien	Pasien memilih melihat detail penyakit pasien pada riwayat kesehatan yang berisi: Tanggal dan waktu, dan Nama penyakit.

Sumber: 7treesdigital (2014)

- b) Pengguna: Dokter

Tabel 4.2 Pengguna Dokter

No	Nama Fungsi	Deskripsi
1	Lihat Profil Dokter	Dokter melihat data diri, berupa: Nama, Spesialis, Ruangan, Alamat, dan Nomor Telepon / HP.

Sumber: 7treesdigital (2014)

Tabel 4.2 Pengguna Dokter (Lanjutan)

No	Nama Fungsi	Deskripsi
2	Update Penyakit Pasien	Dokter menambah daftar penyakit pasien dan secara otomatis menambah riwayat kesehatan pasien.
3	Update Catatan Pasien	Dokter melakukan edit catatan khusus untuk pasien
4	Update Daftar Penyakit	Dokter menambah daftar penyakit baru ke dalam daftar penyakit.

Sumber: 7treesdigital (2014)

c) Pengguna: Admin

Tabel 4.3 Pengguna Admin

No	Nama Fungsi	Deskripsi
1	Pendaftaran Pasien dan Dokter	Admin menambahkan informasi tentang pengguna lain (pasien atau dokter) yang ingin mendaftar menjadi pasien atau dokter di rumah sakit dan memberikan username serta password kepada pengguna (pasien dan dokter).
2	Update Artikel	Admin menambah atau menghapus artikel

Sumber: 7treesdigital (2014)

d) Pengguna: Dokter, dan Admin

Tabel 4.4 Pengguna Dokter dan Admin

No	Nama Fungsi	Deskripsi
1	Lihat Daftar Pasien	Dokter dan Admin melihat daftar pasien pada rumah sakit dan riwayat kesehatan pasien tersebut.

Sumber: 7treesdigital (2014)

e) Pengguna: Dokter, dan Pasien

Tabel 4.5 Pengguna Dokter dan Pasien

No	Nama Fungsi	Deskripsi
1	Update Password	Dokter, Pasien melakukan update passwordnya sendiri.
2	Update Profil	Dokter, Pasien melakukan update profilnya sendiri, berupa : Nama, Alamat, dan Telepon Rumah / HP.

Sumber: 7treesdigital (2014)

f) Pengguna: Dokter, Pasien, dan Admin

Tabel 4.6 Pengguna Dokter, Pasien dan Admin

No	Nama Fungsi	Deskripsi
1	Login	Dokter, Pasien serta Admin memasukan username dan password.

Sumber: 7treesdigital (2014)

g) Pengguna: Dokter, Pasien, Admin, dan Pengunjung

Tabel 4.7 Pengguna Dokter, Pasien, Admin, dan Pengunjung

No	Nama Fungsi	Deskripsi
1	Lihat Artikel	Dokter, Pasien, Admin dan Pengunjung dapat melihat artikel tentang penyakit.
2	Lihat Daftar Dokter	Dokter, Pasien, Admin dan Pengunjung dapat melihat informasi tentang Dokter yang bekerja di rumah sakit seperti nama dan keterangan spesialis penyakitnya.
3	Akses Sistem Prediksi (Demam)	Dokter, Pasien, Admin dan Pengunjung memasukkan gejala penyakit, kemudian akan ditampilkan deskripsi lengkap tentang penyakit tersebut serta pengobatan dan rekomendasinya.
4	Lihat Daftar Penyakit	Dokter, Pasien, Admin dan Pengunjung melihat keterangan penyakit: Nama Penyakit, Kategori Penyakit, Daftar nama dokter spesialis dan Gejala Penyakit.

Sumber: 7treesdigital (2014)

4.3 Mekanisme Kuesioner

Untuk mekanisme penyebaran kuesioner pada penelitian ini menggunakan langkah-langkahnya sebagai berikut:

1. Pra-kuesioner
2. Validasi kuesioner
3. Penentuan responden
4. Penyebaran kuesioner
5. Hasil perhitungan

4.4 Hasil Kuesioner

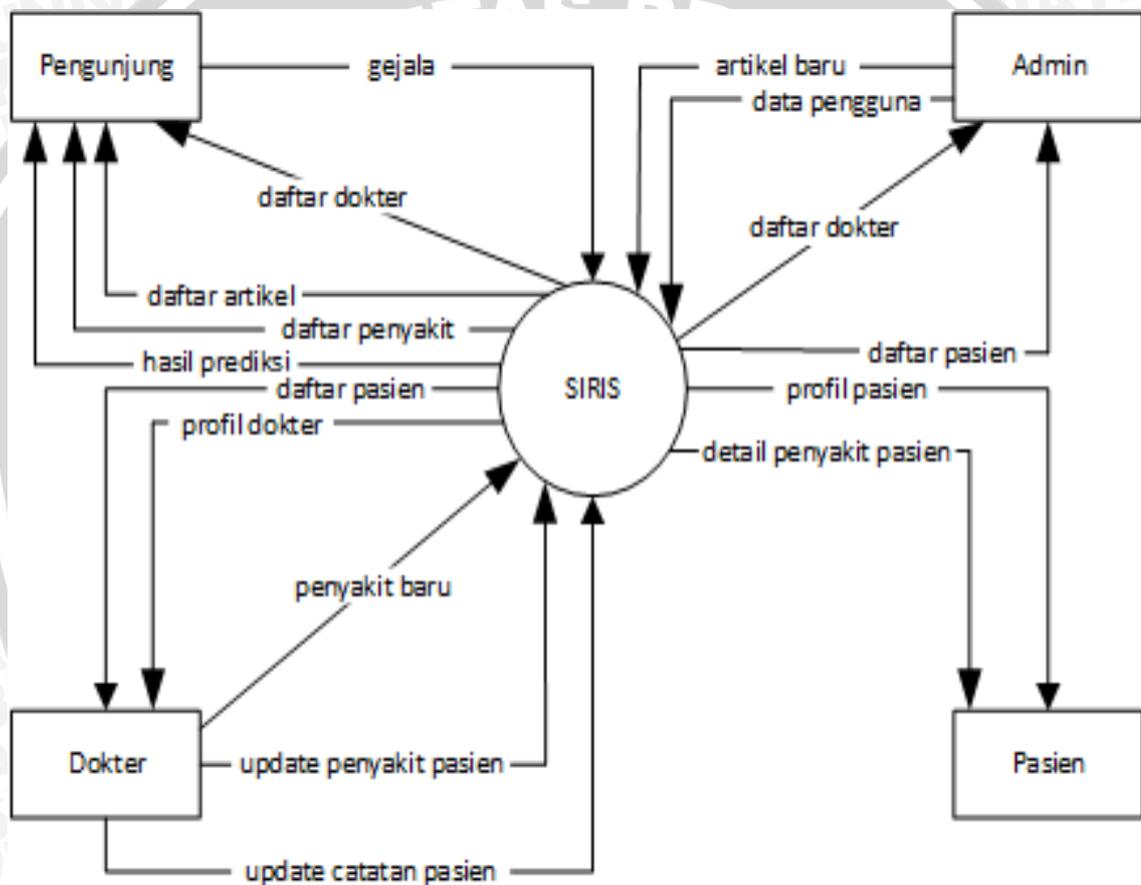
Dalam mekanisme kuesioner tersebut, maka mendapatkan hasil berupa nilai perhitungan dari *scale factor* dan *effort adjustment multiplier*. Pada kuesioner untuk perhitungan *scale factor* meliputi *Precendentedness* (PREC), *Development Fleksibility* (FLEX), *Risk Resolution* (RESL), *Team Cohesion* (TEAM), *Process Maturity* (PMAT) dalam kelima faktor tersebut menjelaskan tentang faktor-faktor yang diperlukan dalam pembangunan sebuah sistem. Jenis penilaian dalam *scale factor* berupa *Very Low* (VL), *Low* (L), *Nominal* (N), *High* (H), *Very High* (VH) dan *Extra High* (EH).

Sedangkan pada perhitungan *Effort Adjustment Multiplier* meliputi keandalan perangkat lunak (RELY), ukuran *database* (DATA), kompleksitas produk (CLPX), pengembangan kode yang diarahkan dalam penggunaan kembali (RUSE), dokumentasi sesuai perangkat lunak (DOCU), keandalan waktu eksekusi (TIME), keandalan penyimpanan (STOR), kompleksitas dari *software* dan *hardware* (PVOL), kemampuan analisis sistem (ACAP), kemampuan *programmer* (PCAP), kontinuitas personil (PCON), pengalaman tim dalam mengembangkan aplikasi (AEXP), pengalaman tim dalam pengembangan terkait *platform* aplikasi (PEXP), penilaian *cost driver* terkait *programmer* (LTEX), penggunaan *software* bantuan (TOOL),

penilaian *cost driver* terkait perangkat lunak (SITE), penilaian *cost driver* terkait pengembangan proyek (SCED) dalam ketujuh belas faktor tersebut menjelaskan tentang keandalan sebuah sistem. Jenis penilaian dalam *Effort Adjustment Multiplier* berupa *Very Low* (VL), *Low* (L), *Nominal* (N), *High* (H), *Very High* (VH) dan *Extra High* (EH).

4.5 Dokumen Perancangan

Dalam dokumen perancangan ini, peneliti melakukan observasi langsung ke *software house* 7Treesdigital. Dokumen yang dibutuhkan penelitian ini dalam menghitung estimasi pengerjaan proyek adalah dokumen DFD dan ERD dari sistem informasi rekam medis (SIRIS). Berikut adalah dokumen DFD dan ERD dari sistem SIRIS:



Gambar 4.1 context diagram SIRIS

Sumber: 7treesdigital (2014)

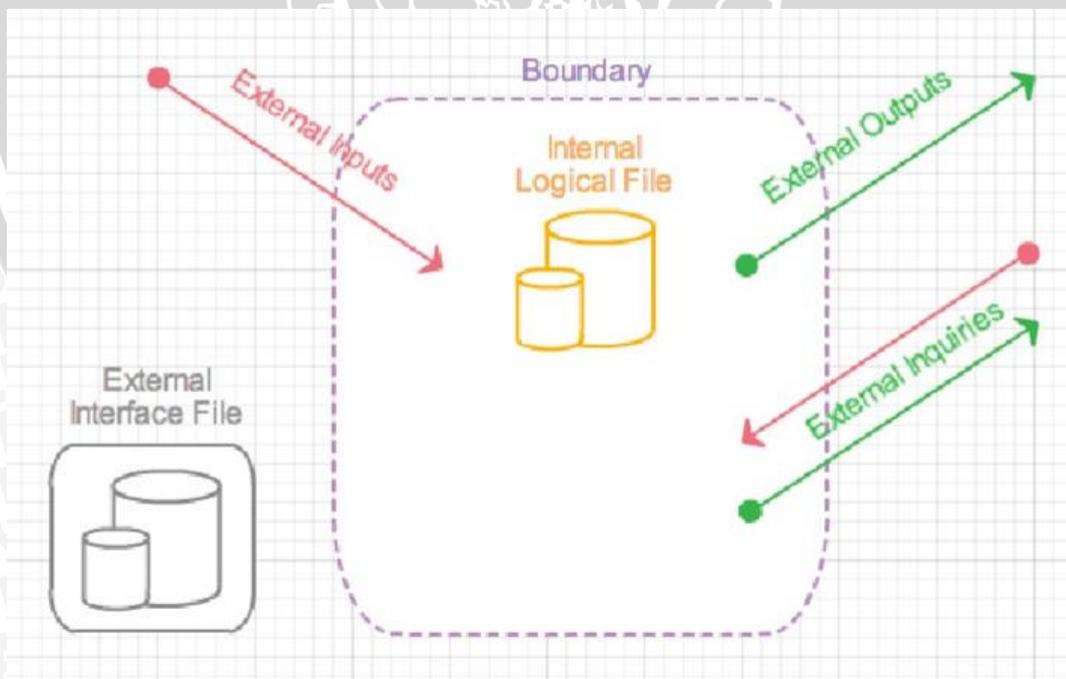
Keterangan gambar 4.1:

1. User pengunjung dapat mengakses gejala penyakit melalui sistem SIRIS.
2. User dokter dapat menambahkan dan memperbarui penyakit baru, penyakit pasien dan menambahkan catatan untuk pasien pada sistem SIRIS.
3. User admin dapat menambahkan artikel baru dan pengguna baru pada sistem SIRIS.

Keterangan gambar 4.2:

1. Pada proses 1 (Login), melakukan login kedalam sistem SIRIS dengan menginputkan *username* dan *password* kemudian akan validasi pengguna dan akan masuk ke halaman *home* sebagai pengguna sistem.
2. Pada proses 2 (Akses Artikel), pada proses ini dapat menampilkan daftar artikel yang tersedia pada sistem SIRIS.
3. Pada proses 3 (Update Penyakit Baru), pada proses ini dapat memperbarui penyakit baru kedalam sistem SIRIS setelah melakukan validasi penyakit.
4. Pada proses 4 (Akses Pasien), pada proses ini menerima masukkan berupa data penyakit dan id pasien kemudian akan diverifikasi data tersebut untuk disimpan kedalam *database* pasien setelah itu akan menampilkan detail dari pasien.
5. Pada proses 5 (Akses Dokter), pada proses ini menerima masukkan berupa id dokter yang kemudian akan diverifikasi data tersebut untuk disimpan kedalam *database* dokter. Setelah data tersebut selesai disimpan kemudian akan menampilkan detail dokter pada sistem SIRIS.
6. Pada proses 6 (Pendaftaran Pengguna), pada proses ini nenerima masukkan dari data pengguna yang akan menghasilkan data pasien.
7. Pada proses 7 (Sistem Prediksi Demam), pada proses ini nenerima masukkan dari data gejala kemudian akan diolah pada data training yang akan menghasilkan berupa hasil prediksi penyakit.

Untuk menganalisis tipe fungsi pengguna pada DFD level 1 harus memahami komponen-komponen yang ada terlebih dahulu. Komponen-komponen dalam tipe fungsi pengguna akan dijabarkan pada gambar 4.3 berikut:

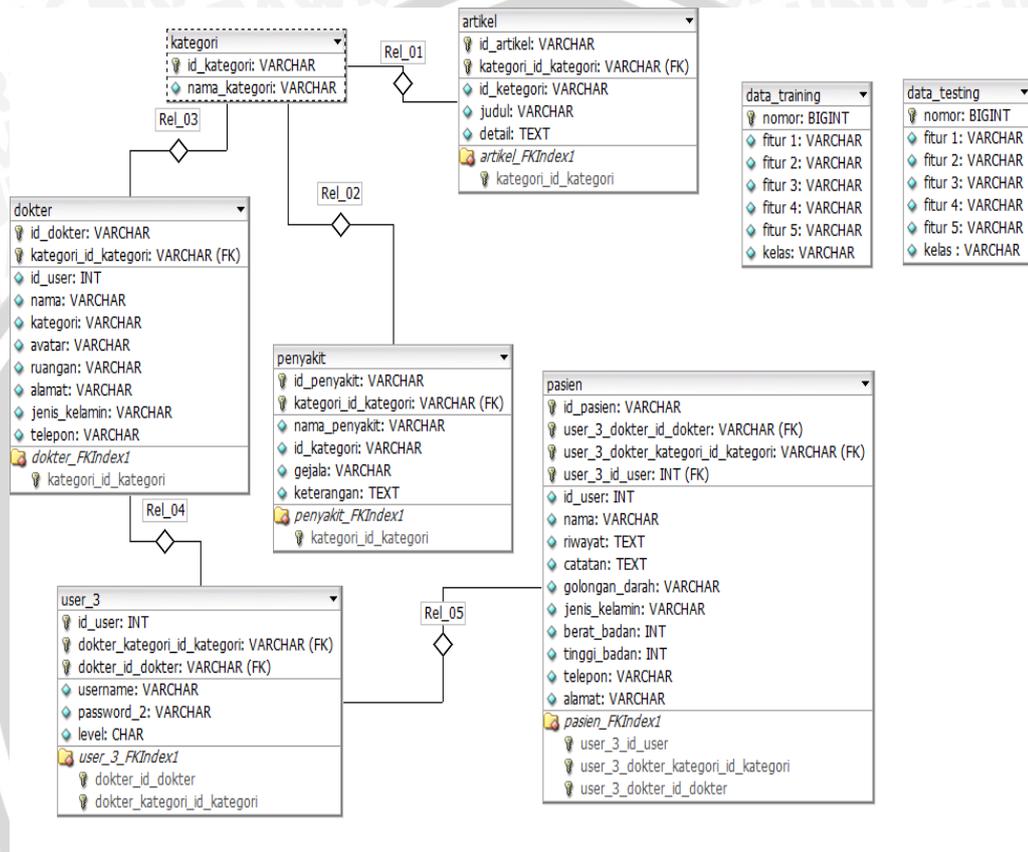


Gambar 4.3 Komponen Tipe Fungsi Pengguna

Sumber: Litvinenko (2010)

Keterangan gambar 4.3:

1. *External Input* adalah informasi yang mendapat masukkan dari pengguna.
2. *External Output* adalah informasi yang dihasilkan oleh sistem.
3. *External Inquiry* adalah informasi data yang tidak dapat dikirimkan.
4. *Internal Logical Files* adalah data yang dapat dipelihara secara internal.
5. *External Logical Files* adalah data yang dapat dipelihara secara external dengan bantuan aplikasi lain.



Gambar 4.4 Entity Relationship Diagram SIRIS

Sumber: 7treesdigital (2014)

Keterangan gambar 4.4:

1. *Database Dokter* menampilkan id user, nama dokter, kategori, avatar, ruangan, alamat, jenis kelamin hingga nomor telepon yang akan ditampilkan pada sistem.
2. *Database user_3* menampilkan id user, dokter kategori, id dokter, username, password, dan level yang akan mendapatkan masukkan dari pengguna.
3. *Database pasien* menampilkan id pasien, id dokter, id kategori, id user, nama, riwayat, catatan, golongan darah, jenis kelamin, berat badan, tinggi badan, nomor telepon, alamat yang akan mendapatkan masukkan dari pasien.
4. *Database penyakit* menampilkan id penyakit, id kategori, nama penyakit, gejala, keterangan yang akan mendapatkan masukkan dari pengguna.
5. *Database kategori* akan menampilkan data id kategori dan nama kategori yang akan mengelompokkan penyakit berdasarkan setiap kategorinya.

6. *Database* artikel akan menampilkan dan menyimpan data id artikel, id kategori, judul, detail penjelasan dari artikel yang ditampilkan pada sistem SIRIS.
7. *Database* data training akan menampilkan dan menyimpan data masukan dari pengguna berupa data nomor, fitur 1 hingga fitur 5 pada sistem untuk mengetahui gejala yang diderita oleh pasien.
8. *Database* data testing akan menampilkan dan menyimpan data masukan dari pengguna berupa data nomor, fitur 1 hingga fitur 5 pada sistem untuk mengetahui gejala yang diderita oleh pasien.



BAB 5 ANALISIS DAN PERHITUNGAN

5.1 Perhitungan *Function Point*

Function Point adalah pengukuran suatu proyek perangkat lunak dengan cara mengkuantisasi yang berarti mengkonversi kegunaan pemrosesan informasi yang berhubungan dengan tipe data berkas, data keluaran, masukan data maupun kontrol eksternal. Nilai dalam pembobotan tiap fungsi pada *unadjusted function point* didapatkan dari hasil analisis dari DFD dari sistem SIRIS.

Merujuk pada gambar 4.1 yang terlampir pada bab 4, menunjukkan DFD context diagram (level 0) dari sistem SIRIS. Untuk mendapatkan aliran data sebuah informasi yang lebih lengkap, maka diperlukan proses dekomposisi dari *level context diagram* tersebut menjadi *data flow diagram* (level 1) yang sudah tertera pada gambar 4.2. Dekomposisi *data flow diagram* terbatas pada level 1 sebagai acuan dalam menentukan tipe fungsi pengguna sesuai pendekatan dari contoh perhitungan manual *function point* yang terdapat pada bab II bagian 2.5.3 terkait *data flow diagram*.

5.1.1 Analisis Perhitungan Total *Unadjusted Function Point* SIRIS

Berikut ini adalah hasil dari analisis terkait perhitungan *unadjusted function point* berdasarkan analisis pada *data flow diagram* dan *database* dari sistem SIRIS yang menghasilkan identifikasi pada lima tipe fungsi pengguna. Adapun dalam analisis ini tidak terdapat *External Interface File* (EIF) dan *External Inquiry* (EQ), karena pada sistem SIRIS tidak ditemukan berkas eksternal (EIF) dan nilai masukan yang menghasilkan keluaran seketika (EQ). Untuk hasil identifikasi tiap tipe fungsi yang lain akan dijelaskan sebagai berikut:

a) *Internal Logical File* (ILF) adalah berupa data penyimpanan (*database*) yang ada pada sistem yang dianalisis dari DFD, berikut hasil analisis dari sistem SIRIS:

1. Data Training
2. Data Artikel
3. Data Kategori
4. Data Pasien
5. Data Penyakit
6. Data Dokter

Dari hasil analisis pada tipe fungsi pengguna *Internal Logical File* (ILF) menghasilkan 6 poin tipe data pengguna terdiri dari data training, data artikel, data kategori, data pasien, data penyakit, data dokter. Kemudian mengelompokkan *attribute* pada *database* sesuai penjelasan pada pengertian ILF, berikut hasil dari analisisnya ada pada tabel 5.1:

Tabel 5.1 Pengelompokan Tipe Fungsi Pengguna (ILF)

No	Tipe Data Pengguna	Dibuat (<i>database</i>)	Dirawat (<i>database</i>)
1	Data Training	Kelas	Fitur 1
2	Data Artikel	Judul	Fitur 2
3	Data Kategori	Detail	Fitur 3
4	Data Pasien	Nama_kategori	Fitur 4
5	Data Penyakit	Nama_penyakit	Fitur 5
6	Data Dokter	Gejala	ID_penyakit
		Keterangan	ID_pasien
		Nama	ID_dokter
		Riwayat	ID_user
		Catatan	ID_artikel
		Golongan_darah	ID_kategori
		Jenis_kelamin	
		Berat_badan	
		Tinggi_badan	
		Telepon	
		Alamat	
		Kategori	
		Avatar	
		Ruangan	

b) *External Input* (EI) adalah nilai setiap data yang berasal dari masukkan oleh pengguna yang dianalisis dari DFD dan SIRIS, berikut hasil analisis dari sistem SIRIS:

1. Data Training
2. Data Artikel
3. Data Kategori
4. Data Pasien
5. Data Penyakit
6. Data Dokter

Dari hasil analisis pada tipe fungsi pengguna *External Input* (EI) menghasilkan 6 poin tipe data pengguna terdiri dari data training, data artikel, data kategori, data pasien, data penyakit, data dokter. Kemudian mengelompokkan *attribute* pada *database* sesuai penjelasan pada pengertian EI, berikut hasil dari analisisnya ada pada tabel 5.2:

Tabel 5.2 Pengelompokan Tipe Fungsi Pengguna (EI)

No	Tipe Data Pengguna	Menambah Data (<i>database</i>)	Mengubah Data (<i>database</i>)
1	Data Training	Kelas	Fitur 1
2	Data Artikel	Judul	Fitur 2
3	Data Kategori	Detail	Fitur 3
4	Data Pasien	Nama_kategori	Fitur 4
5	Data Penyakit	Nama_penyakit	Fitur 5
6	Data Dokter	Gejala	Golongan_darah
		Keterangan	Jenis_kelamin
		Nama	Berat_badan
		Riwayat	Tinggi_badan
		Catatan	Telepon
		Kategori	Alamat
		ID_penyakit	Avatar
		ID_pasien	Ruangan
		ID_dokter	
		ID_user	
		ID_artikel	
		ID_kategori	

c) *External Output* (EO) adalah nilai keluaran yang dihasilkan setelah mendapatkan inputan dari pengguna yang dianalisis dari DFD dan SIRIS, berikut hasil analisis dari sistem SIRIS:

1. Data Training
2. Data Artikel
3. Data Kategori
4. Data Pasien
5. Data Penyakit
6. Data Dokter

Dari hasil analisis pada tipe fungsi pengguna *External Output* (EO) menghasilkan 6 poin tipe data pengguna terdiri dari data training, data artikel, data kategori, data pasien, data penyakit, data dokter. Kemudian mengelompokkan *attribute* pada *database* sesuai penjelasan pada pengertian EO, berikut hasil dari analisisnya ada pada tabel 5.3:

Tabel 5.3 Pengelompokkan Tipe Fungsi Pengguna (EO)

No	Tipe Data Pengguna	Menghasilkan Keluaran (<i>database</i>)
1	Data Training	Kelas
2	Data Artikel	Judul
3	Data Kategori	Detail
4	Data Pasien	Nama_kategori
5	Data Penyakit	Nama_penyakit
6	Data Dokter	Gejala Keterangan Nama Riwayat Catatan Kategori ID_penyakit ID_pasien ID_dokter ID_user ID_artikel ID_kategori Golongan_darah Jenis_kelamin Berat_badan Tinggi_badan Telepon Alamat Avatar Ruangan Fitur 1 Fitur 2 Fitur 3 Fitur 4 Fitur 5

d) *External Interface Files* (EIF) adalah berupa berkas eksternal yang dibagikan dari sistem yang dianalisis dari DFD dan SIRIS.

e) *External Inquiry* (EQ) adalah berupa nilai masukkan yang menghasilkan keluaran seketika dari sistem yang dianalisis dari DFD dan SIRIS.



Setelah itu untuk menentukan bobot kompleksitas *unadjusted function point* dari masing-masing tipe fungsi pengguna harus menyesuaikan dengan aturan *Data Element Type (DET)*, *Record Element Type (RET)*, *File Type Reference (FTR)* yang ditunjukkan pada tabel 2.1. Kemudian diklasifikasikan DET, RET dan FTR berdasarkan analisis dari *data flow diagram* dan *database* dari sistem SIRIS yang telah akan ditunjukkan pada tabel 5.4 berikut adalah penentuan bobot kompleksitas tiap tipe fungsi pengguna:

Tabel 5.4 Penentuan bobot RET, DET dan FTR pada UFP

Fungsi	RET	DET	FTR
ILF	Data training	Fitur 1	
	Data Artikel	Fitur 2	
	Data Kategori	Fitur 3	
	Data Penyakit	Fitur 4	
	Data Pasien	Fitur 5	
	Data Dokter	Kelas	
		ID_artikel	
		ID_kategori	
		Judul	
		Detail	
		Nama_kategori	
		ID_penyakit	
		Nama_penyakit	
		Gejala	
		Keterangan	
		ID_pasien	
		ID_dokter	
		ID_user	
		Nama	
		Riwayat	
	Catatan		
	Golongan_darah		
	Jenis_kelamin		
	Berat_badan		
	Tinggi_badan		
	Telepon		
	Alamat		
	Kategori		

Tabel 5.4 Penentuan bobot RET, DET dan FTR pada UFP (Lanjutan)

Fungsi	RET	DET	FTR
		Avatar Ruangan	
	Total 6 RET	Total 30 DET	Total 0 FTR
EI		Nama_penyakit Gejala Keterangan ID_pasien ID_dokter ID_user Nama Riwayat Catatan Golongan_darah Fitur 1 Fitur 2 Fitur 3 Fitur 4 Fitur 5 Kelas ID_artikel ID_kategori Judul Detail Nama_kategori ID_penyakit Jenis_kelamin Berat_badan Tinggi_badan Telepon Alamat Kategori Avatar Ruangan	Data training Data Artikel Data Kategori Data Penyakit Data Pasien Data Dokter
	Total 0 RET	Total 30 DET	Total 6 FTR

Tabel 5.4 Penentuan bobot RET, DET dan FTR pada UFP (Lanjutan)

EO	ID_pasien	Data training	
	ID_dokter	Data Artikel	
	ID_user	Data Kategori	
	Nama	Data Penyakit	
	Riwayat	Data Pasien	
	Catatan	Data Dokter	
	Golongan_darah		
	Jenis_kelamin		
	Berat_badan		
	Tinggi_badan		
	Telepon		
	Alamat		
	Kategori		
	Kelas		
	ID_artikel		
	ID_kategori		
	Judul		
	Detail		
	Nama_kategori		
	ID_penyakit		
Nama_penyakit			
Gejala			
Keterangan			
Avatar			
Ruangan			
Fitur 1			
Fitur 2			
Fitur 3			
Fitur 4			
Fitur 5			
	Total 0 RET	Total 30 DET	Total 6 FTR
EIF	-	-	-
	Total 0 RET	Total 0 DET	Total 0 FTR
EQ	-	-	-
	Total 0 RET	Total 0 DET	Total 0 FTR

Dari penjabaran hasil analisis penentuan bobot kompleksitas dari tiap fungsi pengguna didapatkan keseluruhan perhitungan UFP yang akan dijelaskan pada tabel 5.5 sebagai berikut:

Tabel 5.5 Perhitungan *Unadjusted Function Points* (UFP)

Tipe Fungsi Pengguna	Jumlah Tipe Fungsi Pengguna	Perhitungan Bobot Kompleksitas UFP			Bobot Kompleksitas Fungsi Pengguna	Nilai Kompleksitas Fungsi Pengguna	UFP
		DET	RET	FTR			
ILF	6	6	30	-	High	15	90
EI	6	-	30	6	High	6	36
EO	6	-	30	6	High	7	42
EIF	-	-	-	-	Low	5	0
EQ	-	-	-	-	Low	3	0
Total UFP							168

5.1.2 Konversi Hasil *Unadjusted Function Point* menjadi *Kilo Line of Code*

Hasil analisis dari tiap tipe fungsi dari sistem SIRIS, sehingga didapatkan hasil *unadjusted function point*. Hasil dari perhitungan *unadjusted function point* (UFP) kemudian dikalikan dengan 40, nilai 40 didapatkan dari standart konversi dari *Source Line Of Code* untuk bahasa HTML yang merupakan bahasa pemrograman yang digunakan oleh tim pengembang untuk mengembangkan sistem SIRIS. Setelah didapatkan nilai dari *unadjusted function point* (UFP) dalam satuan *Source Line Of Code* (SLOC), dikonversi nilai tersebut menjadi *Kilo Line Of Source Code* (KLOC), dengan cara dari SLOC dibagi 1000 sehingga didapatkan hasil KLOC.

Tabel 5.6 Hasil Perhitungan *Function Point*

Analisis <i>Function Point</i>								
Nama Sistem	ILF	EI	EO	EIF	EQ	UFP	SLOC	KLOC
SIRIS	90	36	42	-	-	168	6,720	6,72
Total KLOC								6,72

5.2 Perhitungan *Scale Factor*

Pada perhitungan *scale factor* ini dilakukan penilaian melalui penyebaran kuesioner yang telah divalidasi melalui *expert judgment*, perhitungan ini lebih menekankan objektivitas dari inputan user berupa nilai pembobotan disetiap parameternya. Dalam pelaksanaan penyebaran kuesioner ini melibatkan responden anggota dari tim pengembang proyek sistem SIRIS yang tergabung dalam 7Treesdigital. Rekap hasil penyebaran kuesioner *scale factor* (lampiran D). Hasil kuesioner *scale factor* digunakan untuk menghitung nilai *scale factor* yang ditampilkan pada tabel 5.7 berikut:

Tabel 5.7 Hasil Penilaian *Scale Factor*

Faktor Skala	Rating Level	Hasil Penilaian
Precedentedness	VERY HIGH, HIGH, LOW, HIGH, NOMINAL	$(0,5 \times 1,24) + (0,5 \times 2,48) + (0,5 \times 4,96) + (0,5 \times 2,48) + (0,5 \times 3,72) = 7,44$
Flexibility	VERY HIGH, NOMINAL, NOMINAL, NOMINAL, HIGH	$(0,5 \times 1,01) + (0,5 \times 3,04) + (0,5 \times 3,04) + (0,5 \times 3,04) + (0,5 \times 2,03) = 6,08$
Risk Resolution	LOW, NOMINAL, LOW, NOMINAL, LOW	$(0,5 \times 5,65) + (0,5 \times 4,24) + (0,5 \times 5,65) + (0,5 \times 4,24) + (0,5 \times 5,65) = 12,715$
Team Cohesion	VERY HIGH, VERY HIGH, HIGH, VERY HIGH, VERY HIGH	$(0,5 \times 1,10) + (0,5 \times 1,10) + (0,5 \times 2,19) + (0,5 \times 1,10) + (0,5 \times 1,10) = 3,295$
Process Maturity	NOMINAL, LOW, NOMINAL, LOW, NOMINAL	$(0,5 \times 4,68) + (0,5 \times 6,24) + (0,5 \times 4,68) + (0,5 \times 6,24) + (0,5 \times 4,68) = 13,26$

Dengan menggunakan rumus persamaan 2.4, maka didapatkan hasil dari *scale factor* (E) sebagai berikut:

$$E = 0,91 + 0,01 \times (7,44 + 6,08 + 12,715 + 3,295 + 13,26) = 1,33$$

Menggunakan rumus persamaan 2.4 didapatkan nilai 0,91 dan 0,01 berupa angka ketentuan dari COCOMO II, kemudian didapatkan nilai 7,44+6,08+12,715 + 3,295 + 13,26 yang berasal dari penyebaran kuesioner kemudian dikalikan dengan 0,5 nilai ketentuan dari COCOMO II. Setelah itu menghasilkan nilai *scale factor* 1,33.

5.3 Perhitungan *Effort Adjustment Multiplier*

Data mengenai *effort adjustment multiplier* atau bisa disebut dengan faktor penggerak biaya penilaian ini dikumpulkan melalui metode penyebaran kuesioner yang sama dilakukan pada faktor skala. Pelaksanaan kuesioner ini melibatkan responden dari anggota tim pengembang dari 7Treesdigital. Kuesioner dari *effort adjustment multiplier* (lampiran C), kemudian untuk hasil rekap dari penyebaran kuesioner *effort adjustment multiplier* terlampir pada (lampiran D). Pada tabel 5.8 berikut adalah hasil dari penyebaran kuesioner *effort adjustment multiplier*:

Tabel 5.8 Hasil penilaian *Effort Adjustment Multiplier*

Faktor Skala	Rating Level	Hasil Penilaian
RELY	LOW, VERY HIGH, HIGH, VERY HIGH, LOW	$(0,5 \times 0,92) + (0,5 \times 1,26) + (0,5 \times 1,1) + (0,5 \times 1,26) + (0,5 \times 0,92) = 2,73$
DATA	NOMINAL, NOMINAL, NOMINAL, NOMINAL, NOMINAL	$(0,5 \times 1) + (0,5 \times 1) + (0,5 \times 1) + (0,5 \times 1) + (0,5 \times 1) = 2,5$
CPLX	HIGH, LOW, LOW, LOW, HIGH	$(0,5 \times 1,17) + (0,5 \times 0,87) + (0,5 \times 0,87) + (0,5 \times 0,87) + (0,5 \times 1,17) = 2,475$
RUSE	EXTRA HIGH, VERY HIGH, NOMINAL, VERY HIGH, VERY HIGH	$(0,5 \times 1,24) + (0,5 \times 1,15) + (0,5 \times 1) + (0,5 \times 1,15) + (0,5 \times 1,15) = 2,845$

Tabel 5.8 Hasil penilaian *Effort Adjustment Multiplier* (Lanjutan)

DOCU	NOMINAL, NOMINAL, NOMINAL, NOMINAL, NOMINAL	$(0,5 \times 1) + (0,5 \times 1) + (0,5 \times 1) + (0,5 \times 1) + (0,5 \times 1) = 2,5$
TIME	NOMINAL, NOMINAL, NOMINAL, NOMINAL, NOMINAL	$(0,5 \times 1) + (0,5 \times 1) + (0,5 \times 1) + (0,5 \times 1) + (0,5 \times 1) = 2,5$
STOR	NOMINAL, NOMINAL, HIGH, NOMINAL, NOMINAL	$(0,5 \times 1) + (0,5 \times 1) + (0,5 \times 1,05) + (0,5 \times 1) + (0,5 \times 1) = 2,525$
PVOL	VERY HIGH, LOW NOMINAL, LOW, LOW	$(0,5 \times 1,3) + (0,5 \times 0,87) + (0,5 \times 1) + (0,5 \times 0,87) + (0,5 \times 0,87) = 2,455$
ACAP	NOMINAL, LOW, NOMINAL, LOW, VERY HIGH	$(0,5 \times 1) + (0,5 \times 1,19) + (0,5 \times 1) + (0,5 \times 1,19) + (0,5 \times 0,71) = 2,545$
PCAP	VERY HIGH, VERY HIGH, VERY HIGH, VERY HIGH, VERY HIGH	$(0,5 \times 1,29) + (0,5 \times 1,29) + (0,5 \times 1,29) + (0,5 \times 1,29) + (0,5 \times 1,29) = 1,9$
PCON	VERY LOW, NOMINAL, VERY LOW, NOMINAL, VERY LOW	$(0,5 \times 1,29) + (0,5 \times 1) + (0,5 \times 1,29) + (0,5 \times 1) + (0,5 \times 1,29) = 2,935$
AEXP	LOW, LOW, VERY LOW, LOW, NOMINAL	$(0,5 \times 1,1) + (0,5 \times 1,1) + (0,5 \times 1,22) + (0,5 \times 1,1) + (0,5 \times 1) = 2,76$
PEXP	HIGH, LOW, VERY LOW, LOW, VERY HIGH	$(0,5 \times 0,91) + (0,5 \times 1,09) + (0,5 \times 1,19) + (0,5 \times 1,09) + (0,5 \times 0,85) = 2,565$
LTEX	NOMINAL, VERY LOW, NOMINAL, VERY LOW, VERY HIGH	$(0,5 \times 1) + (0,5 \times 1,20) + (0,5 \times 1) + (0,5 \times 1,20) + (0,5 \times 0,84) = 2,62$
TOOL	VERY LOW, NOMINAL, LOW, NOMINAL, LOW	$(0,5 \times 1,17) + (0,5 \times 1) + (0,5 \times 1,09) + (0,5 \times 1) + (0,5 \times 1,09) = 2,675$
SITE	NOMINAL, HIGH, HIGH, HIGH, LOW	$(0,5 \times 1) + (0,5 \times 0,93) + (0,5 \times 1,93) + (0,5 \times 0,93) + (0,5 \times 1,09) = 2,44$
SCED	VERY LOW, NOMINAL, LOW, NOMINAL, NOMINAL	$(0,5 \times 1,43) + (0,5 \times 1) + (0,5 \times 1,14) + (0,5 \times 1) + (0,5 \times 1) = 2,785$

5.4 Perhitungan Estimasi Usaha (*Person-Month*)

Perhitungan Estimasi Usaha atau perkiraan waktu yang dibutuhkan untuk pengerjaan sistem SIRIS ini, dapat diketahui dengan cara menggunakan persamaan 2.5 sebagai berikut:

$$PM = 2,94 \times (6,72)^{1,33} \times 2,5 = 92,61$$

Menggunakan rumus persamaan 2.5, nilai 2,94 berupa angka ketentuan dari COCOMO II, nilai 6,72 didapatkan dari nilai UFP yang telah dikonversi ke KLOC, nilai 1,33 didapatkan dari nilai *scale factor*, nilai 2,5 didapatkan melalui penyebaran kuesioner *effort adjusted multiplier* kemudian dihitung rata-rata. Setelah semua selesai dijumlahkan maka mendapatkan 92,61 nilai PM.

Hasil dari *Person-Month* terdiri dari 152 jam waktu kerja yang didapatkan dari buku COCOMO II, sehingga dalam proyek ini memiliki waktu kerja sebanyak 14076 jam.

5.5 Perhitungan Estimasi Biaya

Dalam perhitungan perkiraan suatu biaya sistem SIRIS, sebelumnya harus mendapatkan hasil estimasi dari biaya total, estimasi usaha dalam satuan *person-month* yang kemudian dibagi dengan durasi pengerjaan proyek untuk mendapatkan nilai rata-rata pekerja yang dibutuhkan atau *average staffing*. Perhitungan durasi proyek dapat diketahui melalui persamaan 2.6 berikut:

$$\text{TDEV} = [3,67 \times (92,61)^{(0,28+0,2 \times (1,33-0,91))}] = 9,14 \text{ atau } 9 \text{ bulan.}$$

Menggunakan rumus persamaan 2.6 maka didapatkan nilai 3,67 dari ketentuan COCOMO II, nilai 92,61 didapatkan dari nilai *person-month*, nilai E didapatkan dari nilai *scale factor*. Kemudian setelah dihitung didapatkan nilai 9,14 atau setara dengan 9 bulan.

Kemudian untuk mendapatkan nilai *average staffing* menggunakan persamaan 2.3 berikut:

$$P = 92,61 \times 9 = 10,29 \text{ atau } 10 \text{ orang}$$

Nilai dari PM didapatkan dari perhitungan estimasi usaha (*Person-Month*) dengan nilai 92,61 dan durasi didapatkan dari nilai estimasi waktu dengan hasil 9 bulan waktu pengerjaan, sehingga nilai *average staffing*nya adalah 10 orang karena dibulatkan kebawah. Dengan ini pengerjaan sistem SIRIS terdiri dari 10 orang. Berdasarkan acuan dari buku *Information Technology Project Management* terkait struktur organisasi fungsional (Marchewka, 2003), jadi total estimasi perhitungan biaya dalam pembuatan sistem, digunakan persamaan 2.7 berikut:

$$\text{Biaya perbulan} = 10 \text{ orang} \times \text{Rp. } 2.200.000 = \text{Rp. } 22.000.000\text{-, (Perbulan)}$$

$$\text{Biaya total} = \text{Rp. } 22.000.000 \times 9 \text{ bulan} = \text{Rp. } 198.000.000\text{-,}$$

Nilai perhitungan biaya perbulan maka didapatkan 10 orang pekerja dikalikan dengan gaji/bulan, maka didapatkan nilai Rp. 22.000.000 untuk gaji seluruh karyawan perbulannya. Kemudian dihitung biaya total dengan cara gaji/orang dalam 1 bulan dikalikan dengan 9 bulan waktu pengerjaan, maka didapatkan nilai Rp. 198.000.000,-

Nilai Average cost labor didapatkan dari standar gaji UMR kabupaten malang. Setelah mengetahui perhitungan rata-rata pekerja, durasi pengerjaan dan gaji pekerja dari perhitungan metode COCOMO II, setelah itu membuat penjadwalan pengerjaan menggunakan *gantchart* pada dokumen WBS yang terlampir pada (lampiran E).

5.6 Analisis Perhitungan dengan *Guesstimate*

Analisis perhitungan dengan menggunakan metode *guesstimate* ini menjelaskan tentang perhitungan harga suatu proyek, dengan jumlah pekerja beserta waktu pengerjaannya. Dalam penggunaan metode *guesstimate* ini harus memiliki *project manager* yang berpengalaman minimal 2 tahun kerja, untuk menganalisis dan menentukan durasi pengerjaan proyek, jumlah pekerja dan harga suatu proyek. Berdasarkan hasil wawancara (lampiran B) dan survey langsung dengan CEO 7Treesdigital, mendapatkan informasi berupa:

1) Durasi pengerjaan proyek sekitar 3 bulan dimulai pada bulan oktober 2014 hingga januari 2015.

2) Anggota tim pengembang *website* sistem SIRIS berjumlah 5 orang sebagai berikut:

Project Manager : Mochammad Irsyad

Programmer dan Desain UI : Agung Surya

Programmer dan DB Desainer : Fauzan Tolabi

Analisis dan Dokumentasi : Nadya Rizky Arimurti

Analisis dan Desain UI : Lia Hasanah

Dengan tugas untuk masing-masing anggota tim proyek sebagai berikut:

a) *Project Manager*

Project manager adalah seorang kapten dalam pengembangan suatu proyek yang bertugas memimpin dan menggerakkan suatu proyek. Tanggung jawab *project manager* adalah untuk memenuhi kebutuhan yang berupa kebutuhan tugas, kebutuhan tim, maupun kebutuhan secara individual. *Project manager* menjadi penghubung antara strategi dan tim.

b) *Programmer*

Programmer atau Pengembang dibagi tugas menjadi *Back-End* dan *Front-End*. Pengembang *Back-End* ini mempunyai tugas untuk membuat halaman *administrator*, membuat halaman *login*, halaman dokter, serta halaman prediksi demam.

Pengembang *Front-End* ini bertugas untuk membuat *front-end* pada halaman home, halaman dokter, halaman pasien, halaman penyakit, halaman artikel, halaman prediksi serta halaman *login*.

c) *Desainer UI*

Desainer bertugas untuk membuat desain, konsep, dan tata letak (*mockup*) berdasarkan pengetahuan tentang prinsip-prinsip tata letak.

d) *DB Desainer*

Database Desainer bertugas untuk menganalisis data apasaja yang perlu disimpan kedalam database serta merancang database beserta *attribute* yang dibutuhkan dalam pembangunan sistem.

e) *Analisis*

Analisis bertugas pada awal proyek, untuk menganalisis ruang lingkup sistem dengan tujuan untuk mendapatkan fokus awal proyek dengan mendefinisikan visi proyek agar menjadi sesuatu yang realistis. Tugasnya analisis meliputi pembuatan Usecase, DFD, ERD dsb.

f) Dokumentasi

Dokumentasi bertugas untuk merangkum dan mendokumentasikan pengerjaan dari awal hingga akhir proyek kedalam *hardcopy* dan mencatat segala perubahan yang terjadi pada saat pengembangan sistem informasi ini. Tugas dokumentasi meliputi pembuatan dokumen perancangan, dokumen kontrak perjanjian dan pembuatan gantt chart.

3) Pada 7Treesdigital mematok gaji perjabatan dalam jangka waktu per-bulan, yaitu dengan harga UMR kabupaten malang Rp. 2.200.000,- kecuali *Project Manager* akan mendapatkan gaji Rp. 2.500.000,-. Dengan menggunakan persamaan 2.7 total estimasi biaya pembuatan website sistem informasi rekam medis adalah sebagai berikut:

4 orang x Rp. 2.200.000 = Rp. 8.800.000,- (Biaya perbulan)

1 orang x Rp. 2.500.000 = Rp. 2.500.000,- (Biaya perbulan)

Total biaya perbulan adalah: Rp. 11.300.000,-

Nilai gaji *project manager* pada 7trees berbeda dengan gaji tim pengembang lainnya, maka didapatkan biaya perbulannya adalah **Rp. 11.300.000,-**

Kemudian dikalikan dengan durasi pengerjaan proyek sistem informasi rekam medis, yaitu 3 bulan maka hasilnya adalah:

Total perbulan Rp. 11.300.000 x 3 bulan = Rp. 33.900.000,-

Dengan harga gaji sebesar Rp. 33.900.000 / 3 bulan, maka sistem ini dijual seharga Rp. 68.000.000,-. pihak 7trees menjual 2x lipat dari harga gaji yang dibayarkan kepada pekerjanya.

5.7 Analisis Perhitungan dengan COCOMO II

1. Perhitungan Estimasi Usaha (*Person-Month*)

$$PM = 2,94 \times (6,72)^{1,33} \times 2,5 = 92,61$$

Hasil dari Person-Month terdiri dari 152 jam waktu kerja per PM sehingga dalam proyek ini memiliki waktu kerja sebanyak 14076 jam.

2. Durasi Proyek

$$TDEV = [3,67 \times (92,61)^{(0,28+0,2 \times (1,33-0,91))}] = 9,14 \text{ atau } 9 \text{ bulan.}$$

3. Average Staffing

$$\text{Average Staffing} = PM / \text{Duration}$$

Nilai dari PM didapatkan dari perhitungan estimasi usaha (*Person-Month*) dengan nilai 92,61 dan duration didapatkan dari nilai estimasi waktu dengan hasil 9 bulan waktu pengerjaan, sehingga nilai *average staffing*nya adalah 10 orang. Dengan ini pekerja yang dibutuhkan dalam pembuatan sistem informasi rekam medis terdiri dari 10 orang.

4. Total Estimasi Biaya Pembuatan SIRIS

$$\text{Biaya total} = \text{Average Staffing} \times \text{Average Labor Cost}$$

Biaya gaji yang digunakan berdasarkan nilai gaji UMR kabupaten malang, maka dapat dihitung total biaya perbulannya adalah Rp. 22.000.000,-. Kemudian dikalikan dengan durasi waktu pengerjaan proyek sistem SIRIS yaitu 9 bulan maka didapatkan hasil perhitungan berikut:

$$\text{Rp. } 22.000.000 \times 9 \text{ bulan} = \text{Rp. } 198.000.000,-$$

5.8 Analisis Hasil COCOMO II dan *Guesstimate*

Berikut adalah hasil analisis dari penelitian ini berupa perbandingan antar metode COCOMO II dan *guesstimate*, yang berisikan data berupa durasi pengerjaan, sumber daya manusia, dan biaya total dari pengerjaan keseluruhan yang akan disajikan pada tabel 5.6 berikut:

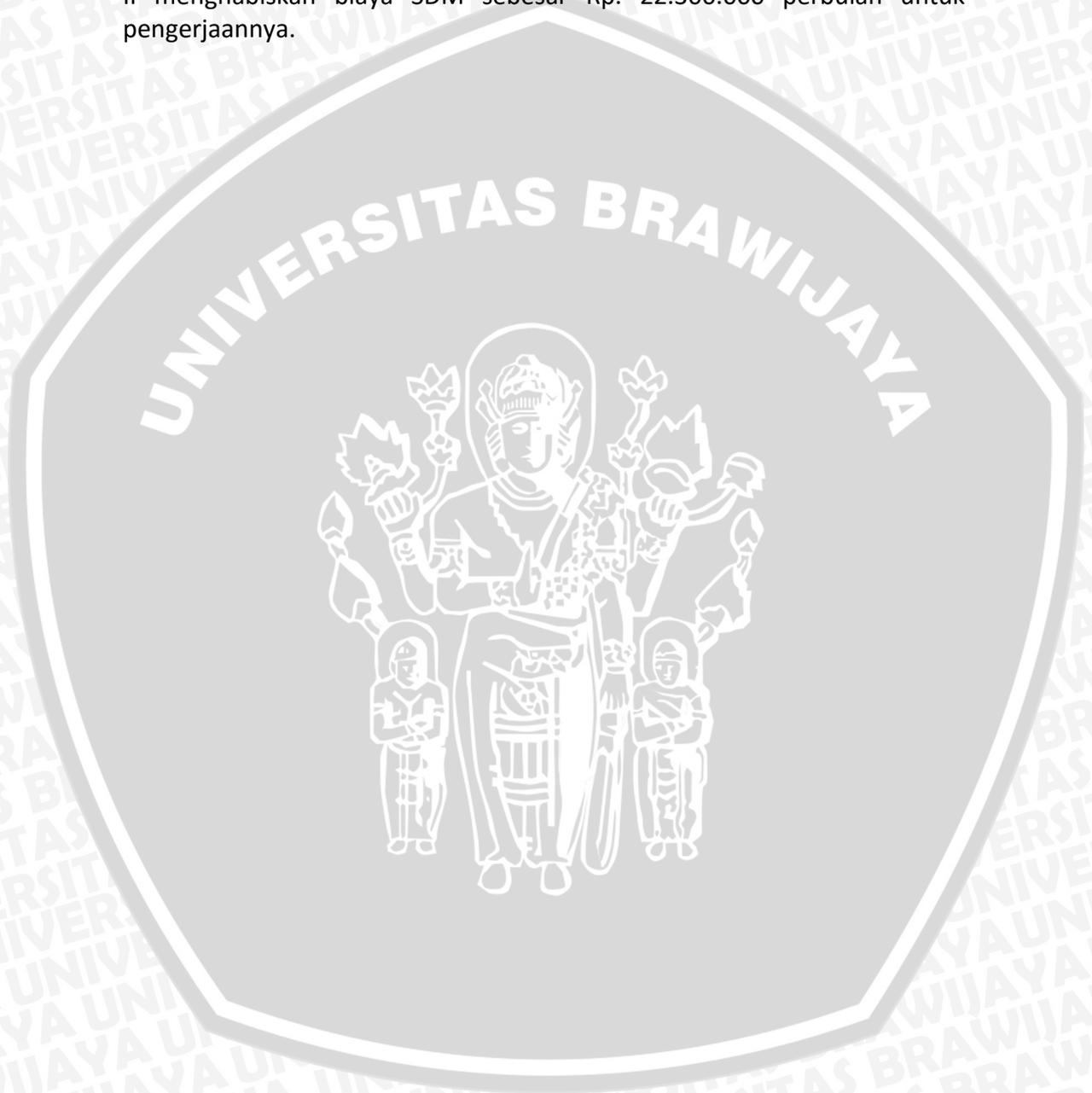
Tabel 5.9 Analisis Hasil COCOMO II dan *Guesstimate*

Penjelasan	<i>Guesstimate</i>	COCOMO II
Durasi Pengerjaan	3 bulan	9 bulan
Sumber Daya Manusia	5 orang	10 orang
Biaya Sumber Daya Manusia (Perbulan)	Rp. 11.300.000 ,-	Rp. 22.000.000 ,-
Biaya total (Biaya SDM dikali Durasi)	Rp. 33.900.000 ,-	Rp. 198.000.000 ,-

Kesimpulan pada tabel 5.9 diatas menjelaskan perbandingan antara metode COCOMO II dan *Guesstimate*. Dalam tabel tersebut terdapat pembengkakan durasi, sumber daya manusia, dan total biaya pada metode COCOMO II dibandingkan dengan metode *guesstimate*. Faktor-faktor penyebabnya adalah:

1. Pada durasi pengerjaannya metode COCOMO II menggunakan durasi waktu 9 bulan, adapun dalam kenyataannya hanya menggunakan waktu 3 bulan pada proyek SIRIS. Dalam hal ini kurang sesuai dengan durasi dari perhitungan COCOMO II, karena COCOMO II lebih mementingkan pembagian pekerjaannya secara merata setiap tahapan-tahapan sedangkan metode *guesstimate* lebih menekankan waktu pengerjaan secepat mungkin.
2. Pada metode *guesstimate* menggunakan 5 orang SDM untuk pengerjaannya sedangkan COCOMO II menggunakan 10 orang. Dari segi biaya memang metode *guesstimate* lebih sedikit dalam menggunakan SDM, akan tetapi pada kenyataannya dalam pengerjaannya banyak yang bekerja secara ganda. Sedangkan metode COCOMO II lebih mementingkan pembagian pekerjaan yang jelas dengan memberikan estimasi 10 orang dalam pengerjaannya, agar tidak ada yang bekerja secara ganda.
3. Perbedaan pada kedua metode terlihat dari segi SDM, metode *guesstimate* menggunakan lebih sedikit SDM dengan memaksakan durasi yang lebih cepat ketimbang metode COCOMO II. Dengan cara ini metode *guesstimate* hanya mengeluarkan biaya SDM sebesar Rp. 11.300.000 perbulan untuk

membayar gaji pekerjaanya, sedangkan metode COCOMO II lebih terstruktur dengan baik dari hal SDM. Dalam hal SDM metode COCOMO II menggunakan banyak orang dalam pengerjaan proyeknya untuk menghindari pekerjaan secara ganda, dan durasi pengerjaannya selama 9 bulan karena pada metode COCOMO II lebih tertata secara tahapan-tahapan pengerjaan untuk mencegah kegagalan disetiap tahapannya. Dari segi biaya metode COCOMO II menghabiskan biaya SDM sebesar Rp. 22.300.000 perbulan untuk pengerjaannya.



BAB 6 Penutup

6.1 Kesimpulan

Berdasarkan hasil penelitian ini yang dilakukan dalam manajemen proyek menggunakan sistem informasi rekam medis pada rumah sakit pendidikan pada fakultas kedokteran universitas padjajaran bandung, maka kesimpulan yang dapat diambil adalah sebagai berikut:

1. Dalam mengimplementasikan perhitungan metode COCOMO II dengan submodel *post architecture* untuk estimasi biaya *direct cost*, tahapan-tahapannya sebagai berikut:
 - a. Tahap pertama, pengumpulan data terkait perancangan sistem. Data yang dibutuhkan berupa dokumen ERD, DFD serta penyebaran kuesioner.
 - b. Tahap kedua, pada tahap ini menghitung *unadjusted function point* dengan cara mengidentifikasi tipe fungsi pengguna (EI, EO, EIF, ILF, EQ) serta menentukan bobot kompleksitas pengguna yang dianalisis dari dokumen DFD. Kemudian menentukan DET, RET, dan FTR serta menentukan bobot kompleksitas dari dokumen ERD.
 - c. Tahap ketiga, pada tahap ini dilakukannya penyebaran kuesioner dari *scale factor* dan *effort adjustment multiplier*.
 - d. Tahap keempat, pada tahap ini menentukan nilai bahasa pemrograman yang digunakan dari tabel nilai (*Quantitative Software Management*) QSM.
 - e. Tahap kelima, pada tahap ini menghitung estimasi usaha dengan menggunakan rumus estimasi usaha. Nilai-nilai yang dibutuhkan dalam perhitungan estimasi usaha tersebut didapat dari nilai kuesioner *scale factor*, nilai *function point* yang sudah dikonversi kedalam satuan *kilo line of code* (KLOC), dan nilai dari *effort adjustment multiplier*.
 - f. Tahap keenam, pada tahap terakhir ini menghitung estimasi biaya dengan cara menggunakan rumus estimasi biaya. Nilai yang dibutuhkan dalam perhitungan yaitu nilai *person-month*/estimasi usaha dan nilai *scale factor*.

Dari tahapan-tahapan implementasi perhitungan COCOMO II tersebut didapatkan hasil perhitungan estimasi biaya *direct cost*. Nilai estimasi biaya *direct cost* sudah terhitung pada tahap enam yaitu perhitungan estimasi biaya yang menentukan nilai gaji yang diterima oleh pekerja.

2. Penggunaan dokumen WBS dengan *gantchart* dalam proyek SIRIS, berguna untuk mendetailkan penjadwalan semua pengerjaan. Sedangkan metode COCOMO II digunakan untuk menghitung estimasi SDM, durasi dan biaya suatu proyek. Relasi antara dokumen WBS dan COCOMO II adalah, dokumen WBS berperan penting dari segi penjadwalan dalam pengerjaan proyek untuk menghasilkan pengerjaan sesuai durasi proyek secara tepat waktu sedangkan COCOMO II berguna untuk memberikan estimasi sumber daya manusia yang dibutuhkan, durasi proyek dan biaya suatu proyek.

3. Hasil dari perbandingan antara metode COCOMO II dan metode *guesstimate* adalah:

- a. Pada metode COCOMO II menghasilkan estimasi usaha (*person-month*) sebesar 92,61 dengan durasi pengerjaan selama 9 bulan dan rata-rata pekerjanya sebanyak 10 orang. Sedangkan metode *guesstimate* pada estimasi usaha memerlukan durasi sebanyak 5 orang dengan durasi pengerjaan selama 3 bulan.
- b. Pada metode COCOMO II menghasilkan estimasi biaya pengerjaan sebesar Rp. 22.000.000 /bulan dan menghabiskan biaya total sebesar Rp. 198.000.000 selama 9 bulan, sedangkan metode *guesstimate* menghasilkan estimasi biaya sebesar Rp. 11.300.000 /bulan dan menghabiskan biaya total sebesar Rp. 11.900.000 selama 3 bulan.

Perbedaan tersebut menunjukkan metode COCOMO II lebih banyak menggunakan SDM dan waktu pengerjaan proyeknya cenderung lebih lama kedua hal tersebut yang mendasari pembengkakan biaya, karena metode COCOMO II menghitung estimasi SDM, waktu proyek dan biaya secara terstruktur. COCOMO II mementingkan setiap SDM dapat bekerja sesuai porsinya masing-masing dan tidak ada SDM yang bekerja ganda, seperti yang dilakukan pada metode *guesstimate* yang mementingkan harga proyek tinggi dengan SDM yang sedikit agar mendapatkan untung yang lebih tinggi.

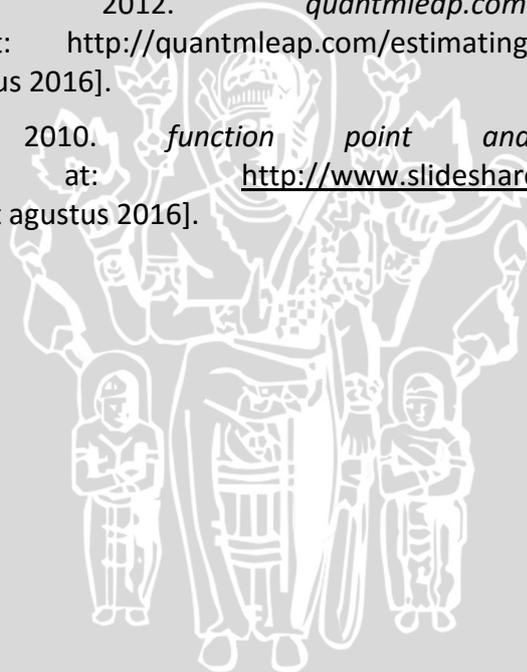
6.2 Saran

1. Untuk pengembangan perhitungan *function point* selanjutnya, diharapkan menggunakan submodel selain *post architecture* agar dapat mendalami metode COCOMO II.
2. Untuk pengembangan perhitungan *function point* selanjutnya selanjutnya, apabila menggunakan submodel *post architecture* dalam melakukan perhitungan agar tidak hanya menggunakan *unadjusted function point* melainkan bisa menggunakan *object point* ataupun perhitungan jumlah dari *Line Of Code*.

DAFTAR PUSTAKA

- Boehm, B., 1981. *software engineering economics*. California: University of Southern. [Online] Tersedia di: <https://www.amazon.com/Software-Engineering-Economics-Barry-Boehm/> [Diakses 17 Maret 2016].
- Boehm, B., 1997. *COCOMO II Model Definition Manual* (Version 1.4 ed.). University of Southern California.[pdf]
- Boehm, B., 2000. *Software Cost Estimation with COCOMO II*. California: University of Southern. [Online] Tersedia di: <https://www.amazon.com/Software-Cost-Estimation-Cocomo-II/> [Diakses 17 Maret 2016].
- Martini, G., 2012. perbandingan model algoritmik dan non algoritmik untuk estimasi biaya perangkat lunak.[pdf]
- Momoh et al., 2008. A Work Breakdown Structure for Implementing and Costing an ERP Project. VI, 94-103.[pdf]
- Noerlina., 2008. Perencanaan Manajemen Proyek Sistem Informasi Dan Teknologi Informasi Online Bisnis. 11, 440-450.[pdf]
- Lestari, O. D., 2015. Analisis Alokasi Sumber Daya Manusia Dalam Manajemen Proyek Sistem Informasi Website TRAVELCAR.CO.ID pada PT. JAGAT INDONESIA.
- Pranatha et al., 2012. Analisis Perkiraan Biaya Pembuatan Enterprise Resource Planning Modul Pabrik Gula Di Perusahaan Perkebunan Dengan Metode Cocomo II, 1-6.[pdf]
- Primaraka, A., 2008. Estimasi biaya pembuatan perangkat lunak menggunakan metode COCOMO II pada sistem informasi pelaporan kegiatan pembangunan, 1-7.[pdf]
- Rev. 2003. *Work Breakdown Structure* (Office of Engineering and Construction Management ed).[pdf] Amerika.
- Ridge, C., 2016. *Quantitative Software Management*. Diambil kembali dari Quantitative Software Management [online]. Tersedia di: <<http://www.qsm.com/resources/function-point-languages-table>> [Diakses 15 Maret 2016]
- Roy, C. R., 2001. Expert judgement in cost estimating: Modelling the reasoning process, 4-5. [pdf]
- Santosa, B., 2003. *Manajemen Proyek*. [pdf] Surabaya: Guna Widya.
- Schwalbe, K., 2014. *Information Technology Project Management* (7th edition ed.).[pdf] Course Technology.
- Siew et al., 2011. Improving Estimation Accuracy of the COCOMO II Using an Adaptive Fuzzy Logic Model, 27-30. [pdf]

- Soeharto, I., 2001. Manajemen Proyek dari Konseptual Sampai Operasional (Jilid 2 ed.)[pdf]. Jakarta: Erlangga.
- 7Treesdigital, 2014. Dokumen Perancangan Sistem. [pdf] PT. Dikasi Satu Digital.
- 7Treesdigital, 2014. Company Profile. [pdf] PT. Dikasi Satu Digital.
- Marchewka, J. R., 2003. Information Technology Project Management (1st edition).[pdf]
- Pressman, Roger S., 2012. Software engineering: a practitioner's approach-5th ed. McGraw: Hill series in computer scine.[pdf]
- Brady et al., 2010. Exploring the use of entity-relationship diagramming as a technique to support grounded theory inquiry [pdf]. Bradford: Emerald Group Publishing.
- Sarno et al., 2002. Pengembangan Metode Analogy Untuk Estimasi Biaya Rancang Bangun Perangkat Lunak.[pdf] Volume Vol. 6 No.2.
- Marom, S., 2012. *quantmleap.com*. [Online] Available at: <http://quantmleap.com/estimating-vs-guesstimating/> [Diakses Agustus 2016].
- Litvinenko, A., 2010. *function point analysis*. [Online] Available at: <http://www.slideshare.net/Programeter> [Diakses jum'at agustus 2016].



LAMPIRAN A

PENJELASAN TENTANG MANAJEMEN PROYEK

Laporan Hasil Wawancara

I. Latar Belakang

Puji syukur atas kehadiran Tuhan Yang Maha Esa, karena berkat rahmat dan hidayah-Nya sehingga kami dapat melaksanakan kegiatan ini dengan lancar dan sebagai mana mestinya.

Kegiatan wawancara ini merupakan salah satu pencarian informasi terkait yang diperlukan oleh pewawancara untuk kebutuhan tugas akhir (Skripsi) yang bertujuan untuk memperoleh informasi dari narasumber. Dalam hal ini topiknya adalah "Manajemen Proyek pada 7Tressdigital" oleh karena itu kami mewawancarai salah satu software house yang ada dikota malang.

II. Maksud Dan Tujuan

1. Memenuhi informasi terkait manajemen proyek yang ada disoftware house
2. Memahami lebih dalam terkait manajemen proyek

III. Topik Wawancara

Penjelasan tentang manajemen proyek pada 7Tressdigital

IV. Waktu dan Tempat Kegiatan

Wawancara ini dilaksanakan pada:

Hari / Tanggal : Sabtu, 16 April 2016.
Pukul : 09.00 WIB s/d selesai.
Tempat : Kantor 7Tressdigital

V. Laporan Hasil Wawancara

Narasumber : Yugo Yudansha, S.kom, CIMP
Pewawancara : Aldi Yudha Maswinandar

VI. Hasil Wawancara

Pewawancara:

1. Apakah di 7Treesdigital sudah menggunakan dokumen-dokumen yang lengkap seperti dokumen perancangan?
2. Jika di 7Tress sudah ada, dokumen-dokumen apa saja yang ada disetiap proyeknya?
3. Apakah di 7Treesdigital memiliki manajemen proyek sendiri ?

Narasumber:

1. Iya, ada.

2. Seperti diketahui dokumen yang digunakan meliputi dari dokumen penawaran, kontrak perjanjian kerja dan dokumen kebutuhan. Akan tetapi project kedepannya, akan dibuatkan dokumen untuk perancangan suatu proyek yang isinya lebih lengkap tentang proyek itu tersebut.
3. Iya, kita punya. Kita menggunakan Gantt Chart dalam setiap pengerjaan untuk mengontrol setiap pekerjaan dari tim. Saya menggunakan Ganttchart ini baru 2 tahun terakhir, karena saya berfikir ganttchart ini penting dalam pengerjaan suatu proyek apalagi didunia teknologi. Jadi, kebanyakan pada software house lain tidak menggunakan manajemen proyek dengan benar, dan hasilnya dapat dilihat susah merancang alokasi sumber daya manusia, kemudian banyak pekerjaan yang sangat membuang-buang waktu, pengerjaan proyek mundur pasti membuang cost dan pastinya klient tidak puas.



SURAT KETERANGAN

Yang bertanda tangan dibawah ini, saya :

Nama : Aldi Yudha Maswinandar

NIM : 125150407111043

Jurusan : Sistem Informasi

Telah melaksanakan wawancara tentang "Manajemen Proyek", dengan narasumber:

Nama : Yugo Yudansha, S.kom, CIMP

Jabatan : CEO 7Treesdigital

Demikian surat keterangan ini saya buat dengan sebenar-benarnya sebagai bukti telah melakukan wawancara.

Malang, 16 April 2016

Narasumber

Yugo Yudansha, S.kom, CIMP

Pewawancara

Aldi Yudha Maswinandar

LAMPIRAN B

PENGERJAAN SISTEM SIRIS

Laporan Hasil Wawancara

I. Latar Belakang

Puji syukur atas kehadiran Tuhan Yang Maha Esa, karena berkat rahmat dan hidayah-Nya sehingga kami dapat melaksanakan kegiatan ini dengan lancar dan sebagai mana mestinya.

Kegiatan wawancara ini merupakan salah satu pencarian informasi terkait yang diperlukan oleh pewawancara untuk kebutuhan tugas akhir (Skripsi) yang bertujuan untuk memperoleh informasi dari narasumber. Dalam hal ini topiknya adalah "Pengerjaan sistem SIRIS (Sistem Informasi Rekam Medis) pada 7Treesdigital" oleh karena itu kami mewawancarai salah satu software house yang ada di kota Malang.

II. Maksud Dan Tujuan

Menggal informasi terkait pengerjaan sistem SIRIS

III. Topik Wawancara

Pengerjaan sistem SIRIS yang ada pada 7Treesdigital

IV. Waktu dan Tempat Kegiatan

Wawancara ini dilaksanakan pada:

Hari / Tanggal : Rabu, 18 Mei 2016.
Pukul : 10.00 WIB s/d selesai.
Tempat : Kantor 7Treesdigital

V. Laporan Hasil Wawancara

Narasumber : Yugo Yudansha, S.kom, CIMP

Pewawancara : Aldi Yudha Maswinandar

VI. Hasil Wawancara

Pewawancara:

1. Berapa lama durasi pengerjaan sistem SIRIS ?
2. Berapa anggota tim pengembang sistem SIRIS ?
3. Berapa gaji anggota tim perbulannya ?
4. Berapa nilai proyek sistem SIRIS ?

Narasumber:

1. Untuk durasi pengerjaan sistem ini sekitar 3 bulan, sejak bulan oktober 2014 hingga januari 2015.
2. Anggota tim pengembang yang terlibat terdiri dari 5 orang, *project manager* saat itu mohammad irsyad, agung surya sebagai *programmer* dan desain *interface*, fauzan tolabi sebagai *programmer dan database designer*, nadya rizky sebagai analisis dan dokumentasi, dan lia hasanah sebagai analisis dan desain *interface*.

3. Gaji yang diterima untuk tim pengembang sebesar UMR kabupaten malang dengan nilai Rp. 2.500.000 ,- dan untuk *project manager* karena sebagai penanggung jawab pada proyek PM mendapatkan gaji sebesar Rp. 2.500.000,-.
4. Jadi, harga proyek SIRIS ini sebesar Rp. 68.000.000,-.

SURAT KETERANGAN

Yang bertanda tangan dibawah ini, saya :

Nama : Aldi Yudha Maswinandar

NIM : 125150407111043

Jurusan : Sistem Informasi

Telah melaksanakan wawancara tentang "Pengerjaan sistem SIRIS (Sistem Informasi Rekam Medis)", dengan narasumber:

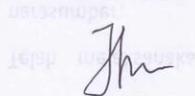
Nama : Yugo Yudansha, S.kom, CIMP

Jabatan : CEO 7Treesdigital

Demikian surat keterangan ini saya buat dengan sebenar-benarnya sebagai bukti telah melakukan wawancara.

Malang, 18 Mei 2016

Narasumber



Yugo Yudansha, S.kom, CIMP

Pewawancara



Aldi Yudha Maswinandar

LAMPIRAN C
KUESIONER PENILAIAN SCALE FACTOR dan EFFORT MULTIPLIER
SISTEM INFORMASI REKAM MEDIS
(Studi kasus : 7Tressdigital)

Nama : Fauzan Tolab
 Jabatan : Programmer
 Telp : -
 Jenis Kelamin : Laki - laki
 Pengalaman bekerjasama : 1x > 1x > 2x
 dengan tim proyek
 Tgl pengisian kuesioner : 17 Mei 2016

PETUNJUK UMUM

Survei dengan cara penyebaran kuesioner ini adalah salah satu cara yang digunakan untuk mengetahui beberapa faktor yang berhubungan pada estimasi biaya pada sistem informasi rekam medis, sistem ini adalah sistem yang dikembangkan oleh 7Tressdigital. Hasil dari penyebaran kuesioner ini nantinya akan digunakan penulis sebagai bahan pengerjaan tugas akhir (skripsi). Mohon kerjasama Bapak/Ibu/Saudara dalam memberikan penilaian yang jujur dan apa adanya secara real, terima kasih atas perhatiannya yang telah membantu studi penelitian ini.

Petunjuk Pengisian Kuesioner :

1. Sebelum mengisikan penilaian Bapak/Ibu/Saudara dimohon untuk membaca gambaran umum perusahaan dan kondisi proyek sistem informasi rekam medis.
2. Dalam mengisikan nilai, Bapak/Ibu/Saudara dituntut mengisikan dengan pengetahuan atau pengalaman yang anda miliki terkait dengan pengerjaan sebuah proyek sistem informasi rekam medis.
3. Nilai yang diberikan adalah satu penilaian saja yang berada pada range *very low* sampai *very high* (dicentang pada kolom yang telah disediakan).

Gambaran Umum :

1. SIRIS (Sistem Informasi Rekam Medis) adalah sistem yang membantu dokter memberikan catatan kepada pasien yang ditanganinya.
2. Pada SIRIS (Sistem Informasi Rekam Medis) juga dapat memberikan rekam medis pasien setelah diperiksa oleh dokter.
3. 7treesdigital berperan dalam pembuatan sistem ini, yang nantinya akan diterapkan ke rumah sakit yang membutuhkan sistem SIRIS ini, fitur yang bisa dilakukan oleh sistem SIRIS ini berupa:

Kuesioner Scale Factor Tentang Sistem Informasi Rekam Medis (SIRIS)

Indikator VL (Very Low), L (Low), N (Normal), H (High), VH (Very High) hingga EH (Extra High)

Scala factor	Deskripsi	Penilaian					
		VL	L	N	H	VH	EH
<i>Precedentedness</i>	<p>Faktor skala ini merupakan faktor skala yang menggambarkan pengalaman masa lalu organisasi dengan proyek-proyek sejenis.</p> <p>Nilai berkisaran mulai : (VL) : Yang mengindikasi bahwa organisasi tidak memiliki pengalaman sebelumnya dengan proyek sejenis. (L) : Pernah terjadi sebelumnya (N) : Umumnya familiar (H) : Hingga ekstra tinggi (VH) : Menunjukkan bahwa organisasi memiliki pemahaman mendalam terhadap proyek tersebut.</p>			✓			
<i>Fleksibility</i>	<p>Faktor skala ini merupakan faktor skala yang menggambarkan kemampuan klien dalam menentukan tujuan dan mengkomunikasikan kebutuhan perangkat lunak kepada tim pengembang.</p> <p>Nilai berkisaran mulai : (VL) : Jika dalam proses pengembangan sistem benar-benar dikendalikan oleh klien dan definisi kebutuhannya sangat jelas. (L) : Klien memiliki definisi kebutuhan yang jelas, hanya saja ada beberapa kebutuhan yang tidak terdefinisi dengan benar. (N) : Klien memiliki tujuan dalam mengembangkan sistem akan tetapi tidak bisa mendefinisikan kebutuhan (H) : Klien hanya memiliki referensi proyek lain (VH) : Klien hanya menyerahkan pengerjaan sepenuhnya ke organisasi dalam hal pengerjaan, dan klien hanya melakukan sedikit intervensi</p>				✓		
<i>Risk Resolution</i>	<p>Faktor skala ini merupakan faktor skala yang menggambarkan proses identifikasi risiko yang terkait dengan proyek, untuk menghasilkan pengukuran jumlah risiko yang terkait dengan proyek.</p> <p>Nilai berkisaran mulai : (VL) : Sedikitnya 20% (L) : Beberapa 40% (N) : Sering 60% (H) : Umumnya 75%</p>		✓				



	(VH) : Kebanyakan 90% (EH) : Keseluruhan 100%							
<i>Team Cohesion</i>	<p>Faktor skala ini merupakan faktor skala yang menggambarkan seberapa baik tim pengembangan bekerja sama.</p> <p>Nilai berkisaran mulai :</p> <p>(VL) : Nnilai ini dapat diberikan apabila tim pengembang tidak tahu menahu tentang <i>jobdesk</i> yang diberikan</p> <p>(L) : Nilai ini dapat diberikan apabila tim pengembang jarang berinteraksi satu sama yang lain.</p> <p>(N) : Nilai ini dapat diberikan apabila tim pengembang jarang melakukan diskusi bersama, akan tetapi pernah diskusi via media.</p> <p>(H) : Nilai ini dapat diberikan apabila tim pengembang hanya melakukan diskusi bersama sebanyak 4x pertemuan sampai proyek selesai.</p> <p>(VH) : Nilai ini dapat diberikan apabila tim pengembang solid dalam pengerjaan dan sering melakukan diskusi bersama terkait proyek tersebut.</p>						✓	
<i>Proces Maturity</i>	<p>Faktor skala ini merupakan faktor skala yang menggambarkan kematangan proses pengembangan perangkat lunak dalam organisasi. Hal ini didasarkan pada Model Kematangan Kemampuan Rekayasa Perangkat Lunak Institute (CMM) yang didefinisikan berdasarkan lima tingkat kematangan untuk proses pengembangan perangkat lunak. Berikut adalah deskripsi dari tiap tingkatan kematangan kemampuan rekayasa perangkat lunak :</p> <p>i. Tingkat awal (<i>Initial Level</i>)</p> <p>Pada tingkat ini, organisasi tidak memiliki prosedur manajemen yang efektif atau rencana proyek. Prosedur formal untuk pengendalian proyek ada dan tidak ada mekanisme organisasi untuk memastikan bahwa prosedur tersebut digunakan secara konsisten. Di level ini, organisasi berhasil mengembangkan perangkat lunak namun karakteristik perangkat lunak (kualitas dll) dan proses (anggaran, jadwal dll) akan tak terduga.</p> <p>ii. <i>Repeatable Level</i></p> <p>Pada tingkat ini, proses-proses dasar manajemen proyek sudah dikembangkan untuk melacak biaya, jadwal, dan fungsionalitas perangkat lunak. Disebut tingkat berulang karena organisasi sudah mendokumentasikan proses-proses yang perlu agar dapat berhasil mengulangi proyek dari jenis yang sama.</p>			✓				



	<p>iii. <i>Defined Level</i> Pada tingkat ini, organisasi memiliki proses dan kegiatan rekayasa perangkat lunak (definisi kebutuhan, development, maintenance) sudah didokumentasikan dan diintegrasikan dalam proses pengembangan perangkat lunak yang standar untuk digunakan dalam organisasi. Keseluruhan proyek dikembangkan dan dipelihara menggunakan proses yang standar dan resmi diakui dalam organisasi.</p> <p>iv. <i>Managed Level</i> Sebuah organisasi pada level 4 memiliki ukuran yang rinci tentang proses pengembangan perangkat lunak dan kualitas produk yang telah dikumpulkan dan didokumentasikan, baik proses/kualitas produk secara kuantitatif yang dapat dimengerti dan dikontrol. Proses dan pengukuran produk dikumpulkan dan dimasukkan ke dalam aktivitas perbaikan proses.</p> <p>v. <i>Optimising Level</i> Pada tingkat ini, sebuah organisasi berkomitmen untuk perbaikan proses yang berkesinambungan. Proses perbaikan yang dianggarkan dan direncanakan dan merupakan bagian integral dari proses organisasi.</p> <p>Dalam penilaian <i>process maturity</i>, nilai diberikan berdasarkan tingkat kematangan perangkat lunak yang telah dijelaskan dibagian atas. Contoh penilaian : VR = tingkat awal, L = repeatable level, N = defined level, H = managed level, dan VH = optimising level.</p>						
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--	--	--	--

Kuesioner Effort Multiplier Tentang Sistem Informasi Rekam Medis (SIRIS)

Indikator VL (Very Low), L (Low), N (Nominal), H (High), VH (Very High) hingga EH (Extra High)

Scala factor	Deskripsi	Penilaian					
		VL	L	N	H	VH	EH
RELY	<p>RELY (Keandalan perangkat lunak yg disyaratkan): Penilaian <i>cost driver</i> terkait sejauh mana perangkat lunak menjalankan aplikasi sesuai fungsinya selama periode waktu.</p> <p>Very Low : Jika kegagalan hanya sekedar berdampak atas ketidak nyamanan pengguna Low : Jika kegagalan terjadi pada pengguna, akan tetapi mudah diperbaiki pada tingkat rendah Nominal : Jika kegagalan terjadi pada pengguna, akan tetapi mudah diperbaiki pada tingkat tinggi High : Jika kegagalan merugikan dalam segi waktu dan tenaga kerja Very High : Jika kegagalan merugikan dalam segi dana Extra High : Jika kegagalan beresiko bagi kehidupan manusia.</p>		✓				
DATA	<p>DATA (Ukuran Database): Penilaian <i>cost driver</i> terkait ukuran database yang digunakan. Ukuran dapat diperoleh dengan menghitung D/P.</p> <p>Low : DB bytes/Pgm SLOC < 10 Nominal : Jika 10 D/P < 100 High : Jika 100 D/P < 1000 Very High : Jika DB bytes/Pgm = 1000</p>			✓			
CPLX	<p>CPLX (Kompleksitas Produk): Penilaian <i>cost driver</i> terkait perangkat lunak dan perangkat keras dalam melakukan tugasnya, seperti <i>platform</i> (arsitektur, sistem operasi, bahasa pemrograman dan antarmuka yang terkait), sistem manajemen <i>database</i>, browser yang sesuai digunakan dalam menjalankan aplikasi ini.</p> <p>Very Low : Tidak menggunakan <i>platform</i> Low : Hanya menggunakan 1 (satu) dari sekian banyak <i>platform</i> Nominal : Hanya menggunakan 2 (dua) dari sekian banyak <i>platform</i> High : Menggunakan perangkat lunak Very High : Menggunakan perangkat keras Extra High : Menggunakan perangkat keras, perangkat lunak beserta <i>platform</i></p>				✓		
DOCU	<p>DOCU (Dokumentasi sesuai dengan kebutuhan siklus hidup perangkat lunak): Penilaian <i>cost driver</i> dalam hal kesesuaian dokumentasi proyek terhadap kebutuhan siklus hidup perangkat lunak.</p>						

	<p>Very Low : Hanya beberapa kebutuhan yang didokumentasikan Low : Hanya sebagian saja kebutuhan yang didokumentasikan Nominal : Dokumentasi sesuai dengan kebutuhan perangkat lunak High : Berlebihan dalam pendokumentasian kebutuhan perangkat lunak Very High : Sangat berlebihan dalam pendokumentasian kebutuhan perangkat lunak</p>				✓			
ACAP	<p>ACAP (Kemampuan Analisis Sistem): Penilaian <i>cost driver</i> terkait kemampuan personel dalam analisis dan desain, efisiensi dan ketelitian, serta kemampuan untuk berkomunikasi dan bekerja sama. Dalam hal ini, dapat dinilai dari sertifikasi yang sudah didapatkan personel atau pengalaman kerja tim dalam suatu proyek.</p> <p>Very Low : Dari 10 proyek terakhir, analisis diproyek ini setidaknya pernah berpartisipasi sebanyak 2 kali Low : Dari 10 proyek terakhir, analisis diproyek ini setidaknya pernah berpartisipasi sebanyak 4 kali Nominal : Dari 10 proyek terakhir, analisis diproyek ini setidaknya pernah berpartisipasi sebanyak 6 kali High : Dari 10 proyek terakhir, analisis diproyek ini setidaknya pernah berpartisipasi sebanyak 7 kali Very High : Dari 10 proyek terakhir, analisis diproyek ini setidaknya pernah berpartisipasi sebanyak 9 kali</p>					✓		
PCAP	<p>PCAP (Kemampuan Programmer): Penilaian <i>cost driver</i> terkait kemampuan programmer dalam efisiensi penulisan kode program, ketelitian, dan kemampuan untuk berkomunikasi dan bekerja sama sebagai sebuah tim. Dengan kata lain sudah berapa banyak proyek di mana <i>programmer</i> tersebut ikut terlibat didalamnya.</p> <p>Very Low : Dari 10 proyek terakhir, programmer diproyek ini setidaknya pernah berpartisipasi sebanyak 2 kali Low : Dari 10 proyek terakhir, programmer diproyek ini setidaknya pernah berpartisipasi sebanyak 4 kali Nominal : Dari 10 proyek terakhir, programmer diproyek ini setidaknya pernah berpartisipasi sebanyak 6 kali High : Dari 10 proyek terakhir, programmer diproyek ini setidaknya pernah berpartisipasi sebanyak 7 kali Very High : Dari 10 proyek terakhir, programmer diproyek ini setidaknya pernah berpartisipasi sebanyak 9 kali</p>					✓	✗	
PCON	<p>PCON (Kontinuitas Personil): Penilaian <i>cost driver</i> terkait tingkat pergantian personel tiap tahun pada proyek.</p> <p>Very Low : 48% per tahun Low : 24% per tahun Nominal : 12% per tahun High : 6% per tahun Very High : 3% per tahun</p>		✓					
RUSE	<p>RUSE (Pengembangan kode yang diarahkan untuk penggunaan kembali): Penilaian <i>cost driver</i> terkait tingkat upaya yang diperlukan untuk mengembangkan komponen yang dimaksudkan untuk digunakan kembali pada proyek-proyek yang sedang berjalan atau proyek di masa mendatang.</p> <p>Low : Tidak digunakan dalam proyek, program dan produk manapun Nominal : Tidak digunakan dalam proyek lain</p>							✓

	High : Tidak digunakan dalam program lain Very High : Tidak digunakan dalam produk lain Extra High : Akan digunakan dalam produk lain						
TIME	TIME (Kendala waktu eksekusi): Penilaian <i>cost driver</i> terkait persentase kendala waktu eksekusi yang diharapkan dapat digunakan pada sistem perangkat lunak. Nominal : 50% dari presentase waktu eksekusi yang tersedia High : 70% dari presentase waktu eksekusi yang tersedia Very High : 85% dari presentase waktu eksekusi yang tersedia Extra High : 95% dari presentase waktu eksekusi yang tersedia			✓			
STOR	STOR (Kendala Penyimpanan Utama): Penilaian <i>cost driver</i> terkait persentase tingkat kendala penyimpanan utama yang dikenakan pada sistem perangkat lunak. Nominal : 50% dari penyimpanan yang tersedia High : 70% dari penyimpanan yang tersedia Very High : 85% dari penyimpanan yang tersedia Extra High : 95% dari penyimpanan yang tersedia			✓			
PVOL	PVOL (Kompleksitas dari hardware dan software): Penilaian <i>cost driver</i> terkait perubahan yang terjadi pada <i>hardware</i> dan <i>software</i> (OS, DBMS) pada kurun waktu tertentu. Low : Jika terjadi perubahan kecil selama 12 bulan, perubahan besar selama 1 bulan Nominal : Jika terjadi perubahan kecil 6 bulan, perubahan besar selama 2 minggu High : Jika terjadi perubahan kecil 2 bulan, perubahan besar selama 1 minggu Very High : Jika terjadi perubahan kecil 2 minggu, perubahan besar selama 2 hari	✗	✓				
AEXP	AEXP (Pengalaman tim dalam mengembangkan aplikasi): Penilaian <i>cost driver</i> terkait tingkat pengalaman kerja tim proyek pada suatu proyek pengembangan aplikasi sistem perangkat lunak atau subsistem. Very Low : 2 bulan Low : 6 bulan Nominal : 1 tahun High : 3 tahun Very High : 6 tahun			✓			
PEXP	PEXP (Pengalaman tim dalam pengembangan terkait platform aplikasi): Penilaian <i>cost driver</i> terkait pemahaman tim dalam menggunakan platform, interface, database, jaringan & middleware.						

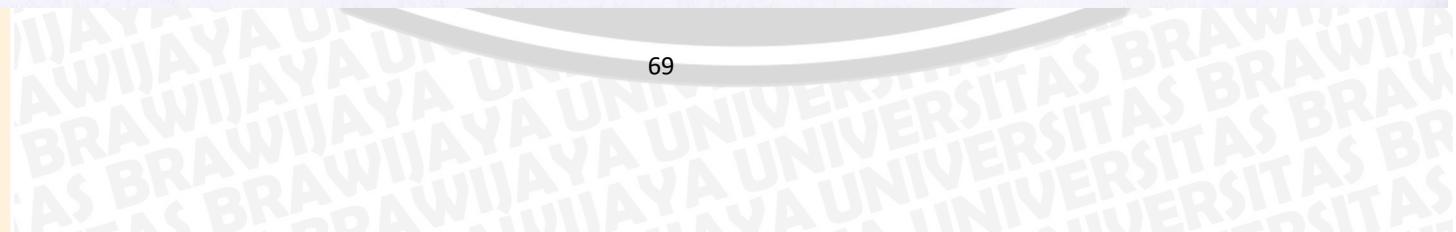
	<p>Very Low : 2 bulan Low : 6 bulan Nominal : 1 tahun High : 3 tahun Very High : 6 tahun</p>						✓	
LTEX	<p>LTEX: Penilaian <i>cost driver</i> terkait pengalaman dalam pemrograman dengan bahasa tertentu dan pemanfaatan <i>CASE tools</i> dari tim proyek dalam mngembangkan perangkat lunak.</p> <p>Very Low : 2 bulan Low : 6 bulan Nominal : 1 tahun High : 3 tahun Very High : 6 tahun</p>						✓	
TOOL	<p>TOOL (Penggunaan software bantuan): Penilaian <i>cost driver</i> terkait penggunaan <i>CASE tool</i> dalam pengembangan perangkat lunak pada proyek, seperti dari mengubah kode yang sederhana menjadi terintegrasi.</p> <p>Very Low : Edit, debug Low : Sederhana, backend case, integrasi dalam skala kecil Nominal : Basic lifecycle tool, cukup integrasi High : kuat, Mature lifecycle tool, cukup integrasi Very High : Kuat, matang, terintegrasi dengan baik</p>		✓					
SITE	<p>SITE: Penilaian <i>cost driver</i> terkait cara distribusi yang digunakan dalam pengembangan perangkat lunak pada proyek, apakah masih via telepon, fax, email, video conference, atau bahkan sudah menggunakan interactive multimedia.</p> <p>Very Low : Komunikasi telepon dan email Low : Komunikasi telepon pribadi dan FAX Nominal : Komunikasi melalui email pribadi High : Komunikasi melalui wideband electronic communications, seperti smartphone Very High : Komunikasi melalui wideband electronic communications, video conference Extra High : Komunikasi melalui interactive multimedia, seperti presentasi yang interactive, quiz interactive, website dsb.</p>		✓					
SCED	<p>SCED: Penilaian <i>cost driver</i> terkait tingkat persentase dari percepatan atau kemunduran jadwal terhadap jadwal suatu proyek yang telah ditetapkan sebelumnya.</p>			✓				



Very Low : 75% dari waktu normal Low : 85% dari waktu normal Nominal : 100% High : 130% dari waktu normal Very High : 160% dari waktu normal									
----------------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--	--	--	--	--	--	--

Handwritten signature

KANZAN TO LABI



LAMPIRAN C
KUESIONER PENILAIAN SCALE FACTOR dan EFFORT MULTIPLIER
SISTEM INFORMASI REKAM MEDIS
(Studi kasus: 7Treesdigital)

Nama : M. Irsyad

Jabatan : Programmer

Telp : -

Jenis Kelamin : Laki - Laki

Pengalaman bekerjasama : 1x > 1x > 2x
dengan tim proyek

Tgl pengisian kuesioner : 17 Mei 2016

PETUNJUK UMUM

Survei dengan cara penyebaran kuesioner ini adalah salah satu cara yang digunakan untuk mengetahui beberapa faktor yang berhubungan pada estimasi biaya pada sistem informasi rekam medis, sistem ini adalah sistem yang dikembangkan oleh 7Treesdigital. Hasil dari penyebaran kuesioner ini nantinya akan digunakan penulis sebagai bahan pengerjaan tugas akhir (skripsi). Mohon kerjasama Bapak/Ibu/Saudara dalam memberikan penilaian yang jujur dan apa adanya secara real, terima kasih atas perhatiannya yang telah membantu studi penelitian ini.

Petunjuk Pengisian Kuesioner :

1. Sebelum mengisikan penilaian Bapak/Ibu/Saudara dimohon untuk membaca gambaran umum perusahaan dan kondisi proyek sistem informasi rekam medis.
2. Dalam mengisikan nilai, Bapak/Ibu/Saudara dituntut mengisikan dengan pengetahuan atau pengalaman yang anda miliki terkait dengan pengerjaan sebuah proyek sistem informasi rekam medis.
3. Nilai yang diberikan adalah satu penilaian saja yang berada pada range *very low* sampai *very high* (dicentang pada kolom yang telah disediakan).

Gambaran Umum :

1. SIRIS (Sistem Informasi Rekam Medis) adalah sistem yang membantu dokter memberikan catatan kepada pasien yang ditanganinya.
2. Pada SIRIS (Sistem Informasi Rekam Medis) juga dapat memberikan rekam medis pasien setelah diperiksa oleh dokter.
3. 7treesdigital berperan dalam pembuatan sistem ini, yang nantinya akan diterapkan ke rumah sakit yang membutuhkan sistem SIRIS ini, fitur yang bisa dilakukan oleh sistem SIRIS ini berupa:

Gambaran Umum :

1. SIRIS (Sistem Informasi Rekam Medis) adalah sistem yang membantu dokter memberikan catatan kepada pasien yang ditanganinya.
2. Pada SIRIS (Sistem Informasi Rekam Medis) juga dapat memberikan rekam medis pasien setelah diperiksa oleh dokter.
3. 7treesdigital berperan dalam pembuatan sistem ini, yang nantinya akan diterapkan ke rumah sakit yang membutuhkan sistem SIRIS ini, fitur yang bisa dilakukan oleh sistem SIRIS ini berupa:

Kuesioner Scale Factor Tentang Sistem Informasi Rekam Medis (SIRIS)

Indikator VL (Very Low), L (Low), N (Normal), H (High), VH (Very High) hingga EH (Extra High)

Scala factor	Deskripsi	Penilaian					
		VL	L	N	H	VH	EH
<i>Precedentedness</i>	<p>Faktor skala ini merupakan faktor skala yang menggambarkan pengalaman masa lalu organisasi dengan proyek-proyek sejenis.</p> <p>Nilai berkisaran mulai :</p> <p>(VL) : Yang mengindikasi bahwa organisasi tidak memiliki pengalaman sebelumnya dengan proyek sejenis. (L) : Pernah terjadi sebelumnya (N) : Umumnya familiar (H) : Hingga ekstra tinggi (VH) : Menunjukkan bahwa organisasi memiliki pemahaman mendalam terhadap proyek tersebut.</p>				✓		
<i>Fleksibility</i>	<p>Faktor skala ini merupakan faktor skala yang menggambarkan kemampuan klien dalam menentukan tujuan dan mengkomunikasikan kebutuhan perangkat lunak kepada tim pengembang.</p> <p>Nilai berkisaran mulai :</p> <p>(VL) : Jika dalam proses pengembangan sistem benar-benar dikendalikan oleh klien dan definisi kebutuhannya sangat jelas. (L) : Klien memiliki definisi kebutuhan yang jelas, hanya saja ada beberapa kebutuhan yang tidak terdefinisi dengan benar. (N) : Klien memiliki tujuan dalam mengembangkan sistem akan tetapi tidak bisa mendefinisikan kebutuhan (H) : Klien hanya memiliki referensi proyek lain (VH) : Klien hanya menyerahkan pengerjaan sepenuhnya ke organisasi dalam hal pengerjaan, dan klien hanya melakukan sedikit intervensi</p>			✓			
<i>Risk Resolution</i>	<p>Faktor skala ini merupakan faktor skala yang menggambarkan proses identifikasi risiko yang terkait dengan proyek, untuk menghasilkan pengukuran jumlah risiko yang terkait dengan proyek.</p> <p>Nilai berkisaran mulai :</p> <p>(VL) : Sedikitnya 20% (L) : Beberapa 40% (N) : Sering 60% (H) : Umumnya 75%</p>			✓			

	(VH) : Kebanyakan 90% (EH) : Keseluruhan 100%							
<i>Team Cohesion</i>	<p>Faktor skala ini merupakan faktor skala yang menggambarkan seberapa baik tim pengembangan bekerja sama.</p> <p>Nilai berkisaran mulai :</p> <p>(VL) : Nilai ini dapat diberikan apabila tim pengembang tidak tahu menahu tentang <i>jobdesk</i> yang diberikan</p> <p>(L) : Nilai ini dapat diberikan apabila tim pengembang jarang berinteraksi satu sama yang lain.</p> <p>(N) : Nilai ini dapat diberikan apabila tim pengembang jarang melakukan diskusi bersama, akan tetapi pernah diskusi via media.</p> <p>(H) : Nilai ini dapat diberikan apabila tim pengembang hanya melakukan diskusi bersama sebanyak 4x pertemuan sampai proyek selesai.</p> <p>(VH) : Nilai ini dapat diberikan apabila tim pengembang solid dalam pengerjaan dan sering melakukan diskusi bersama terkait proyek tersebut.</p>						✓	
<i>Proses Maturity</i>	<p>Faktor skala ini merupakan faktor skala yang menggambarkan kematangan proses pengembangan perangkat lunak dalam organisasi. Hal ini didasarkan pada Model Kematangan Kemampuan Rekyasa Perangkat Lunak Institute (CMM) yang didefinisikan berdasarkan lima tingkat kematangan untuk proses pengembangan perangkat lunak. Berikut adalah deskripsi dari tiap tingkatan kematangan kemampuan rekyasa perangkat lunak :</p> <p>i. <i>Tingkat awal (Initial Level)</i></p> <p>Pada tingkat ini, organisasi tidak memiliki prosedur manajemen yang efektif atau rencana proyek. Prosedur formal untuk pengendalian proyek ada dan tidak ada mekanisme organisasi untuk memastikan bahwa prosedur tersebut digunakan secara konsisten. Di level ini, organisasi berhasil mengembangkan perangkat lunak namun karakteristik perangkat lunak (kualitas dll) dan proses (anggaran, jadwal dll) akan tak terduga.</p> <p>ii. <i>Repeatable Level</i></p> <p>Pada tingkat ini, proses-proses dasar manajemen proyek sudah dikembangkan untuk melacak biaya, jadwal, dan fungsionalitas perangkat lunak. Disebut tingkat berulang karena organisasi sudah mendokumentasikan proses-proses yang perlu agar dapat berhasil mengulangi proyek dari jenis yang sama.</p>							

	<p>iii. <i>Defined Level</i> Pada tingkat ini, organisasi memiliki proses dan kegiatan rekayasa perangkat lunak (definisi kebutuhan, development, maintenance) sudah didokumentasikan dan diintegrasikan dalam proses pengembangan perangkat lunak yang standar untuk digunakan dalam organisasi. Keseluruhan proyek dikembangkan dan dipelihara menggunakan proses yang standar dan resmi diakui dalam organisasi.</p> <p>iv. <i>Managed Level</i> Sebuah organisasi pada level 4 memiliki ukuran yang rinci tentang proses pengembangan perangkat lunak dan kualitas produk yang telah dikumpulkan dan didokumentasikan, baik proses/kualitas produk secara kuantitatif yang dapat dimengerti dan dikontrol. Proses dan pengukuran produk dikumpulkan dan dimasukkan ke dalam aktivitas perbaikan proses.</p> <p>v. <i>Optimising Level</i> Pada tingkat ini, sebuah organisasi berkomitmen untuk perbaikan proses yang berkesinambungan. Proses perbaikan yang dianggarkan dan direncanakan dan merupakan bagian integral dari proses organisasi.</p> <p>Dalam penilaian <i>process maturity</i>, nilai diberikan berdasarkan tingkat kematangan perangkat lunak yang telah dijelaskan dibagian atas. Contoh penilaian : VR = tingkat awal, L = repeatable level, N = defined level, H = managed level, dan VH = optimising level.</p>						
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--	--	--	--

Kuesioner Effort Multiplier Tentang Sistem Informasi Rekam Medis (SIRIS)

Indikator VL (Very Low), L (Low), N (Nominal), H (High), VH (Very High) hingga EH (Extra High)

Scala factor	Deskripsi	Penilaian					
		VL	L	N	H	VH	EH
RELY	<p>RELY (Keandalan perangkat lunak yg disyaratkan): Penilaian <i>cost driver</i> terkait sejauh mana perangkat lunak menjalankan aplikasi sesuai fungsinya selama periode waktu.</p> <p>Very Low : Jika kegagalan hanya sekedar berdampak atas ketidak nyamanan pengguna Low : Jika kegagalan terjadi pada pengguna, akan tetapi mudah diperbaiki pada tingkat rendah Nominal : Jika kegagalan terjadi pada pengguna, akan tetapi mudah diperbaiki pada tingkat tinggal High : Jika kegagalan merugikan dalam segi waktu dan tenaga kerja Very High : Jika kegagalan merugikan dalam segi dana Extra High : Jika kegagalan beresiko bagi kehidupan manusia.</p>					✓	
DATA	<p>DATA (Ukuran Database): Penilaian <i>cost driver</i> terkait ukuran database yang digunakan. Ukuran dapat diperoleh dengan menghitung D/P.</p> <p>Low : DB bytes/Pgm SLOC < 10 Nominal : Jika 10 D/P < 100 High : Jika 100 D/P < 1000 Very High : Jika DB bytes/Pgm = 1000</p>			✓			
CPLX	<p>CPLX (Kompleksitas Produk): Penilaian <i>cost driver</i> terkait perangkat lunak dan perangkat keras dalam melakukan tugasnya, seperti <i>platform</i> (arsitektur, sistem operasi, bahasa pemrograman dan antarmuka yang terkait), sistem manajemen <i>database</i>, browser yang sesuai digunakan dalam menjalankan aplikasi ini.</p> <p>Very Low : Tidak menggunakan <i>platform</i> Low : Hanya menggunakan 1 (satu) dari sekian banyak <i>platform</i> Nominal : Hanya menggunakan 2 (dua) dari sekian banyak <i>platform</i> High : Menggunakan perangkat lunak Very High : Menggunakan perangkat keras Extra High : Menggunakan perangkat keras, perangkat lunak beserta <i>platform</i></p>		✓				
DOCU	<p>DOCU (Dokumentasi sesuai dengan kebutuhan siklus hidup perangkat lunak): Penilaian <i>cost driver</i> dalam hal kesesuaian dokumentasi proyek terhadap kebutuhan siklus hidup perangkat lunak.</p>						

	<p>Very Low : Hanya beberapa kebutuhan yang didokumentasikan Low : Hanya sebagian saja kebutuhan yang didokumentasikan Nominal : Dokumentasi sesuai dengan kebutuhan perangkat lunak High : Berlebihan dalam pendokumentasian kebutuhan perangkat lunak Very High : Sangat berlebihan dalam pendokumentasian kebutuhan perangkat lunak</p>			✓			
ACAP	<p>ACAP (Kemampuan Analisis Sistem): Penilaian <i>cost driver</i> terkait kemampuan personel dalam analisis dan desain, efisiensi dan ketelitian, serta kemampuan untuk berkomunikasi dan bekerja sama. Dalam hal ini, dapat dinilai dari sertifikasi yang sudah didapatkan personel atau pengalaman kerja tim dalam suatu proyek.</p> <p>Very Low : Dari 10 proyek terakhir, analisis diproyek ini setidaknya pernah berpartisipasi sebanyak 2 kali Low : Dari 10 proyek terakhir, analisis diproyek ini setidaknya pernah berpartisipasi sebanyak 4 kali Nominal : Dari 10 proyek terakhir, analisis diproyek ini setidaknya pernah berpartisipasi sebanyak 6 kali High : Dari 10 proyek terakhir, analisis diproyek ini setidaknya pernah berpartisipasi sebanyak 7 kali Very High : Dari 10 proyek terakhir, analisis diproyek ini setidaknya pernah berpartisipasi sebanyak 9 kali</p>		✓				
PCAP	<p>PCAP (Kemampuan Programmer): Penilaian <i>cost driver</i> terkait kemampuan programmer dalam efisiensi penulisan kode program, ketelitian, dan kemampuan untuk berkomunikasi dan bekerja sama sebagai sebuah tim. Dengan kata lain sudah berapa banyak proyek di mana <i>programmer</i> tersebut ikut terlibat didalamnya.</p> <p>Very Low : Dari 10 proyek terakhir, programmer diproyek ini setidaknya pernah berpartisipasi sebanyak 2 kali Low : Dari 10 proyek terakhir, programmer diproyek ini setidaknya pernah berpartisipasi sebanyak 4 kali Nominal : Dari 10 proyek terakhir, programmer diproyek ini setidaknya pernah berpartisipasi sebanyak 6 kali High : Dari 10 proyek terakhir, programmer diproyek ini setidaknya pernah berpartisipasi sebanyak 7 kali Very High : Dari 10 proyek terakhir, programmer diproyek ini setidaknya pernah berpartisipasi sebanyak 9 kali</p>				✓		
PCON	<p>PCON (Kontinuitas Personil): Penilaian <i>cost driver</i> terkait tingkat pergantian personel tiap tahun pada proyek.</p> <p>Very Low : 48% per tahun Low : 24% per tahun Nominal : 12% per tahun High : 6% per tahun Very High : 3% per tahun</p>			~			
RUSE	<p>RUSE (Pengembangan kode yang diarahkan untuk penggunaan kembali): Penilaian <i>cost driver</i> terkait tingkat upaya yang diperlukan untuk mengembangkan komponen yang dimaksudkan untuk digunakan kembali pada proyek-proyek yang sedang berjalan atau proyek di masa mendatang.</p> <p>Low : Tidak digunakan dalam proyek, program dan produk manapun Nominal : Tidak digunakan dalam proyek lain</p>						

	<p>High : Tidak digunakan dalam program lain Very High : Tidak digunakan dalam produk lain Extra High : Akan digunakan dalam produk lain</p>							✓
TIME	<p>TIME (Kendala waktu eksekusi): Penilaian <i>cost driver</i> terkait persentase kendala waktu eksekusi yang diharapkan dapat digunakan pada sistem perangkat lunak.</p> <p>Nominal : 50% dari presentase waktu eksekusi yang tersedia High : 70% dari presentase waktu eksekusi yang tersedia Very High : 85% dari presentase waktu eksekusi yang tersedia Extra High : 95% dari presentase waktu eksekusi yang tersedia</p>						✓	
STOR	<p>STOR (Kendala Penyimpanan Utama): Penilaian <i>cost driver</i> terkait persentase tingkat kendala penyimpanan utama yang dikenakan pada sistem perangkat lunak.</p> <p>Nominal : 50% dari penyimpanan yang tersedia High : 70% dari penyimpanan yang tersedia Very High : 85% dari penyimpanan yang tersedia Extra High : 95% dari penyimpanan yang tersedia</p>						✓	
PVOL	<p>PVOL (Kompleksitas dari hardware dan software): Penilaian <i>cost driver</i> terkait perubahan yang terjadi pada <i>hardware</i> dan <i>software</i> (OS, DBMS) pada kurun waktu tertentu.</p> <p>Low : Jika terjadi perubahan kecil selama 12 bulan, perubahan besar selama 1 bulan Nominal : Jika terjadi perubahan kecil 6 bulan, perubahan besar selama 2 minggu High : Jika terjadi perubahan kecil 2 bulan, perubahan besar selama 1 minggu Very High : Jika terjadi perubahan kecil 2 minggu, perubahan besar selama 2 hari</p>						✓	
AEXP	<p>AEXP (Pengalaman tim dalam mengembangkan aplikasi): Penilaian <i>cost driver</i> terkait tingkat pengalaman kerja tim proyek pada suatu proyek pengembangan aplikasi sistem perangkat lunak atau subsistem.</p> <p>Very Low : 2 bulan Low : 6 bulan Nominal : 1 tahun High : 3 tahun Very High : 6 tahun</p>						✓	
PEXP	<p>PEXP (Pengalaman tim dalam pengembangan terkait platform aplikasi): Penilaian <i>cost driver</i> terkait pemahaman tim dalam menggunakan platform, interface, database, jaringan & middleware.</p>						✓	

	<p>Very Low : 2 bulan Low : 6 bulan Nominal : 1 tahun High : 3 tahun Very High : 6 tahun</p>						
LTEX	<p>LTEX: Penilaian <i>cost driver</i> terkait pengalaman dalam pemrograman dengan bahasa tertentu dan pemanfaatan <i>CASE tools</i> dari tim proyek dalam mngembangkan perangkat lunak.</p> <p>Very Low : 2 bulan Low : 6 bulan Nominal : 1 tahun High : 3 tahun Very High : 6 tahun</p>	✓					
TOOL	<p>TOOL (Penggunaan software bantuan): Penilaian <i>cost driver</i> terkait penggunaan <i>CASE tool</i> dalam pengembangan perangkat lunak pada proyek, seperti dari mengubah kode yang sederhana menjadi terintegrasi.</p> <p>Very Low : Edit, debug Low : Sederhana, backend case, integrasi dalam sekala kecil Nominal : Basic lifecycle tool, cukup integrasi High : kuat, Mature lifecycle tool, cukup integrasi Very High : Kuat, matang, terintegrasi dengan baik</p>		✓				
SITE	<p>SITE: Penilaian <i>cost driver</i> terkait cara distribusi yang digunakan dalam pengembangan perangkat lunak pada proyek, apakah masih via telepon, fax, email, video conference, atau bahkan sudah menggunakan interactive multimedia.</p> <p>Very Low : Komunikasi telepon dan email Low : Komunikasi telepon pribadi dan FAX Nominal : Komunikasi melalui email pribadi High : Komunikasi melalui wideband electronic communications, seperti smartphone Very High : Komunikasi melalui wideband electronic communications, video conference Extra High : Komunikasi melalui interactive multimedia, seperti presentasi yang interactive, quiz interactive, website dsb.</p>			✓			
SCED	<p>SCED: Penilaian <i>cost driver</i> terkait tingkat persentase dari percepatan atau kemunduran jadwal terhadap jadwal suatu proyek yang telah ditetapkan sebelumnya.</p>						

Very Low : 75% dari waktu normal Low : 85% dari waktu normal Nominal : 100% High : 130% dari waktu normal Very High : 160% dari waktu normal				~				
----------------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--	---	--	--	--	--


N. Irsyad

LAMPIRAN D REKAP HASIL PENYEBARAN KUESIONER

1. SCALE FACTOR

Tabel 1 Kategori Scale Factor

No	Nama	Skala Faktor				
		Precedented ness	Flexibility	Risk Resolution	Team Cohesion	Process Maturity
1	M. Irsyad	HIGH	NOMINAL	NOMINAL	VERY HIGH	LOW
2	Fauzan Tolabi	NOMINAL	HIGH	LOW	VERY HIGH	NOMINAL
3	Agung Surya	HIGH	NOMINAL	NOMINAL	VERY HIGH	LOW
4	Nadya Arimurti	LOW	NOMINAL	LOW	HIGH	NOMINAL
5	Lia Hasanah	VERY LOW	VERY HIGH	LOW	VERY HIGH	NOMINAL

Tabel 2 Nilai Kategori Scale Factor

No	Nama	Skala Faktor				
		Precedented ness	Flexibility	Risk Resolution	Team Cohesion	Process Maturity
1	M. Irsyad	2.48	3.04	4.24	1.10	6.24
2	Fauzan Tolabi	3.72	2.03	5.65	1.10	4.68
3	Agung Surya	2.48	3.04	4.24	1.10	6.24
4	Nadya Arimurti	4.96	3.04	5.65	2.19	4.68
5	Lia Hasanah	6.20	1.01	5.65	1.10	4.68

2. EFFORT ADJUDMENT MULTIPLIER

Tabel 3 Kategori Effort Adjudment Multiplier

No	Skala Faktor	Nama				
		M. Irsyad	Fauzan Tolabi	Agung surya	Nadya Arimurti	Lia Hasanah
1	RELY	VERY HIGH	LOW	VERY HIGH	HIGH	LOW
2	DATA	NOMINAL	NOMINAL	NOMINAL	NOMINAL	NOMINAL
3	CPLX	LOW	HIGH	LOW	LOW	HIGH
4	RUSE	VERY HIGH	VERY HIGH	VERY HIGH	NOMINAL	EXTRA HIGH
5	DOCU	NOMINAL	NOMINAL	NOMINAL	NOMINAL	NOMINAL
6	TIME	NOMINAL	NOMINAL	NOMINAL	NOMINAL	NOMINAL
7	STOR	NOMINAL	NOMINAL	NOMINAL	HIGH	NOMINAL
8	PVOL	LOW	LOW	LOW	NOMINAL	VERY HIGH
9	ACAP	LOW	VERY HIGH	LOW	NOMINAL	NOMINAL



Tabel 3 Kategori Effort Adjudment Multiplier (Lanjutan)

No	Skala Faktor	Nama				
		M. Irsyad	Fauzan Tolabi	Agung surya	Nadya Arimurti	Lia Hasanah
10	PCAP	VERY HIGH	VERY HIGH	VERY HIGH	VERY HIGH	VERY HIGH
11	PCON	NOMINAL	VERY LOW	NOMINAL	VERY LOW	VERY LOW
12	AEXP	LOW	NOMINAL	LOW	VERY LOW	LOW
13	PEXP	LOW	VERY HIGH	LOW	VERY LOW	HIGH
14	LTEX	VERY LOW	VERY HIGH	VERY LOW	NOMINAL	NOMINAL
15	TOOL	NOMINAL	LOW	NOMINAL	LOW	VERY LOW
16	SITE	HIGH	LOW	HIGH	HIGH	NOMINAL
17	SCED	NOMINAL	NOMINAL	NOMINAL	LOW	VERY LOW

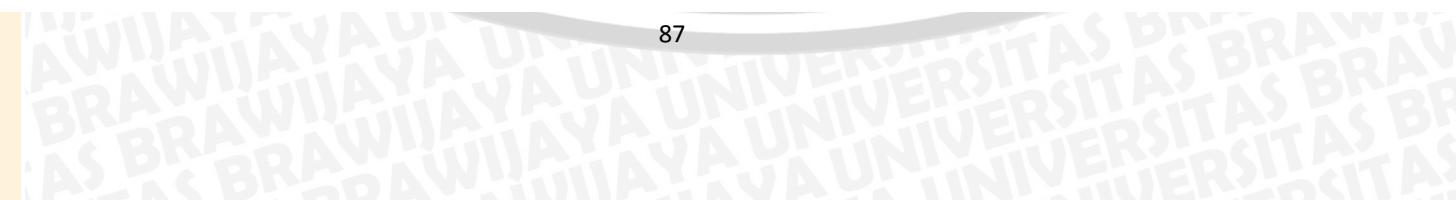
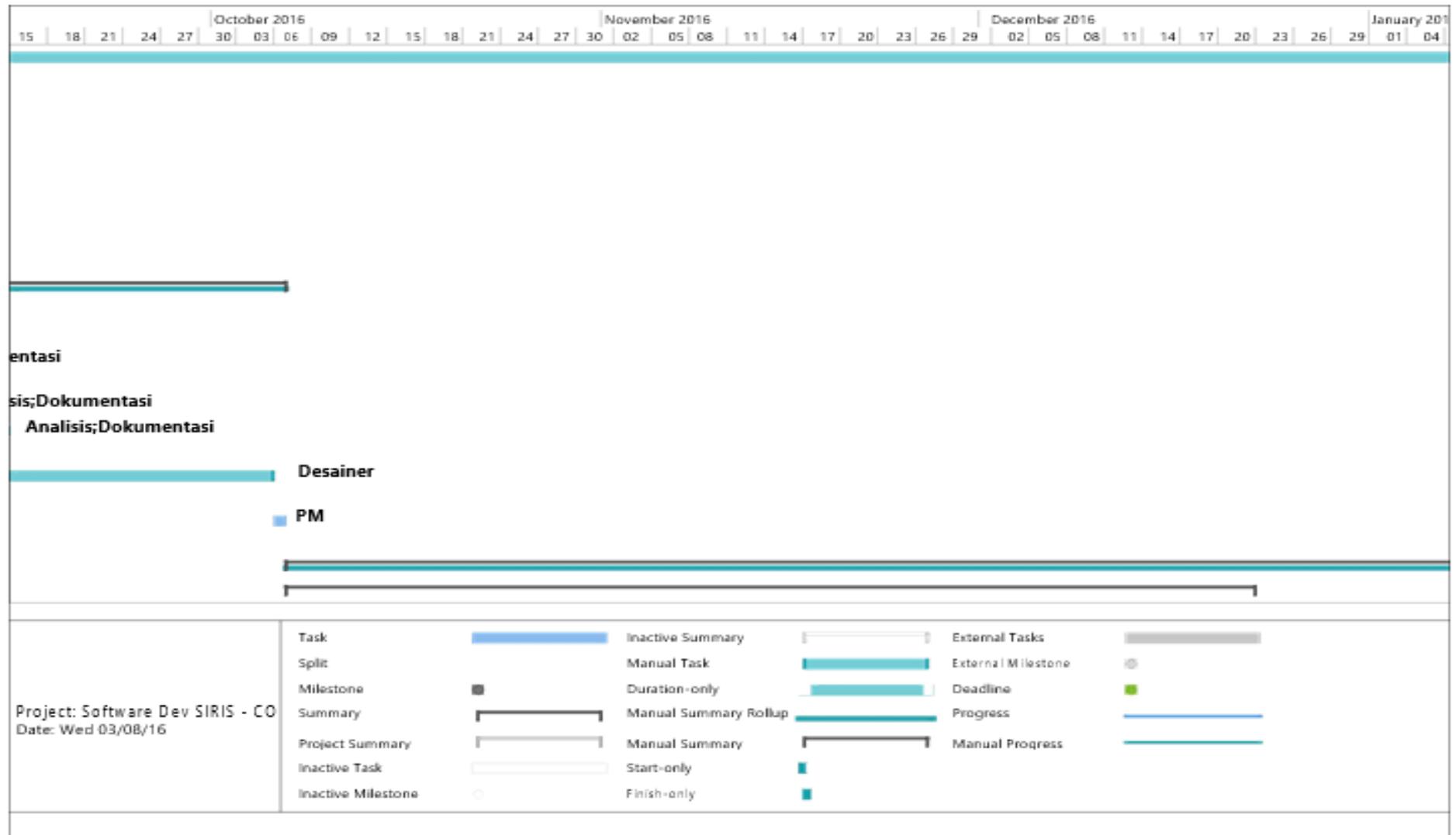
Tabel 4 Nilai Kategori Effort Adjudment Multiplier

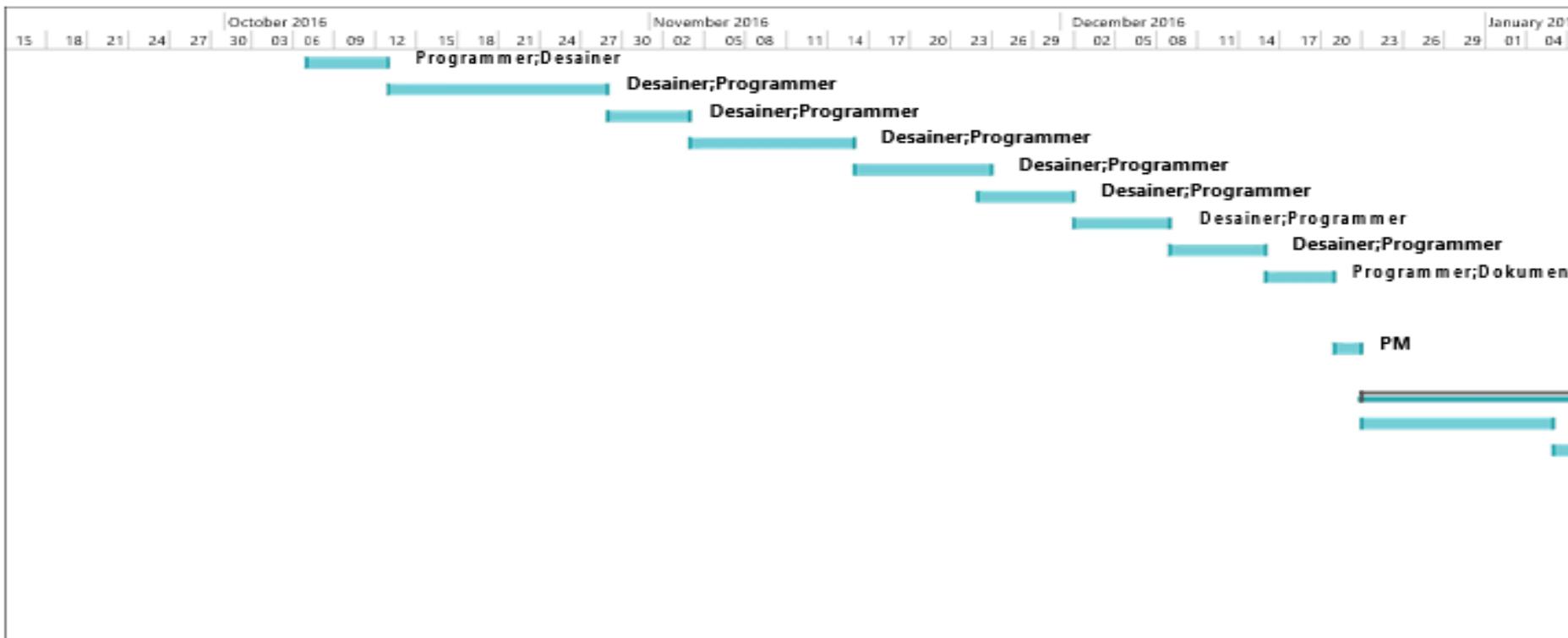
No	Skala Faktor	Nama				
		M. Irsyad	Fauzan Tolabi	Agung surya	Nadya Arimurti	Lia Hasanah
1	RELY	1,26	0,92	1,26	1,1	0,92
2	DATA	1	1	1	1	1
3	CPLX	0,87	1,17	0,87	0,87	1,17
4	RUSE	1,15	1,15	1,15	1	1,24
5	DOCU	1	1	1	1	1
6	TIME	1	1	1	1	1
7	STOR	1	1	1	1,05	1
8	PVOL	0,87	0,87	0,87	1	1,3
9	ACAP	1,19	0,71	1,19	1	1
10	PCAP	0,76	0,76	0,76	0,76	0,76
11	PCON	1	1,29	1	1,29	1,29
12	AEXP	1,1	1	1,1	1,22	1,1
13	PEXP	1,09	0,85	1,09	1,19	0,91
14	LTEX	1,2	0,84	1,2	1	1
15	TOOL	1	1,09	1	1,09	1,17
16	SITE	0,93	1,09	0,93	0,93	1
17	SCED	1	1	1	1,14	1,43

Lampiran E
GanttChart

ID	Task Mod	Task Name	Duration	Start	Finish	Resource Names	August 2016												September 2016			
							01	04	07	10	13	16	19	22	25	28	31	03	06	09	12	
1		Software development	270 days	Mon 01/08/16	Fri 11/08/17		[Gantt bar for Software development]															
2		Tahap Awal Proyek	8 days	Mon 01/08/16	Wed 10/08/16		[Gantt bar for Tahap Awal Proyek]															
3		Diskusi Sistem	2 days	Mon 01/08/16	Tue 02/08/16	PM;Dokumentasi;Analisis	[Gantt bar for Diskusi Sistem]															
4		Definis Kebutuhan	4 days	Wed 03/08/16	Mon 08/08/16	Dokumentasi;Analisis	[Gantt bar for Definis Kebutuhan]															
5		Kontrak Perjanjian	1 day	Tue 09/08/16	Tue 09/08/16	PM;Dokumentasi	[Gantt bar for Kontrak Perjanjian]															
6		Milestone: dokumen perjanjian	1 day	Wed 10/08/16	Wed 10/08/16	PM	[Gantt bar for Milestone: dokumen perjanjian]															
7		Perancangan	41 days	Thu 11/08/16	Thu 06/10/16		[Gantt bar for Perancangan]															
8		Analisis Kebutuhan	7 days	Thu 11/08/16	Fri 19/08/16	Analisis;Dokumenta	[Gantt bar for Analisis Kebutuhan]															
9		Membuat Usecase	4 days	Mon 22/08/16	Thu 25/08/16	Analisis;Dokumenta	[Gantt bar for Membuat Usecase]															
10		Membuat Data Flow Diagram	5 days	Fri 26/08/16	Thu 01/09/16	Analisis;Dokumenta	[Gantt bar for Membuat Data Flow Diagram]															
11		Membuat schema DB	5 days	Fri 02/09/16	Thu 08/09/16	Analisis;Dokumenta	[Gantt bar for Membuat schema DB]															
12		Membuat Entity Relationship Diagram	4 days	Fri 09/09/16	Wed 14/09/16	Analisis;Dokumenta	[Gantt bar for Membuat Entity Relationship Diagram]															
13		Mockup Front-End and Back-End	15 days	Thu 15/09/16	Wed 05/10/16	Desainer	[Gantt bar for Mockup Front-End and Back-End]															
14		Milestone: Dokumen perancangan	1 day	Thu 06/10/16	Thu 06/10/16	PM	[Gantt bar for Milestone: Dokumen perancangan]															
15		Development Sistem	115 days	Fri 07/10/16	Thu 16/03/17		[Gantt bar for Development Sistem]															
16		Front-End Sistem	55 days	Fri 07/10/16	Thu 22/12/16		[Gantt bar for Front-End Sistem]															

Project: Software Dev SIRIS - CO Date: Wed 03/08/16	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Inactive Task		Start-only			
	Inactive Milestone		Finish-only			





Project: Software Dev SIRIS - CO
Date: Wed 03/08/16

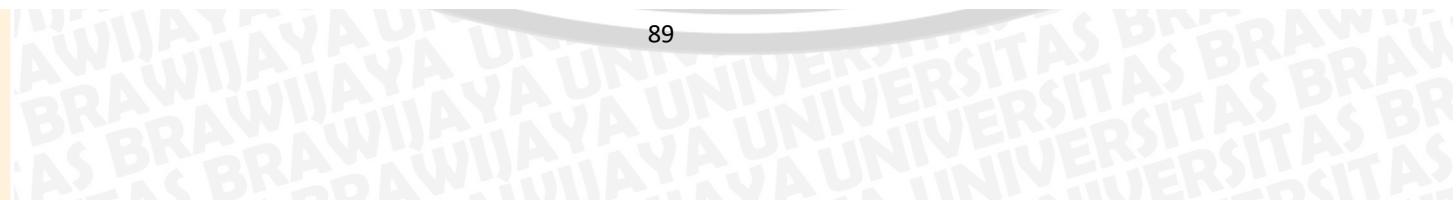
Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			



15 | 18 | 21 | 24 | 27 | **October 2016** | 30 | 03 | 06 | 09 | 12 | 15 | 18 | 21 | 24 | 27 | 30 | **November 2016** | 02 | 05 | 08 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | **December 2016** | 02 | 05 | 08 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | **January 2017** | 01 | 04



Project: Software Dev SIRIS - CO Date: Wed 03/08/16	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Start-only		Manual Progress	
	Inactive Task		Finish-only			
Inactive Milestone						

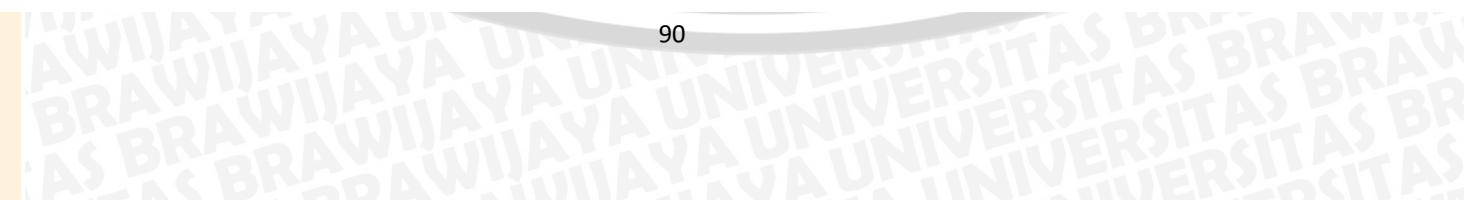




17	February 2017										March 2017										April 2017																
07	10	13	16	19	22	25	28	31	03	06	09	12	15	18	21	24	27	02	05	08	11	14	17	20	23	26	29	01	04	07	10	13	16	19	22	25	28

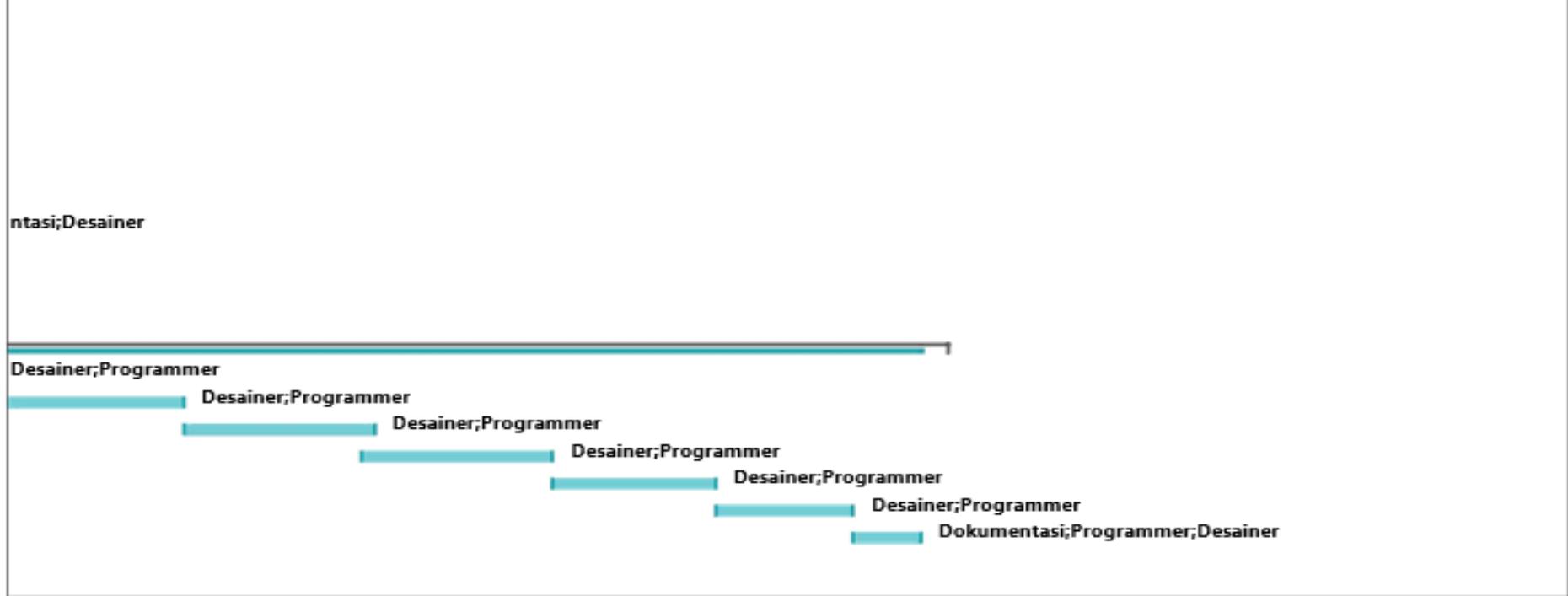


Project: Software Dev SIRIS - CO Date: Wed 03/08/16	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		Manual Summary		Manual Progress	
	Inactive Task		Start-only			
Inactive Milestone		Finish-only				



17 | February 2017 | March 2017 | April 2017

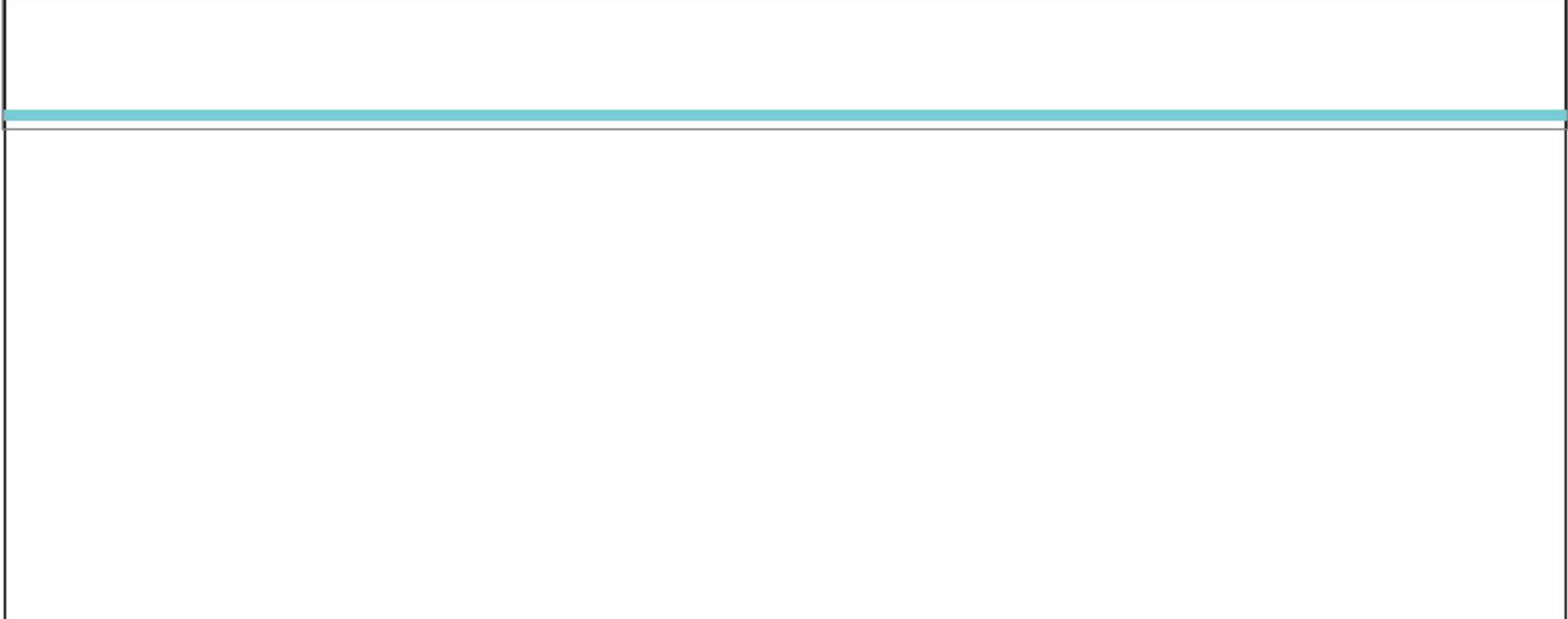
07 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 03 | 06 | 09 | 12 | 15 | 18 | 21 | 24 | 27 | 02 | 05 | 08 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 01 | 04 | 07 | 10 | 13 | 16 | 19 | 22 | 25 | 28



Project: Software Dev SIRIS - CO
Date: Wed 03/08/16

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

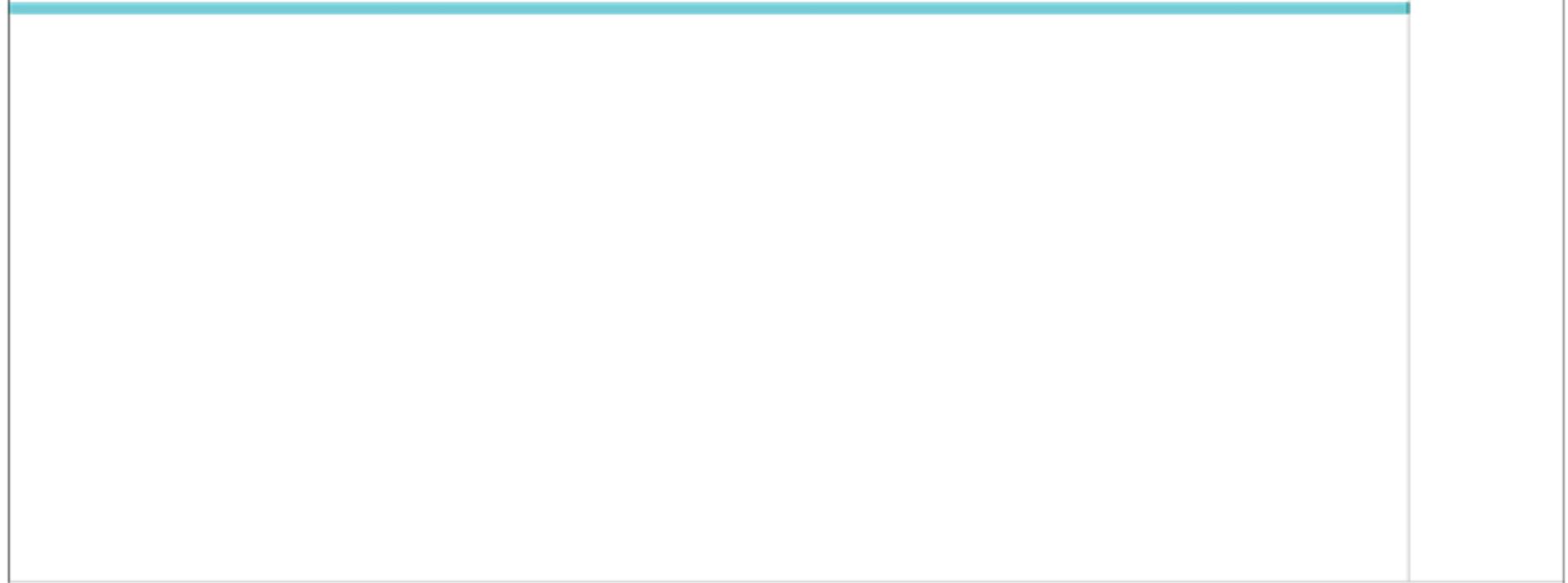
17 | February 2017 | March 2017 | April 2017
 07 | 10 | 13 | 16 | 19 | 22 | 25 | 28 | 31 | 03 | 06 | 09 | 12 | 15 | 18 | 21 | 24 | 27 | 02 | 05 | 08 | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 01 | 04 | 07 | 10 | 13 | 16 | 19 | 22 | 25 | 28



Project: Software Dev SIRIS - CO Date: Wed 03/08/16	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		External Summary		External Progress	
Inactive Task		Start-only				
Inactive Milestone		Finish-only				

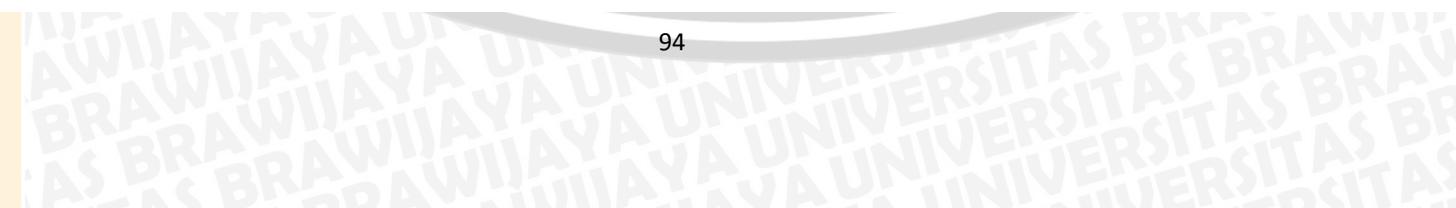


May 2017 | June 2017 | July 2017 | August 2017
 01 04 07 10 13 16 19 22 25 28 31 03 06 09 12 15 18 21 24 27 30 03 06 09 12 15 18 21 24 27 30 02 05 08 11 14 17 20

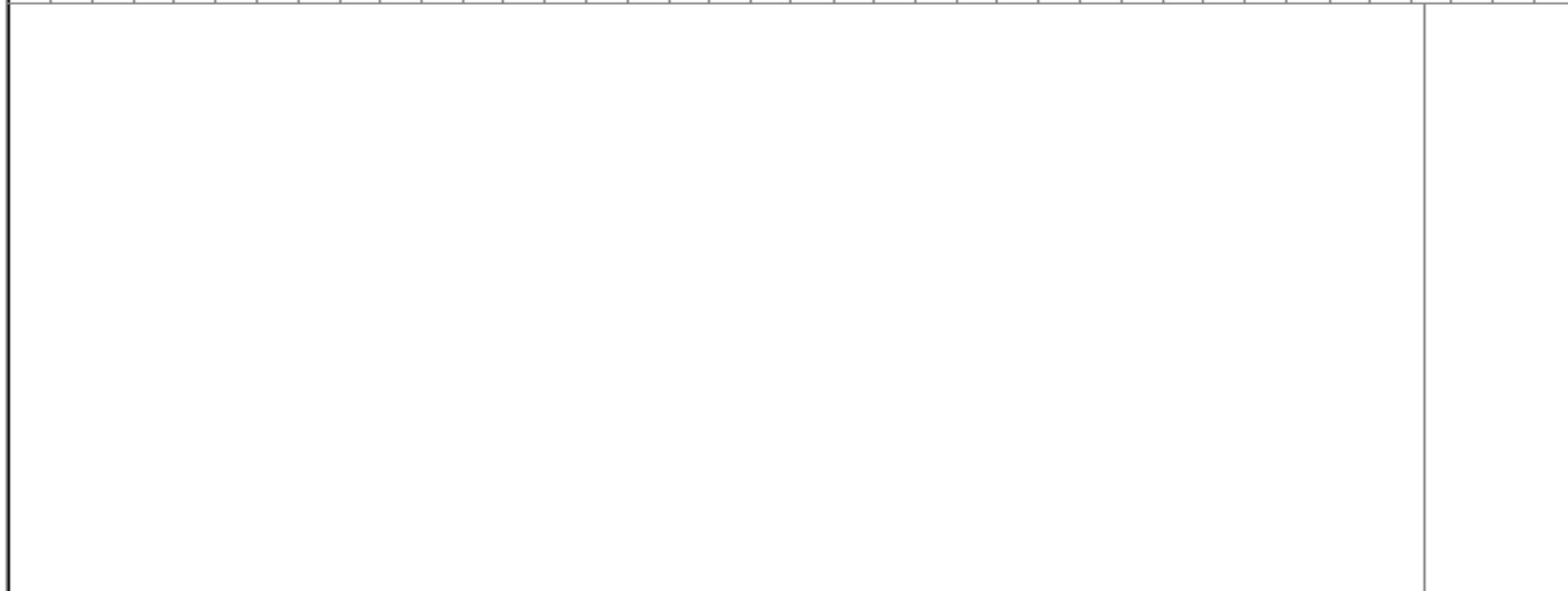


Project: Software Dev SIRIS - CO
Date: Wed 03/08/16

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			



May 2017 | June 2017 | July 2017 | August 2017
 01 04 07 10 13 16 19 22 25 28 31 03 06 09 12 15 18 21 24 27 30 03 06 09 12 15 18 21 24 27 30 02 05 08 11 14 17 20



Project: Software Dev SIRIS - CO Date: Wed 03/08/16	Task		Inactive Summary		External Tasks	
	Split		Manual Task		External Milestone	
	Milestone		Duration-only		Deadline	
	Summary		Manual Summary Rollup		Progress	
	Project Summary		External Summary		External Progress	
	Inactive Task		Start-only			
Inactive Milestone		Finish-only				

