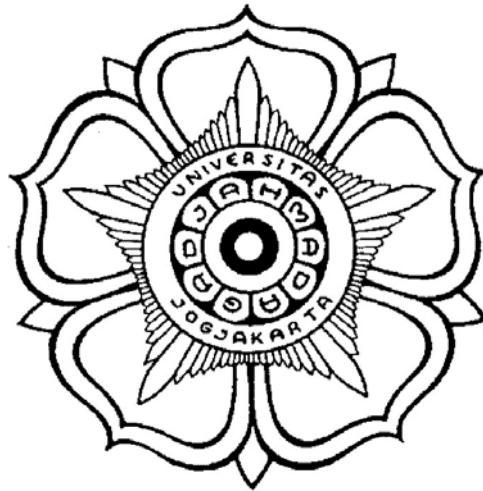


**IMPLEMENTASI METRIK PADA PENGEMBANGAN
PERANGKAT LUNAK**

SKRIPSI



Disusun oleh :

**Wahyu Wijanarko
01/148490/TK/26419**

**JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2005**

LEMBAR PENGESAHAN

SKRIPSI

IMPLEMENTASI METRIK PADA PENGEMBANGAN PERANGKAT LUNAK

Diajukan sebagai syarat untuk memperoleh gelar Sarjana Strata 1
pada Jurusan Teknik Elektro Fakultas Teknik
Universitas Gadjah Mada Yogyakarta

Telah diperiksa dan disetujui

Pada Tanggal :

di Yogyakarta

Pembimbing I

Pembimbing II

Ir. Lukito Edi Nugroho, M.Sc., Ph.D
NIP: 131 963 570

Selo, S.T., M.T., M.Sc
NIP: 132 163 781

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa, sehingga penulis dapat menyelesaikan tugas akhir dengan judul “*Implementasi Metrik Pada Pengembangan Perangkat Lunak*” sebagai salah satu syarat untuk mencapai gelar kesarjanaan pada Jurusan Teknik Elektro Fakultas Teknik Universitas Gadjah Mada Yogyakarta.

Penulisan tugas akhir ini tidak lepas dari dukungan bantuan dan bimbingan berbagai pihak. Dalam kesempatan ini penulis mengucapkan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Ir. Lukito Edi Nugroho, M.Sc., Ph.D, selaku dosen pembimbing yang telah banyak memberikan bimbingan, dukungan, dan semangat hingga terselesaikannya tugas akhir ini.
2. Selo, S.T., M.T., M.Sc., selaku dosen pembimbing yang telah banyak memberikan bimbingan, dukungan, dan arahan, hingga terselesaikannya tugas akhir ini.
3. Semua rekan-rekan KMTE yang telah mendukung penyusunan tugas akhir ini.
4. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung hingga terselesaikannya tugas akhir ini.

Penulis sangat mengharapkan kritik dan saran dari pembaca untuk penyempurnaan karya-karya yang akan datang. Harapan dari penulis, semoga tugas akhir ini dapat bermanfaat bagi siapa saja yang menggunakannya.

Yogyakarta, September 2005

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	v
DAFTAR TABEL.....	vi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penulisan Tugas Akhir.....	4
1.5 Metodologi Penyusunan Tugas Akhir	4
1.6 Sistematika Penulisan Tugas Akhir	5
BAB II METRIK DAN ESTIMASI PERANGKAT LUNAK.....	6
2.1 Metrik Dalam Proses dan Proyek	6
2.1.1 Metrik Proses dan Peningkatan Proses Perangkat Lunak.....	7
2.1.2 Metrik Proyek	9
2.2 Pengukuran Perangkat Lunak	11
2.2.1 Metrik Berorientasi Ukuran	12
2.2.2 Metrik Berorientasi Fungsi	12
2.3 Estimasi Proyek Perangkat Lunak	14
2.3.1 Estimasi Berbasis Masalah	15
2.3.2 Estimasi Berbasis Proses.....	16
2.4 Model Estimasi Empiris.....	17
2.4.1 Struktur Model-Model Estimasi	17
2.4.2 Model COCOMO.....	18
2.4.3 Persamaan pada Perangkat Lunak	20
2.5 Konversi Waktu Tenaga Kerja.....	21
BAB III DESAIN SISTEM	23
3.1 Deskripsi Umum	23
3.2 Spesifikasi Keperluan Perangkat Lunak	23
3.2.1 Sistem yang Sudah Ada	23
3.2.2 Sasaran Pengguna	25
3.2.3 Sistem yang Diusulkan	25
3.2.4 Lingkungan Perangkat Lunak	26
3.2.5 Lingkungan Perangkat Keras dan Jaringan Komputer	26
3.2.6 Spesifikasi Keperluan Fungsional.....	27
3.2.7 Dokumentasi Sistem	31
3.3 Spesifikasi Desain Perangkat Lunak.....	31
3.3.1 Desain Data.....	31
3.3.2 Rancangan Antarmuka.....	32
3.3.3 Rancangan Menu	35
3.3.4 Rancangan Komponen	36
BAB IV IMPLEMENTASI DAN PEMBAHASAN	51
4.1 Petunjuk Umum Estimasi Perangkat Lunak	51
4.2 Implementasi Berdasarkan Teori	53
4.2.1 Estimasi Berbasis LOC.....	53
4.2.2 Estimasi Berbasis FP.....	57

4.2.3	Estimasi Berbasis Proses.....	58
4.2.4	Model Estimasi Empiris.....	59
4.3	Hasil Pengamatan Lapangan.....	61
4.4	Perbandingan Hasil Estimasi	62
BAB V PENUTUP		65
5.1	Kesimpulan	65
5.2	Saran	65
DAFTAR PUSTAKA		66

DAFTAR GAMBAR

Gambar 2.1 Lingkungan pengembangan perangkat lunak	7
Gambar 2.2 Grafik tulang ikan analisa proses pengembangan perangkat lunak	8
Gambar 3.1 Alur Proses Perencanaan Daerah	24
Gambar 3.2 Usulan Jaringan Komputer.....	26
Gambar 3.3 Diagram E-R rancangan basis data	32
Gambar 3.4 Perancangan antarmuka modul	33
Gambar 3.5 Rancangan antarmuka tampilan data	34
Gambar 3.6 Rancangan tampilan pengisian data.....	35
Gambar 3.7 Diagram alir proses login.....	37
Gambar 3.8 Diagram alir proses pengubahan sandi	38
Gambar 3.9 Diagram alir proses administrasi data tahun anggaran	39
Gambar 3.10 Diagram alir proses administrasi data bidang	40
Gambar 3.11 Diagram alir proses administrasi data unit kerja.....	41
Gambar 3.12 Diagram alir proses administrasi data penilai	42
Gambar 3.13 Diagram alir proses administrasi data kriteria penilaian.....	43
Gambar 3.14 Diagram alir proses administrasi data subkriteria penilaian	44
Gambar 3.15 Diagram alir proses administrasi data pengguna	45
Gambar 3.16 Diagram alir proses administrasi data program	46
Gambar 3.17 Diagram alir proses administrasi data usulan kegiatan	47
Gambar 3.18 Diagram alir proses penilaian	48
Gambar 3.19 Diagram alir proses pelaporan	49
Gambar 3.20 Diagram alir proses transfer data	50
Gambar 4.1 Alur kerja estimasi perangkat lunak.....	51

DAFTAR TABEL

Tabel 2.1 Koefisien model COCOMO dasar	19
Tabel 2.2 Koefisien model COCOMO lanjut	20
Tabel 4.1 Analisa estimasi LOC UICF	54
Tabel 4.2 Analisa estimasi LOC DBM	55
Tabel 4.3 Analisa LOC DAM	55
Tabel 4.4 Perhitungan estimasi FP	57
Tabel 4.5 Perhitungan faktor peubah kompleksitas	57
Tabel 4.6 Perhitungan estimasi berbasis proses	58
Tabel 4.7 Hasil akumulasi presensi tim pengembang	62
Tabel 4.8 Perbandingan hasil estimasi antar model	62

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pengukuran merupakan salah satu dasar dari semua disiplin ilmu teknik. Rekayasa perangkat lunak dalam IEEE Standard 610.12 didefinisikan sebagai berikut: “*The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software*”. Seperti profesi di bidang teknik pada umumnya, proses rekayasa perangkat lunak juga memiliki metrik.

Metrik digunakan oleh industri perangkat lunak untuk mengukur proses pembuatan, operasi, dan perawatan perangkat lunak. Melalui metrik, dapat diperoleh informasi-informasi berharga dan parameter-parameter sebagai bahan evaluasi yang obyektif mengenai atribut-atribut dan status dari suatu pengembangan perangkat lunak. Implementasi metrik perangkat lunak pada suatu proses pengembangan perangkat lunak dan pada suatu produk perangkat lunak melibatkan tahapan-tahapan kompleks yang memerlukan pembelajaran yang berkelanjutan, yang pada akhirnya dapat memberikan pengetahuan mengenai status dari suatu proses pembuatan perangkat lunak dan atau suatu produk dari perangkat lunak.

Dengan melakukan evaluasi pada atribut-atribut yang ada dalam perangkat lunak, dapat diperoleh status dari suatu perangkat lunak. Dari hal ini, situasi yang ada dapat diidentifikasi dan diklasifikasikan, yang dapat digunakan untuk membantu dalam mencari peluang-peluang baru yang bisa digunakan untuk

pengembangan dan perbaikan perangkat lunak. Evaluasi seperti ini pada akhirnya dapat digunakan untuk membuat perencanaan dalam perubahan-perubahan yang mungkin perlu diimplementasikan di masa yang akan datang. Atribut-atribut yang diidentifikasi ini juga dapat digunakan sebagai referensi dan bahan pertimbangan bagi proses pengembangan perangkat lunak lainnya.

Metrik perangkat lunak memiliki batasan-batasan yang luas. Metrik perangkat lunak tergantung pada atribut-atribut perangkat lunak yang ingin dinilai kuantitas dan kualitasnya. Secara umum, metrik perangkat lunak dibagi dalam dua kelas yang berbeda, yaitu metrik yang digunakan pada proyek pengembangan perangkat lunak dan metrik yang digunakan pada produk perangkat lunak.

Metrik pada proyek pengembangan perangkat lunak berhubungan dengan tenaga dan pikiran yang diperlukan untuk menyelesaikan proyek, sumber daya yang digunakan untuk menyelesaikannya, dan metodologi yang diterapkan, misalnya: waktu yang diperlukan untuk menyelesaikan, tenaga ahli yang diperlukan, biaya-biaya yang dikeluarkan, dan metodologi yang digunakan dalam pembuatan perangkat lunak.

Banyak tipe dari atribut-atribut dari perangkat lunak yang dapat diukur. Metrik yang diaplikasikan sangat bergantung pada sifat alamiah dari produk perangkat lunak yang dibuat. Misalnya, ingin diketahui seberapa banyak kebutuhan yang dimiliki oleh suatu proyek, spesifikasi perangkat lunak (untuk memperkecil ambiguitas) dan kelengkapan yang digunakan untuk memenuhi seluruh fungsi yang diperlukan. Pada produk dari suatu aplikasi, mungkin ingin diketahui jumlah baris kode, kerumitan, fungsionalitas yang dipenuhi, jumlah

kesalahan yang mungkin dapat terjadi, dan jumlah ujicoba yang dilakukan untuk memastikan bahwa semua keperluan sudah diimplementasikan. Di sisi lain, reliabilitas dari suatu perangkat lunak dapat diukur ketika produk tersebut telah didistribusikan kepada konsumen.

Para ahli di bidang rekayasa perangkat lunak sebenarnya belum berhasil memutuskan metrik yang tepat yang dapat diterima secara universal. Selain itu tiap orang biasanya menggunakan berbagai macam metode spesifik untuk mengukur atribut-atribut yang berbeda dari perangkat lunak yang mereka buat. Hal itu mengakibatkan banyak kontroversi yang timbul karena metrik-metrik yang telah diterapkan sangat beragam. Para ahli di bidang rekayasa perangkat lunak masih harus melalui jalan panjang untuk mencapai suatu rumusan metrik yang tepat untuk melakukan evaluasi terhadap proses pengembangan dan produk dari perangkat lunak.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas, maka dalam tugas akhir ini masalah yang dirumuskan adalah implementasi teori metrik perangkat lunak dalam suatu proyek pengembangan perangkat lunak yang nyata.

1.3 Batasan Masalah

Sesuai dengan rumusan yang telah dipaparkan, maka batasan yang diberlakukan dalam tugas akhir ini adalah implementasi teori mengenai metrik perangkat lunak pada salah satu proyek pengembangan perangkat lunak yang sedang dikerjakan. Metrik digunakan untuk melakukan estimasi tenaga kerja yang

diperlukan untuk mengembangkan perangkat lunak berdasarkan desain yang diajukan. Metrik diterapkan pada fungsionalitas perangkat lunak. Bahasan tidak mencakup estimasi elemen-elemen fisik seperti spesifikasi perangkat keras.

1.4 Tujuan Penulisan Tugas Akhir

Tujuan dari tugas akhir ini adalah mengimplementasikan teori mengenai metrik perangkat lunak pada suatu proyek pembuatan perangkat lunak. Hasil yang diharapkan adalah petunjuk yang digunakan untuk menghitung besar sumber daya manusia yang diperlukan dalam suatu pengembangan perangkat lunak, yang dihitung dengan satuan orang-jam.

1.5 Metodologi Penyusunan Tugas Akhir

Metode yang digunakan dalam penyusunan tugas akhir ini adalah

1. Tinjauan pustaka, mempelajari buku, artikel, dan situs yang terkait dengan metrik pada perangkat lunak.
2. Wawancara, melakukan studi dengan metode wawancara kepada dosen ataupun praktisi yang berhubungan dengan permasalahan yang dibahas dalam tugas akhir ini.
3. Pengumpulan data, mengumpulkan data-data yang terkait dengan proyek pembuatan perangkat lunak yang sedang dikerjakan.
4. Implementasi, mengimplementasikan teori metrik perangkat lunak pada proyek pengembangan perangkat lunak yang sedang dikerjakan.

5. Penulisan tugas akhir, dimulai dari pembuatan proposal sampai dengan pembuatan kesimpulan dari implementasi metrik perangkat lunak yang telah dilakukan.

1.6 Sistematika Penulisan Tugas Akhir

Penulisan tugas akhir ini tersusun dalam 5 (lima) bab dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Bab Pendahuluan berisi latar belakang masalah, rumusan masalah, tujuan penyusunan tugas akhir, metodologi, dan sistematika penyusunan tugas akhir.

BAB II Dasar Teori

Dasar Teori berisi beberapa teori yang mendasari penyusunan tugas akhir ini. Adapun yang dibahas dalam bab ini adalah teori yang berkaitan dengan metrik dan estimasi perangkat lunak.

BAB III Desain Sistem

Pada bab ini diuraikan deskripsi dan desain perangkat lunak yang sedang dikembangkan. Desain perangkat lunak meliputi *Software Requirement Specification* (SRS) dan *Software Design Specification* (SDS).

BAB IV Implementasi dan Pembahasan

Berisi implementasi dan evaluasi terhadap metrik perangkat lunak yang telah diterapkan dalam proses pembuatan perangkat lunak.

BAB V Penutup

Bab Penutup berisi Kesimpulan dan Saran.

BAB II

METRIK DAN ESTIMASI PERANGKAT LUNAK

2.1 Metrik Dalam Proses dan Proyek

Pengukuran adalah salah satu hal yang menjadi bagian dari dunia keteknikan, dengan beberapa di antaranya adalah: pengukuran daya, pengukuran dimensi fisik, pengukuran berat, pengukuran frekuensi, dan sebagainya. Namun demikian, pengukuran jarang dilakukan di dalam dunia rekayasa perangkat lunak. Permasalahan yang sering ditemui adalah adanya kesulitan dalam menetapkan nilai dari obyek yang diukur dan sulitnya menetapkan parameter-parameter yang bisa diukur. Salah satu cara yang bisa digunakan adalah dengan melakukan pengumpulan metrik-metrik dalam suatu proses atau proyek pengembangan perangkat lunak untuk dapat diperoleh suatu indikator proses dan indikator proyek.

Indikator proses digunakan oleh organisasi atau perusahaan yang bergerak dalam bidang rekayasa perangkat lunak untuk memperoleh data-data yang dapat digunakan untuk meningkatkan efisiensi dalam proses pengembangan perangkat lunak.

Indikator proyek digunakan oleh manajer proyek untuk:

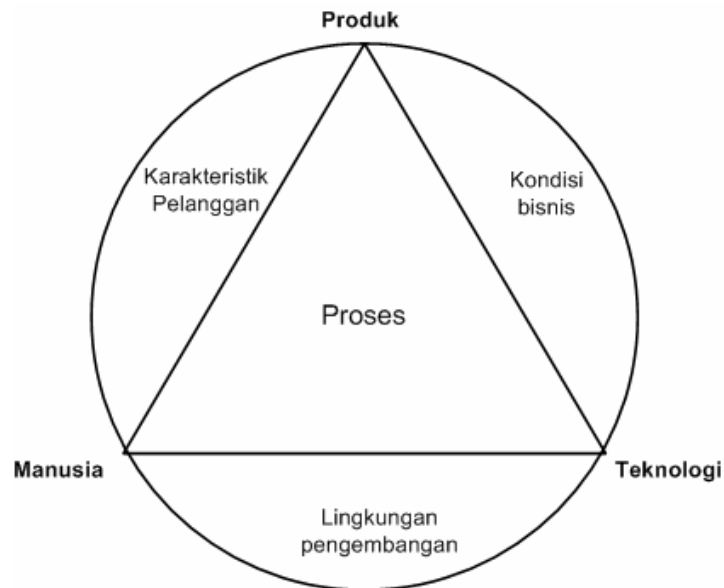
1. Memperoleh status dari proyek yang sedang dikerjakan
2. Menelusuri resiko-resiko yang bisa terjadi
3. Mengatasi masalah sebelum terjadi hal-hal yang bersifat kritis
4. Mengubah alur kerja atau proses apabila diperlukan

5. Melakukan evaluasi kemampuan tim dalam proyek untuk melakukan kontrol terhadap kualitas produk perangkat lunak

Dalam beberapa kasus, suatu metrik yang sama dapat digunakan secara bersama-sama sebagai indikator proses dan indikator proyek.

2.1.1 Metrik Proses dan Peningkatan Proses Perangkat Lunak

Menurut Paulish [1994], proses pada pengembangan perangkat lunak dipaparkan dalam gambar 2.1 sebagai berikut:



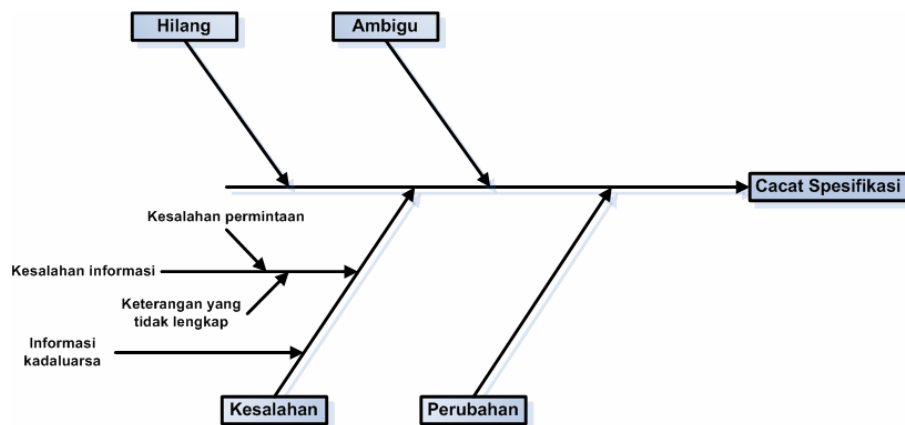
Gambar 2.1 Lingkungan pengembangan perangkat lunak

Dari gambar, diperoleh bahwa proses berada di tengah-tengah segitiga yang terhubung pada tiga faktor utama, yaitu: sumber daya manusia, kompleksitas produk, dan perkembangan teknologi. Dari ketiga faktor tersebut yang paling berpengaruh dalam hal kualitas dan performa proses adalah sumber daya manusia. Segitiga proses terdapat di dalam sebuah lingkaran dari kondisi lingkungan dari proses yang sedang berjalan, yang terdiri dari lingkungan pengembangan sistem,

kondisi bisnis (misalnya: *deadline*, aturan bisnis), dan karakteristik pelanggan atau pengguna (misalnya: kemudahan komunikasi).

Efisiensi dari suatu proses dapat diukur secara tidak langsung dengan cara menentukan sekumpulan metrik berdasarkan kesimpulan yang dapat diambil dari suatu proses. Kesimpulan dapat diambil dari perhitungan kesalahan-kesalahan yang belum dapat diatasi sebelum perangkat lunak diterbitkan maupun kesalahan yang dilaporkan oleh pengguna secara langsung. Kesimpulan juga dapat diambil dari ukuran produktifitas pekerjaan, tenaga kerja yang digunakan, waktu yang dihabiskan, kedisiplinan jadwal, dan ukuran-ukuran lain yang lebih spesifik.

Grady [1992] berpendapat bahwa terdapat penggunaan privat dan publik untuk data-data proses yang diperoleh. Beberapa metrik proses bisa bersifat privat untuk tim dalam proyek, namun bisa bersifat publik untuk seluruh anggota tim.



Gambar 2.2 Grafik tulang ikan analisa proses pengembangan perangkat lunak

Untuk memudahkan suatu organisasi dalam melakukan analisis proses, dibuat suatu pendekatan yang disebut dengan *Statistical Software Process Improvement* (SSPI). SSPI menggunakan analisis kesalahan yang terdapat pada perangkat lunak untuk mengumpulkan informasi mengenai semua kesalahan dan

kekurangan yang dapat terjadi pada aplikasi atau sistem. Analisis kesalahan dilakukan pada hal-hal berikut:

1. Semua kesalahan dan kekurangan yang dikategorikan berdasarkan sumber dari kesalahan
2. Pencatatan biaya untuk memperbaiki tiap kesalahan
3. Jumlah kesalahan dan kekurangan pada tiap kategori dijumlahkan dan diurutkan mulai dari yang paling banyak sampai yang paling sedikit
4. Semua biaya dari kesalahan dan kekurangan dari tiap kategori dihitung
5. Data resultan dianalisa untuk menutupi kategori-kategori yang memerlukan biaya paling besar
6. Perencanaan yang cermat dibuat untuk melakukan perubahan pada proses dengan mengurangi frekuensi kesalahan dan kekurangan yang paling banyak menghabiskan biaya.

Kumpulan dari metrik proses menghasilkan sebuah diagram ‘tulang ikan’ seperti pada gambar 2.2. Diagram yang lengkap dapat digunakan oleh organisasi perangkat lunak dalam melakukan modifikasi proses untuk mengurangi frekuensi kesalahan.

2.1.2 Metrik Proyek

Metrik proyek digunakan oleh manajer proyek dan tim pengembang untuk dapat beradaptasi dengan alur kerja dan aktifitas-aktivitas yang bersifat teknis. Metrik proyek ini pertama kali digunakan pada saat perencanaan proyek. Metrik-metrik yang dikumpulkan dari proyek-proyek di masa lalu digunakan sebagai acuan untuk melakukan estimasi pada proyek yang sedang dikerjakan. Manajer

proyek menggunakan data-data tersebut untuk melakukan pengawasan dan kendali proyek.

Metrik proyek memiliki dua macam sasaran. Pertama, metrik digunakan untuk melakukan optimisasi pada penjadwalan sehingga jadwal dapat dipersingkat. Kedua, metrik digunakan untuk menaksir dan meningkatkan kualitas produk.

Apabila kualitas meningkat, kesalahan dapat berkurang, maka jumlah revisi yang perlu dilakukan ikut berkurang, sehingga biaya produksi dapat ditekan. Menurut model yang disarankan oleh Hetzel [1993], hal-hal yang harus diukur dalam suatu proyek adalah:

1. Masukan, mengukur sumber daya (tenaga kerja, lingkungan) yang diperlukan dalam proyek
2. Keluaran, mengukur hasil yang diperoleh dari suatu proyek pada saat proses rekayasa perangkat lunak
3. Hasil, mengukur indikasi efektifitas dari produk yang dihasilkan

Sebenarnya, model di atas dapat diterapkan pada proses maupun pada proyek. Model tersebut dapat juga digunakan secara berurutan dalam suatu aktivitas kerangka kerja ke aktivitas kerangka kerja lainnya. Keluaran kerangka kerja tertentu bisa menjadi masukan bagi kerangka kerja yang lain dalam sebuah proyek. Di dalam aliran dari kerangka kerja satu ke kerangka kerja yang lain ini bisa diperoleh metrik hasil sebagai indikasi efektifitas antar kerangka kerja yang saling berhubungan.

2.2 Pengukuran Perangkat Lunak

Pengukuran dapat dipisahkan dalam dua kategori, yaitu pengukuran langsung dan pengukuran tidak langsung. Pengukuran langsung dalam proses rekayasa perangkat lunak berhubungan dengan biaya dan sumber daya yang diperlukan, misalnya: pengukuran jumlah baris kode, kecepatan eksekusi, ukuran memori, dan kesalahan yang ditemui dalam suatu periode waktu. Pengukuran tidak langsung dari suatu produk berhubungan dengan fungsionalitas, kualitas, kompleksitas, efisiensi, reliabilitas, dan lain sebagainya. Pengukuran secara langsung lebih mudah dilakukan, karena hasil dapat diperoleh secara langsung, sedangkan pengukuran tidak langsung lebih sulit dilakukan, karena harus melalui proses yang lebih kompleks.

Metrik dipisahkan menjadi metrik proses, proyek, dan produk. Metrik produk bersifat privat untuk individu dan sering dikombinasikan untuk membuat metrik proyek yang bersifat publik bagi tim pengembang. Metrik proyek kemudian dikonsolidasikan untuk membuat metrik proses yang publik untuk seluruh organisasi atau perusahaan. Kesulitan yang biasanya dihadapi adalah pada saat melakukan kombinasi pada metrik-metrik yang diukur disebabkan karena sering terjadi perbedaan metrik antara individu satu dengan individu lainnya. Masalah tersebut biasa diatasi apabila dilakukan normalisasi pada proses pengukuran. Dengan adanya normalisasi maka dapat dilakukan perbandingan metrik pada cakupan yang lebih luas.

2.2.1 Metrik Berorientasi Ukuran

Metrik berorientasi ukuran diperoleh dengan cara melakukan normalisasi ukuran kualitas dan produktivitas dengan menghitung ukuran dari perangkat lunak yang dibuat. Ukuran yang biasanya dijadikan sebagai acuan normalisasi adalah LOC (*lines of code*). Dari pengukuran jumlah LOC pada suatu perangkat lunak, dapat diperoleh:

1. Kesalahan per KLOC (ribuan LOC)
2. Kekurangan atau cacat pada spesifikasi per KLOC
3. Harga per LOC
4. Jumlah halaman dokumentasi per LOC

Selain itu, beberapa metrik yang bisa dihitung adalah:

1. Kesalahan per orang-bulan
2. LOC per orang-bulan
3. Harga per halaman dokumentasi

Menurut Jones [1986], metrik berorientasi ukuran tidak dapat diterima secara universal sebagai cara terbaik untuk mengukur proses rekayasa perangkat lunak. Alasan yang dikemukakan adalah kadang-kadang fungsionalitas program dapat dicapai dengan baris program yang lebih sedikit. Selain itu, untuk melakukan estimasi LOC harus digunakan analisis desain tingkat tinggi.

2.2.2 Metrik Berorientasi Fungsi

Metrik berorientasi fungsi menggunakan ukuran fungsionalitas yang dihasilkan oleh aplikasi sebagai nilai normalisasi. Fungsionalitas tidak dapat diukur secara langsung, sehingga untuk memperolehnya digunakan pengukuran

langsung terlebih dahulu, lalu hasil pengukuran langsung tersebut digunakan sebagai masukan. Metrik berorientasi fungsi pertama kali diusulkan oleh Albrecht [1979], yang menyarankan pengukuran yang disebut *function point* (FP). FP diperoleh dengan menggunakan hubungan empiris berdasarkan pengukuran langsung dan estimasi terhadap kompleksitas perangkat lunak.

Terdapat lima karakteristik yang digunakan sebagai acuan, yaitu:

1. Jumlah masukan (*user inputs*)
2. Jumlah keluaran (*user outputs*)
3. Jumlah permintaan (*inquiry*)
4. Jumlah berkas
5. Jumlah antarmuka eksternal

Jumlah-jumlah tersebut dikalikan dengan faktor pemberat, sesuai dengan kompleksitas (sederhana, sedang, kompleks) dari tiap karakteristik acuan. Untuk mengukurnya, digunakan persamaan:

$$\text{FP} = \text{jumlah total} \times [0.65 + 0.01 \times \Sigma (F_i)] \quad (2-1)$$

F_i ($i = 1$ sampai dengan 14) merupakan nilai peubah kompleksitas berdasarkan pertanyaan-pertanyaan yang diusulkan oleh Arthur [1985], dengan skala 0 (tidak penting) sampai dengan 5 (sangat penting):

1. Apakah sistem memerlukan *backup* dan *recovery*?
2. Apakah komunikasi data diperlukan?
3. Apakah terdapat fungsi pemrosesan terdistribusi?
4. Apakah performa sangat penting?
5. Apakah sistem akan berjalan pada lingkungan operasional yang berat?

6. Apakah sistem memerlukan data entri secara online?
7. Apakah data entri online memerlukan transaksi masukan untuk membuat operasi dengan ‘banyak layar’?
8. Apakah berkas aplikasi (*master*) diambil secara online?
9. Apakah masukan, keluaran, berkas, atau permintaan bersifat kompleks?
10. Apakah pemrosesan internal bersifat kompleks?
11. Apakah kode yang dibuat dapat dipakai ulang?
12. Apakah konversi dan instalasi termasuk di dalam desain?
13. Apakah sistem didesain untuk instalasi lebih dari satu dalam organisasi yang berbeda?
14. Apakah aplikasi didesain untuk memfasilitasi perubahan dan mudah digunakan oleh pengguna?

Setelah FP dihitung, dapat digunakan metode yang serupa dengan LOC untuk melakukan normalisasi ukuran pada produktivitas, kualitas, dan atribut-atribut lainnya.

2.3 Estimasi Proyek Perangkat Lunak

Estimasi biaya dan tenaga sukar untuk menjadi sebuah ilmu pasti. Banyak faktor seperti manusia, teknis, lingkungan, atau politik, yang dapat mempengaruhi besarnya biaya dan tenaga yang diperlukan untuk mengembangkan perangkat lunak. Bagaimanapun juga, estimasi proyek perangkat lunak dapat ditransformasikan dari sesuatu yang abstrak menjadi suatu urutan langkah-langkah sistematis yang menghasilkan perkiraan-perkiraan dengan tingkat resiko yang dapat diterima.

Ada beberapa cara yang dapat digunakan untuk melakukan perkiraan proyek:

1. Perkiraan keterlambatan suatu proyek dapat diselesaikan
2. Perkiraan berdasarkan proyek serupa yang telah selesai dikerjakan
3. Menggunakan teknik dekomposisi sederhana untuk membuat suatu kisaran biaya dan tenaga
4. Menggunakan satu atau lebih model empiris untuk perkiraan biaya dan tenaga

Teknik yang disarankan oleh para ahli di bidang perangkat lunak adalah dengan menggunakan kombinasi antara teknik dekomposisi dan model empiris.

Teknik dekomposisi menggunakan pendekatan dengan cara membagi suatu proyek ke dalam beberapa fungsi mayor yang berhubungan dengan aktivitas rekayasa perangkat lunak. Model estimasi empiris dapat digunakan sebagai pelengkap bagi teknik dekomposisi.

2.3.1 Estimasi Berbasis Masalah

Estimasi berbasis LOC dan FP memiliki teknik yang unik. Keduanya memiliki karakteristik sendiri. Perencana proyek memulai dengan kumpulan pernyataan yang berisi kerangka dan batasan-batasan dari perangkat lunak dan dari pernyataan-pernyataan tersebut kemudian mencoba untuk melakukan dekomposisi perangkat lunak ke dalam banyak fungsi permasalahan (*problem function*) dan melakukan estimasi variabel-variabel (LOC dan FP) pada tiap fungsi permasalahan. Sebagai alternatif, perencana proyek dapat memilih

komponen untuk menentukan ukuran, seperti kelas obyek, perubahan, atau proses bisnis yang terpengaruh.

Metrik produktivitas (misalnya: LOC/pm atau FP/pm⁹) diaplikasikan ke dalam variabel estimasi. Akronim *pm* digunakan untuk menyatakan satuan *person-month* (orang-bulan). Metrik kemudian dikombinasikan untuk memperoleh estimasi dari keseluruhan proyek. Dengan menggunakan data dari proyek di masa lalu, perencana proyek dapat membagi nilai perkiraan ke dalam nilai ukuran (S), yang dihasilkan dari rata-rata antara keadaan optimis (S_{opt}), mendekati (S_m), dan pesimis (S_{pe}), seperti contoh dalam persamaan (2-2) berikut ini:

$$S = (S_{opt} + 4S_m + S_{pe}) / 6 \quad (2-2)$$

2.3.2 Estimasi Berbasis Proses

Metode yang paling sering digunakan untuk melakukan estimasi pada suatu proyek adalah dengan melakukan perkiraan pada proses yang akan digunakan. Dari proses yang ada dilakukan dekomposisi sehingga diperoleh proses-proses yang relatif lebih kecil dan tenaga yang diperlukan untuk menyelesaikan tiap proses tersebut.

Seperti pada estimasi berbasis masalah, estimasi berbasis proses dimulai dengan menganalisa fungsi dan cara kerja perangkat lunak. Suatu seri dari aktivitas proses perangkat lunak harus dikerjakan pada tiap proses.

Setelah fungsi dan aktivitas proses dapat diperoleh, maka perencana proyek memperkirakan tenaga yang diperlukan untuk menyelesaikan suatu tahapan proses pada tiap fungsi perangkat lunak. Biaya dan tenaga untuk tiap

fungsi kemudian dijumlahkan, dan diperoleh estimasi biaya dan tenaga keseluruhan untuk menyelesaikan proyek perangkat lunak.

2.4 Model Estimasi Empiris

Model estimasi perangkat lunak menggunakan formula yang diperoleh secara empiris untuk memperkirakan tenaga sebagai sebuah fungsi dari LOC atau FP. Data empiris yang paling banyak mendukung dalam model-model estimasi diperoleh dari sampel proyek yang jumlahnya terbatas. Dengan alasan ini, maka tidak ada model estimasi yang cocok untuk semua lingkungan pengembangan perangkat lunak. Maka dari itu, penerapan hasil yang diperoleh dari model-model yang sudah disediakan harus digunakan secara bijaksana sesuai dengan keadaan lingkungan masing-masing pengembang.

2.4.1 Struktur Model-Model Estimasi

Model estimasi diperoleh melalui analisis regresi pada data yang diperoleh pada proyek-proyek di masa lalu. Keseluruhan struktur dari beberapa model mengambil dari persamaan yang dipaparkan oleh Matson [1994]:

$$E = A + B \times (ev)^C \quad (2-3)$$

Dimana E adalah *effort* dalam orang-bulan, dan A, B, C adalah konstanta yang diperoleh secara empiris. Dalam perkembangannya persamaan tersebut dikembangkan sesuai dengan proyek yang dikerjakan oleh masing-masing pengembang perangkat lunak. Perbedaan persamaan yang digunakan oleh masing-masing pengembang dikarenakan persamaan tersebut harus dikalibrasi terlebih dahulu, sesuai dengan kondisi maupun kebutuhan.

2.4.2 Model COCOMO

COCOMO adalah sebuah model yang didesain oleh Barry Boehm untuk memperoleh perkiraan dari jumlah orang-bulan yang diperlukan untuk mengembangkan suatu produk perangkat lunak. Satu hasil observasi yang paling penting dalam model ini adalah bahwa motivasi dari tiap orang yang terlibat ditempatkan sebagai titik berat. Hal ini menunjukkan bahwa kepemimpinan dan kerja sama tim merupakan sesuatu yang penting, namun demikian poin pada bagian ini sering diabaikan.

Model COCOMO dapat diaplikasikan dalam tiga tingkatan kelas:

1. Proyek organik, adalah proyek dengan ukuran relatif kecil, dengan anggota tim yang sudah berpengalaman, dan mampu bekerja pada permintaan yang relatif fleksibel
2. Proyek sedang (semi-terpisah), adalah proyek yang memiliki ukuran dan tingkat kerumitan yang sedang, dan tiap anggota tim memiliki tingkat keahlian yang berbeda
3. Proyek terintegrasi, adalah proyek yang dibangun dengan spesifikasi dan operasi yang ketat

Persamaan dasar model COCOMO adalah:

$$E = a_b (\mathbf{KLOC})^b_b \quad (2-4)$$

$$D = c_b (E)^d_b \quad (2-5)$$

$$P = E / D \quad (2-6)$$

Dimana **E** adalah usaha dalam orang-bulan, **D** adalah waktu pengerjaan dalam satuan bulan, **KLOC** adalah estimasi jumlah baris kode dalam ribuan, dan

P adalah jumlah orang yang diperlukan. Koefisien a_b , b_b , c_b , dan d_b diberikan pada tabel berikut:

Tabel 2.1 Koefisien model COCOMO dasar

Proyek Perangkat Lunak	a_b	b_b	c_b	d_b
Organik	2.4	1.05	2.5	0.38
Sedang	3.0	1.12	2.5	0.35
Terintegrasi	3.6	1.20	2.5	0.32

Pengembangan model COCOMO adalah dengan menambahkan atribut yang dapat menentukan jumlah biaya dan tenaga dalam pengembangan perangkat lunak, yang dijabarkan dalam kategori dan subkategori sebagai berikut:

1. Atribut produk
 - a. Reliabilitas perangkat lunak yang diperlukan
 - b. Ukuran basis data aplikasi
 - c. Kompleksitas produk
2. Atribut perangkat keras
 - a. Performa program ketika dijalankan
 - b. Memori yang dipakai
 - c. Stabilitas mesin virtual
 - d. Waktu yang diperlukan untuk mengeksekusi perintah
3. Atribut Sumber Daya Manusia
 - a. Kemampuan analisis
 - b. Kemampuan ahli perangkat lunak

- c. Pengalaman membuat aplikasi
 - d. Pengalaman menggunakan mesin virtual
 - e. Pengalaman dalam menggunakan bahasa pemrograman
4. Atribut proyek
- a. Menggunakan perangkat lunak tambahan
 - b. Metode rekayasa perangkat lunak
 - c. Waktu yang diperlukan

Masing-masing subkatergori diberi bobot antara 0 (sangat rendah) sampai 6 (sangat tinggi), dan kemudian dijumlahkan. Dari pengembangan ini diperoleh persamaan:

$$E = a_i (KLOC)^{b_i} \cdot EAF \quad (2-7)$$

Dimana E adalah usaha dalam satuan orang-bulan, KLOC adalah estimasi jumlah baris kode dalam ribuan, dan EAF adalah faktor hasil penghitungan dari subkategori di atas. Koefisien a_i dan eksponen b_i diberikan pada tabel berikut.

Tabel 2.2 Koefisien model COCOMO lanjut

Proyek Perangkat Lunak	a_i	b_i
Organik	3.2	1.05
Sedang	3.0	1.12
Terintegrasi	2.8	1.20

2.4.3 Persamaan pada Perangkat Lunak

Persamaan perangkat lunak merupakan model variabel jamak yang menghitung suatu distribusi spesifik dari usaha pada jalannya pengembangan

perangkat lunak. Persamaan berikut ini diperoleh dari hasil pengamatan terhadap lebih dari 4000 proyek perangkat lunak:

$$E = [\text{LOC} \times B^{0.333}/P]^3 \times (1/t^4) \quad (2-8)$$

E = usaha dalam orang-bulan atau orang-tahun

t = durasi proyek dalam bulan atau tahun

B = faktor kemampuan khusus

P = parameter produktivitas

Nilai B diambil dengan berdasarkan perkiraan. Untuk program berukuran kecil ($0.5 < \text{KLOC} < 5$), $B = 0.16$. Untuk program yang lebih besar dari 70 KLOC, $B = 0.39$.

Nilai P merefleksikan:

1. Kematangan proses dan praktek manajemen
2. Kualitas rekayasa perangkat lunak
3. Tingkat bahasa pemrograman yang digunakan
4. Keadaan lingkungan perangkat lunak
5. Kemampuan dan pengalaman tim pengembang
6. Kompleksitas aplikasi

Berdasarkan teori, diperoleh $P = 2000$ untuk sistem terapan, $P = 10000$ untuk perangkat lunak pada sistem informasi dan sistem telekomunikasi, dan $P = 28000$ untuk sistem aplikasi bisnis.

2.5 Konversi Waktu Tenaga Kerja

Konversi waktu tenaga kerja pada tugas akhir ini ini diperoleh dari angka pembandingan yang digunakan pada perangkat lunak ConvertAll, dengan hubungan

persamaan antara orang-bulan (OB), orang-jam (OJ), orang-minggu (OM), dan orang-tahun (OT) adalah sebagai berikut:

$$OM = 40 OJ$$

$$OT = 12 OB$$

$$OT = 52 OM$$

Dari persamaan di atas, diperoleh konversi orang-bulan ke orang-jam sebagai berikut:

$$OB = (40 OJ \times 52) / 12$$

$$\mathbf{OB = 173,33 OJ} \quad (2-9)$$

BAB III DESAIN SISTEM

3.1 Deskripsi Umum

Dalam studi ini, yang digunakan sebagai acuan untuk melakukan estimasi pada proyek perangkat lunak adalah spesifikasi keperluan perangkat lunak dan spesifikasi desain perangkat lunak. Proyek yang digunakan sebagai bahan untuk keperluan studi kasus adalah kegiatan pembuatan *prototype* Sistem Informasi Manajemen Perencanaan Pemerintah Daerah Propinsi (SIMANCA) di CV Global Intermedia, Yogyakarta.

3.2 Spesifikasi Keperluan Perangkat Lunak

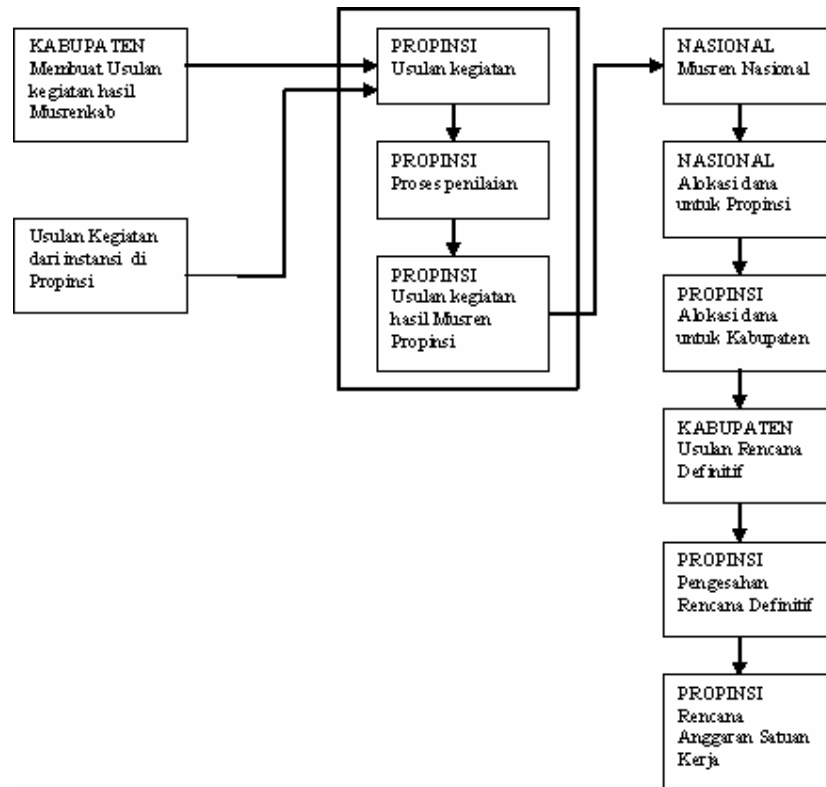
SIMANCA digunakan sebagai alat untuk melakukan otomatisasi pengolahan data pada proses perencanaan kegiatan di pemerintah daerah. Aturan perundangan yang digunakan sebagai acuan adalah Undang-Undang nomor 25 Tahun 2004 tentang Sistem Perencanaan Pembangunan Nasional dan Surat Edaran Bersama Menteri Negara Perencanaan Pembangunan Nasional dan Menteri Dalam Negeri nomor 0259/M.PPN/I/2005 dan nomor 050/166/SJ Tanggal 20 Januari 2005 tentang Petunjuk Teknis Pelaksanaan Musrenbang Tahun 2005.

3.2.1 Sistem yang Sudah Ada

Saat ini sistem yang digunakan sebagai acuan untuk pengembangan SIMANCA adalah sistem perencanaan yang dilakukan secara manual. Setelah dilakukan Musyawarah Perencanaan Pembangunan Kabupaten, dihasilkan usulan-

usulan kegiatan dari Kabupaten. Usulan-usulan ini digabungkan dengan usulan-usulan dari instansi di Propinsi sebagai bahan masukan untuk kegiatan Musyawarah Perencanaan Pembangunan Propinsi. Usulan-usulan tersebut kemudian diseleksi dengan melakukan penilaian dengan metode pembobotan, yang kemudian dikeluarkan sebagai dokumen untuk perencanaan di tingkat nasional.

Secara garis besar alur sistem perencanaan dipaparkan dalam gambar berikut:



Gambar 3.1 Alur Proses Perencanaan Daerah

Penggunaan SIMANCA diharapkan dapat mempermudah instansi terkait dalam proses masukan data dan proses penilaian. Sistem otomatis diharapkan

dapat membuat sistem dapat digunakan dalam waktu-nyata, sehingga laporan-laporan dapat segera diperoleh setelah proses pemasukan data selesai dilakukan.

3.2.2 Sasaran Pengguna

Sasaran pengguna SIMANCA adalah Badan Perencanaan Pemerintah Daerah, Tim Penilai Usulan Kegiatan (TPUK), dan instansi-instansi di Pemerintah Daerah, yang menggunakan antarmuka SIMANCA untuk memasukkan usulan kegiatan dan penilaian kegiatan. Karakteristik pengguna adalah pegawai unit kerja di pemerintah daerah, yang menguasai penggunaan komputer untuk melakukan akses ke web.

3.2.3 Sistem yang Diusulkan

Sistem yang diusulkan adalah sistem informasi dengan fasilitas untuk memasukkan usulan kegiatan, penilaian kegiatan, dan pencetakan hasil penilaian kegiatan. Hak akses yang diberikan adalah: administrator, operator, dan penilai. Administrator sistem memiliki fasilitas untuk memasukkan data-data tambahan yang diperlukan untuk proses pemasukan usulan kegiatan dan penilaian kegiatan, yang meliputi: data tahun anggaran, data bidang dan unit kerja, data pengguna, data penilai, dan parameter penilaian.

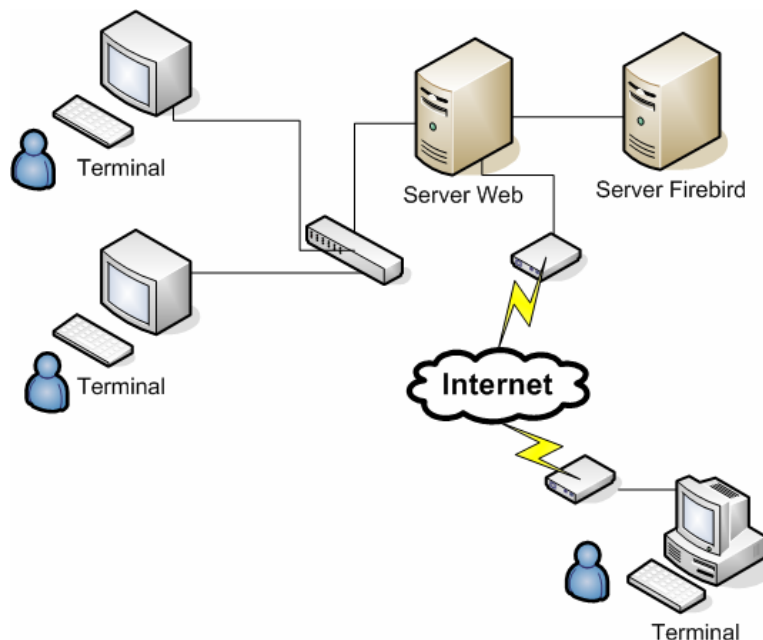
Sistem dirancang menggunakan antarmuka berbasis web, dengan akses ke basis data dalam waktu-nyata, sehingga kondisi data terbaru dapat diperoleh secara langsung pada saat sistem diakses oleh pengguna.

3.2.4 Lingkungan Perangkat Lunak

SIMANCA dijalankan pada lingkungan *server* dengan sistem operasi Linux, dengan antarmuka berbasis web, yang dapat diakses melalui jaringan komputer. Server web yang digunakan adalah Apache PHP dan basis data yang digunakan adalah Firebird.

3.2.5 Lingkungan Perangkat Keras dan Jaringan Komputer

Perangkat keras yang diperlukan sebagai pusat penyimpanan data SIMANCA adalah komputer yang mampu berfungsi sebagai server web dan server basis data Firebird, yang dapat diakses oleh pengguna melalui jaringan komputer intranet maupun internet. Skema jaringan yang diusulkan adalah sebagai berikut:



Gambar 3.2 Usulan Jaringan Komputer

Perangkat yang digunakan untuk mengakses SIMANCA adalah komputer atau peralatan yang bisa terhubung ke server SIMANCA melalui intranet atau internet, yang memiliki fasilitas penjelajah web yang mampu menampilkan HTML dan mengeksekusi javascript.

3.2.6 Spesifikasi Keperluan Fungsional

SIMANCA dibagi dalam fungsi-fungsi spesifik, yang dijabarkan dalam fungsi-fungsi sebagai berikut:

Fungsi Login

Fungsi login digunakan untuk autentikasi pengguna SIMANCA dengan memasukkan nama pengguna dan sandi yang telah ditentukan. Nama pengguna dan sandi digunakan untuk menjaga keamanan sistem dan mengatur hak akses yang berbeda untuk tiap pengguna.

Fungsi Pengubahan Sandi

Fungsi pengubahan sandi (*password*) digunakan untuk melakukan pengubahan sandi oleh masing-masing pengguna. Masukan dari sistem ini adalah sandi lama untuk autentikasi pengguna dan sandi baru. Fungsi ini merupakan salah satu fasilitas yang digunakan untuk keperluan keamanan sistem. Fungsi ini dapat diakses oleh semua pengguna yang telah masuk ke SIMANCA.

Fungsi Administrasi Data Tahun Anggaran

Tahun anggaran adalah tahun yang berlaku pada pelaksanaan kegiatan yang diusulkan. Tahun digunakan sebagai parameter dasar untuk pengisian data-data oleh pengguna SIMANCA. Administrasi data tahun anggaran terdiri dari:

fungsi menampilkan data tahun, menambah data tahun, dan menghapus data tahun. Fungsi setting tahun hanya dapat dijalankan oleh administrator.

Fungsi Administrasi Data Bidang

Fungsi administrasi data bidang digunakan untuk data bidang dari himpunan unit kerja yang ada di pemerintah daerah. Data bidang tergantung dengan data tahun, sehingga untuk tahun anggaran yang berbeda bisa diperoleh data bidang yang berbeda, tergantung aturan yang berlaku pada tahun anggaran tersebut. Administrasi data bidang terdiri dari: fungsi menampilkan data bidang, menambah data bidang, mengubah data bidang, dan menghapus data bidang. Fungsi administrasi data bidang hanya dapat dijalankan oleh administrator.

Fungsi Administrasi Data Unit Kerja

Data unit kerja berada di bawah bidang dan tahun anggaran. Satu bidang membawahi satu atau lebih unit kerja di dalam pemerintahan. Administrasi data unit kerja terdiri dari: menampilkan data unit kerja, menambah data unit kerja, mengubah data unit kerja, dan menghapus data unit kerja. Fungsi administrasi data unit kerja hanya dapat dijalankan oleh administrator.

Fungsi Administrasi Data Penilai

Penilai adalah suatu kelompok atau lembaga yang memiliki hak untuk menentukan besarnya bobot pada suatu usulan kegiatan. Data penilai tergantung pada data tahun anggaran. Fungsi administrasi data penilai meliputi: menampilkan data penilai, menambah data penilai, mengubah data penilai, dan menghapus data penilai. Fungsi administrasi data penilai hanya dapat dijalankan oleh administrator.

Fungsi Administrasi Data Kriteria Penilaian

Kriteria penilaian digunakan untuk menentukan parameter-parameter kondisi yang digunakan sebagai acuan penilaian. Kriteria penilaian tergantung pada data tahun anggaran. Fungsi administrasi data penilaian terdiri dari: menampilkan data kriteria, menambah data kriteria, mengubah data kriteria, dan menghapus data kriteria. Fungsi administrasi data kriteria hanya dapat diakses oleh administrator.

Fungsi Administrasi Data Subkriteria Penilaian

Subkriteria penilaian digunakan untuk menentukan parameter-parameter kondisi yang lebih spesifik dari kriteria untuk digunakan sebagai acuan penilaian. Data subkriteria tergantung pada data tahun anggaran dan data kriteria. Fungsi administrasi data penilaian terdiri dari: menampilkan data subkriteria, menambah data subkriteria, mengubah data subkriteria, dan menghapus data subkriteria. Fungsi subkriteria hanya dapat diakses oleh administrator.

Fungsi Administrasi Data Pengguna

Pengguna adalah unit kerja yang diberikan hak untuk masuk dan menggunakan SIMANCA. Akses yang diberikan kepada masing-masing pengguna berbeda tergantung pada data tahun anggaran, data bidang, dan data unit kerja. Hak akses yang diberikan adalah: administrator, operator, dan penilai. Fungsi administrasi data pengguna meliputi: menampilkan data pengguna, menambah data pengguna, mengubah data pengguna, dan menghapus data pengguna. Fungsi administrasi data pengguna hanya dapat diakses oleh administrator.

Fungsi Administrasi Data Program

Fungsi administrasi data program digunakan untuk menentukan program-program yang berjalan pada tahun anggaran tertentu. Data program tergantung pada data tahun anggaran, bidang kerja, dan unit kerja. Fungsi administrasi data program terdiri dari: menampilkan data program, menambah data program, mengubah data program, dan menghapus data program. Fungsi ini dapat diakses oleh administrator dan operator.

Fungsi Administrasi Data Usulan Kegiatan

Fungsi administrasi data usulan kegiatan digunakan untuk menentukan data-data usulan kegiatan yang telah dirumuskan. Data usulan kegiatan tergantung pada data tahun anggaran, bidang, unit kerja, dan data program. Fungsi administrasi data usulan kegiatan terdiri dari: menampilkan data usulan kegiatan, menambah data usulan kegiatan, mengubah data usulan kegiatan, dan menghapus data usulan kegiatan. Fungsi ini dapat diakses oleh administrator dan operator.

Fungsi Penilaian

Fungsi penilaian digunakan sebagai alat untuk melakukan penilaian. Data penilaian tergantung pada: data tahun anggaran, bidang, unit kerja, program, kegiatan, kriteria, dan sub kriteria. Fungsi penilaian terdiri dari menampilkan data penilaian dan mengubah data penilaian. Penilaian dilakukan pada kegiatan untuk tiap-tiap subkriteria. Fungsi penilaian hanya dapat diakses oleh penilai.

Fungsi Pelaporan

Fungsi pelaporan digunakan untuk menampilkan laporan-laporan seperti: laporan hasil penilaian, laporan usulan kegiatan, dan laporan usulan kegiatan yang

diterima atau ditolak. Fungsi pelaporan dapat diakses oleh administrator, operator, dan penilai.

Fungsi Transfer Data

Fungsi transfer data digunakan untuk mengeluarkan data yang dapat digunakan sebagai masukan untuk aplikasi Rencana Anggaran Satuan Kerja (RASK). Data yang dihasilkan berupa teks yang telah disepakati dengan pengembang Sistem Informasi RASK.

3.2.7 Dokumentasi Sistem

Dokumentasi yang direncanakan untuk dibuat adalah dokumentasi yang berisi petunjuk penggunaan, yang ditujukan untuk pengguna SIMANCA. Dokumentasi memuat informasi sekilas tentang perencanaan daerah, petunjuk instalasi, dan petunjuk pengoperasian SIMANCA.

3.3 Spesifikasi Desain Perangkat Lunak

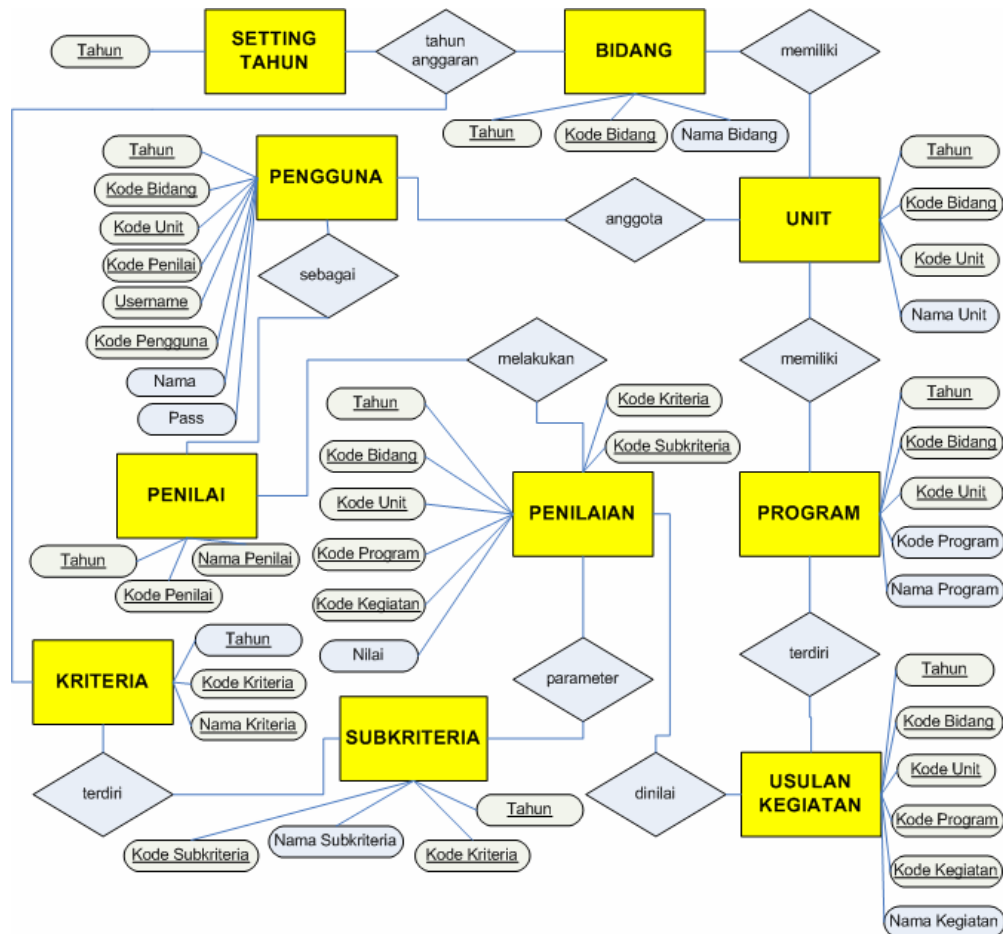
Spesifikasi desain yang terdapat dalam SIMANCA digunakan sebagai acuan untuk manajer proyek dan tim pengembang untuk diterapkan pada proses pengembangan SIMANCA.

3.3.1 Desain Data

Data SIMANCA disimpan dalam sebuah basis data Firebird. Database dibuat untuk memenuhi kebutuhan administrasi data yang dipaparkan dalam spesifikasi keperluan perangkat lunak. Rancangan basis data dibuat dengan mengumpulkan entitas-entitas yang berhubungan dengan SIMANCA, dan menghubungkannya dengan mengacu pada relasi antar entitas-entitas tersebut.

Entitas-entitas yang dimaksud adalah: tahun, bidang, unit kerja, program, usulan kegiatan, kriteria, subkriteria, penilai, penilaian, dan pengguna.

Diagram *Entity-Relationship* (ER) dari basis data SIMANCA dipaparkan dalam gambar 3.3 berikut:

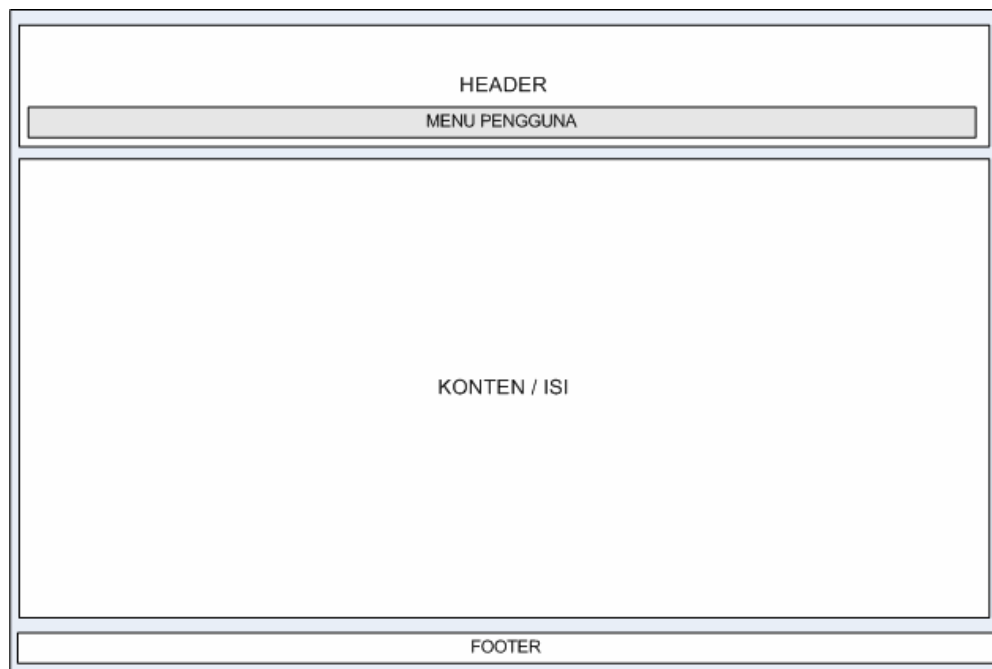


Gambar 3.3 Diagram E-R rancangan basis data

3.3.2 Rancangan Antarmuka

Antarmuka dirancang memiliki 4 (empat) bagian, yaitu: *header*, menu, isi, dan *footer*. *Header* berisi judul dan informasi pengguna SIMANCA. Menu berada di dalam *header*, digunakan sebagai pranala untuk mengakses modul-modul yang ada di SIMANCA. Bagian isi digunakan sebagai tempat untuk meletakkan modul

yang memuat fungsi-fungsi yang dikehendaki dalam SIMANCA. Rancangan bagian pada antarmuka SIMANCA yang dibuat menggunakan perangkat lunak Microsoft Visio 2003 dipaparkan dalam gambar 3.4 berikut:



Gambar 3.4 Perancangan antarmuka modul

Standarisasi rancangan antarmuka dibagi ke dalam 3 (tiga) bagian, yaitu: menampilkan data dan menambah data, dan mengubah data. Menu penghapusan dan penambahan data data ditempatkan sebagai pranala pada halaman tampilan data.

Data ditampilkan pada bagian isi. Di dalam bagian isi terdapat sub bagian, yaitu sub bagian tapis data dan sub bagian tampilan data. Sub bagian tapis data berisi borang yang digunakan untuk melakukan penapisan (*filtering*) data berdasarkan parameter-parameter tertentu yang dijelaskan pada bagian rancangan

komponen. Rancangan antarmuka tampilan data dijabarkan dalam gambar 3.5 berikut ini:



Gambar 3.5 Rancangan antarmuka tampilan data

Rancangan antarmuka untuk menambah data memiliki kemiripan dengan antarmuka pada tampilan untuk mengubah data, karena borang yang menjadi alat pemasukan data memiliki struktur desain yang sama. Perbedaan yang terdapat pada kedua macam borang tersebut adalah borang penambahan berisi data kosong, sedangkan pada borang pengubahan data bersisi data-data yang hendak diubah. Pranala menuju borang penambahan dan pengubahan data diletakkan pada sub bagian tampilan data di borang tampilan data.

Borang penambahan dan pengubahan data dirancang diletakkan di bagian isi. Rancangan antarmuka untuk menambah atau mengubah data terdapat pada gambar sebagai berikut:

The image shows a web form layout. At the top is a header bar labeled 'HEADER' containing a 'MENU PENGGUNA' button. Below this is a main content area with three input fields: 'Isian #1' (text), 'Isian #2' (text), and 'Isian #3' (dropdown menu). At the bottom of the main area are two buttons: 'Masukkan Data' and 'Batal'. At the very bottom is a footer bar labeled 'FOOTER'.

Gambar 3.6 Rancangan tampilan pengisian data

3.3.3 Rancangan Menu

Menu ditampilkan ketika pengguna berhasil masuk ke sistem setelah memasukkan pengguna dan sandi dengan benar. Menu yang ditampilkan berbeda untuk tiap tingkatan pengguna. Pengguna diberikan menu sesuai dengan hak akses yang dimiliki.

Menu Umum

Yang dimaksud dengan menu umum adalah menu yang menjadi pranala ke modul yang bisa diakses oleh semua tingkatan pengguna. Modul-modul yang bisa diakses oleh semua tingkatan pengguna antara lain: modul ubah sandi dan modul pencetakan laporan.

Menu Administrator

Modul-modul yang dapat diakses oleh administrator antara lain: setting tahun anggaran, setting bidang, setting unit kerja, setting program, setting usulan kegiatan, setting kriteria, setting subkriteria, dan setting penilai.

Menu Operator

Modul-modul yang dapat diakses oleh operator antara lain: setting program dan setting kegiatan.

Menu Penilai

Modul yang dapat diakses oleh administrator adalah modul penilaian usulan kegiatan.

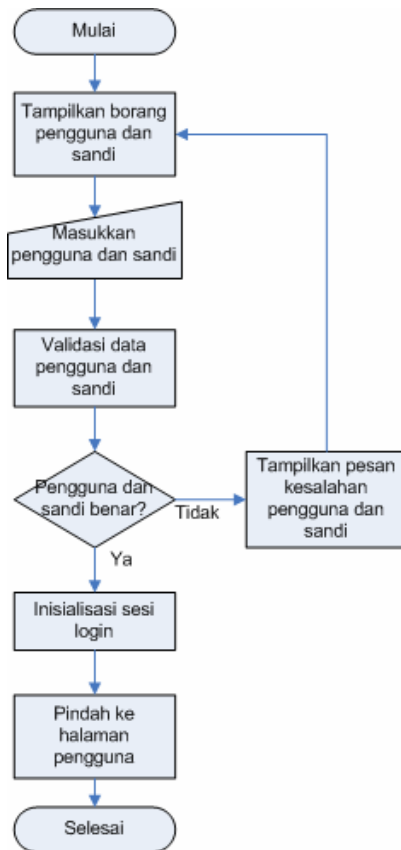
3.3.4 Rancangan Komponen

Rancangan komponen berisi tentang deskripsi dan cara kerja tiap modul pada SIMANCA. Rancangan komponen dibedakan dengan spesifikasi keperluan fungsional.

Login

Identifikasi	Berkas index.php pada <i>folder</i> utama
Fungsi	Autentikasi untuk masuk ke dalam sistem
Hak Akses	Semua pengguna
Antarmuka	Tampilan borang nama pengguna dan sandi
Pemrosesan	Validasi pengguna dan sandi yang dimasukkan

Diagram alir proses login terdapat pada gambar 3.7.

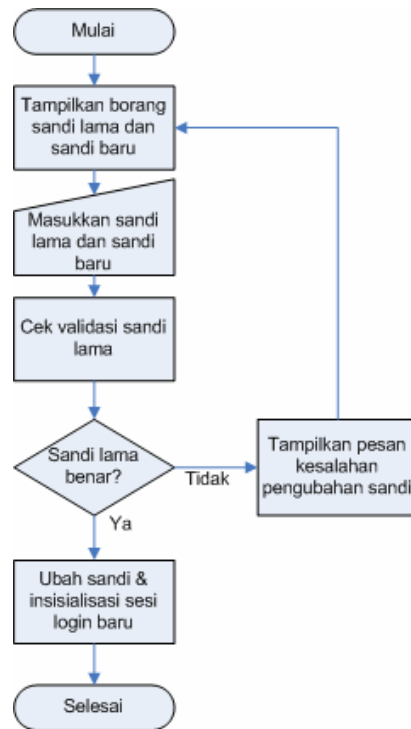


Gambar 3.7 Diagram alir proses login

Pengubahan Sandi

Identifikasi	Folder password
Fungsi	Pengubahan sandi
Hak Akses	Semua pengguna
Antarmuka	Tampilan borang pengubahan sandi
Pemrosesan	Validasi dan penggantian sandi lama dengan sandi baru

Diagram alir proses pengubahan sandi terdapat pada gambar 3.8.

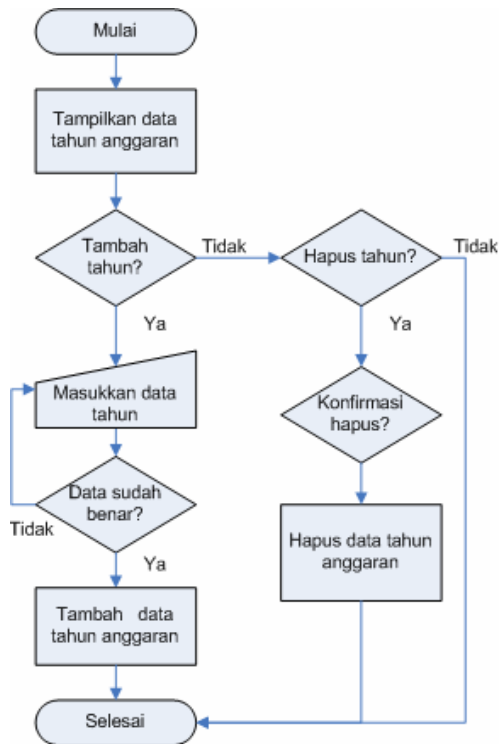


Gambar 3.8 Diagram alir proses pengubahan sandi

Administrasi Data Tahun Anggaran

Identifikasi	<i>Folder setting_tahun</i>
Fungsi	Administrasi data tahun anggaran
Hak Akses	Administrator
Antarmuka	<ul style="list-style-type: none"> • Tampilan data tahun • Tampilan masukan data tahun anggaran
Pemrosesan	<ul style="list-style-type: none"> • Penambahan dan penghapusan data tahun anggaran

Diagram alir proses administrasi data tahun anggaran terdapat pada gambar 3.9.

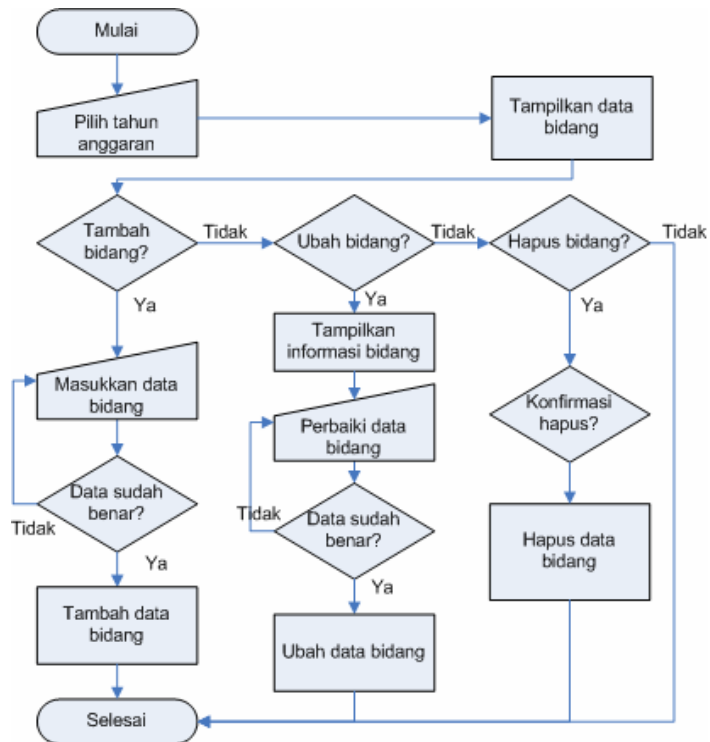


Gambar 3.9 Diagram alir proses administrasi data tahun anggaran

Administrasi Data Bidang

Identifikasi	<i>Folder setting_bidang</i>
Fungsi	Administrasi data bidang
Hak Akses	Administrator
Antarmuka	<ul style="list-style-type: none"> • Tampilan data bidang • Tampilan masukan dan perubahan data bidang
Pemrosesan	<ul style="list-style-type: none"> • Penambahan, perubahan, dan penghapusan data bidang

Diagram alir proses administrasi data bidang terdapat pada gambar 3.10.



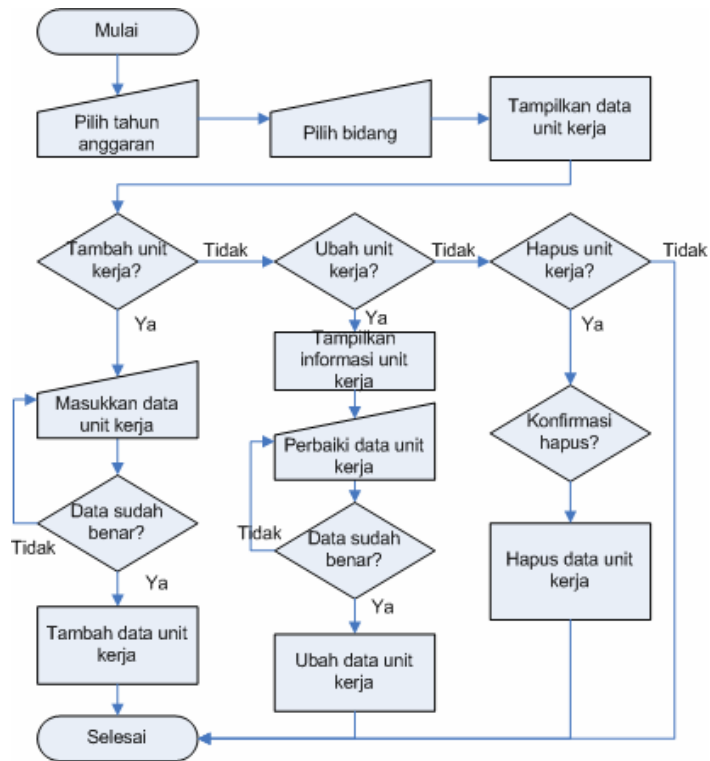
Gambar 3.10 Diagram alir proses administrasi data bidang

Administrasi Data Unit Kerja

Identifikasi	<i>Folder setting_unit</i>
Fungsi	Administrasi data unit kerja
Hak Akses	Administrator
Antarmuka	<ul style="list-style-type: none"> • Tampilan data unit kerja • Tampilan masukan dan perubahan data unit kerja
Pemrosesan	<ul style="list-style-type: none"> • Penambahan, perubahan, dan penghapusan data unit kerja

Diagram alir proses administrasi data unit kerja terdapat pada gambar

3.11.



Gambar 3.11 Diagram alir proses administrasi data unit kerja

Administrasi Data Penilai

Identifikasi	<i>Folder setting_penilai</i>
Fungsi	Administrasi data penilai usulan kegiatan
Hak Akses	Administrator
Antarmuka	<ul style="list-style-type: none"> • Tampilan data penilai • Tampilan masukan dan pengubahan data penilai
Pemrosesan	<ul style="list-style-type: none"> • Penambahan, pengubahan, dan penghapusan data penilai

Diagram alir proses administrasi data penilai terdapat pada gambar 3.12.

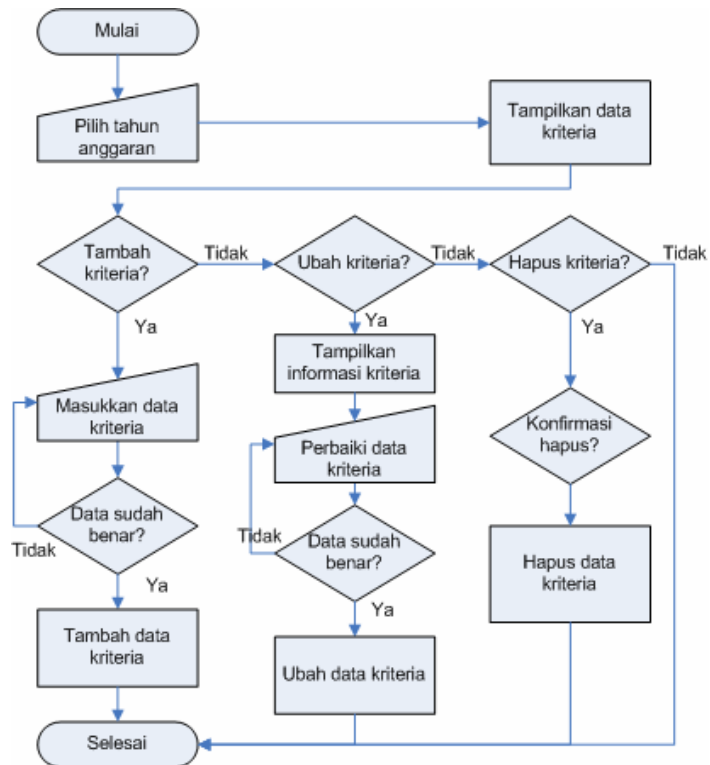


Gambar 3.12 Diagram alir proses administrasi data penilai

Administrasi Data Kriteria Penilaian

Identifikasi	<i>Folder setting_kriteria</i>
Fungsi	Administrasi data kriteria penilaian
Hak Akses	Administrator
Antarmuka	<ul style="list-style-type: none"> Tampilan data kriteria penilaian Tampilan masukan dan perubahan data kriteria penilaian
Pemrosesan	<ul style="list-style-type: none"> Penambahan, perubahan, dan penghapusan data kriteria penilaian

Diagram alir proses administrasi data kriteria penilaian terdapat pada gambar 3.13.

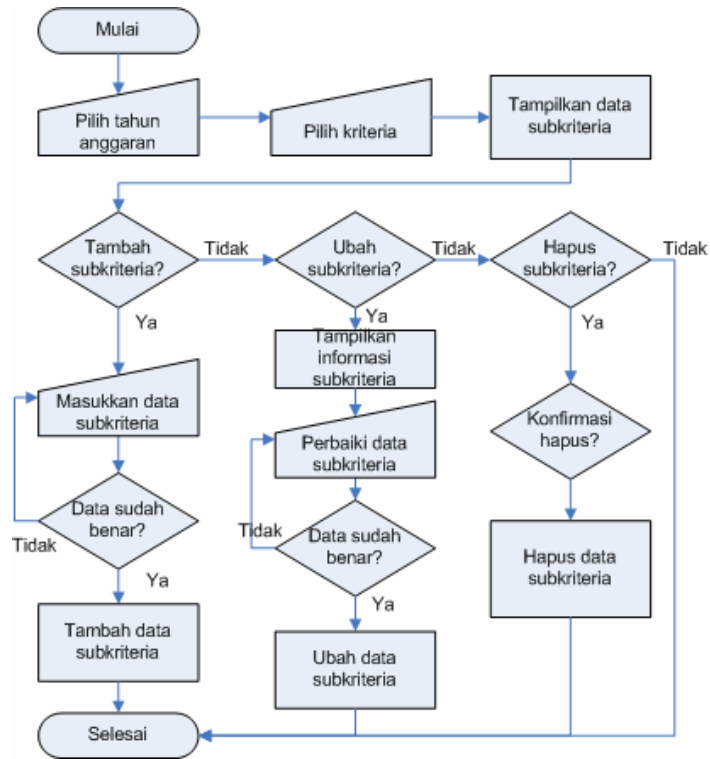


Gambar 3.13 Diagram alir proses administrasi data kriteria penilaian

Administrasi Data Subkriteria Penilaian

Identifikasi	<i>Folder setting_subkriteria</i>
Fungsi	Administrasi data subkriteria penilaian
Hak Akses	Administrator
Antarmuka	<ul style="list-style-type: none"> Tampilan data subkriteria penilaian Tampilan masukan dan perubahan data subkriteria penilaian
Pemrosesan	<ul style="list-style-type: none"> Penambahan, perubahan, dan penghapusan data subkriteria penilaian

Diagram alir proses administrasi data subkriteria penilaian terdapat pada gambar 3.14.



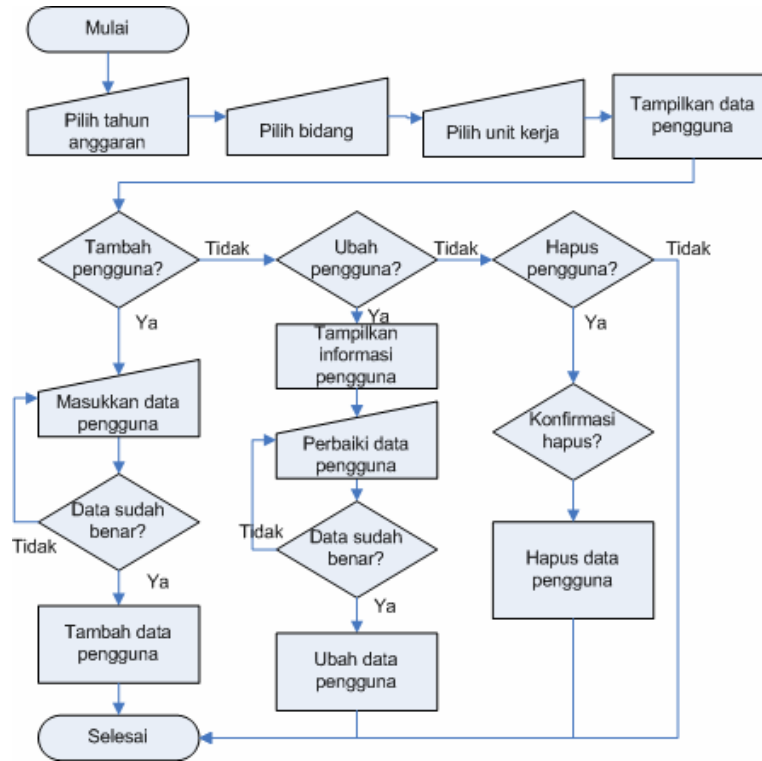
Gambar 3.14 Diagram alir proses administrasi data subkriteria penilaian

Administrasi Data Pengguna

Identifikasi	<i>Folder setting_pengguna</i>
Fungsi	Administrasi data pengguna SIMANCA
Hak Akses	Administrator
Antarmuka	<ul style="list-style-type: none"> • Tampilan data pengguna • Tampilan masukan dan pengubahan data pengguna
Pemrosesan	<ul style="list-style-type: none"> • Penambahan, pengubahan, dan penghapusan data pengguna

Diagram alir proses administrasi data pengguna terdapat pada gambar

3.15.

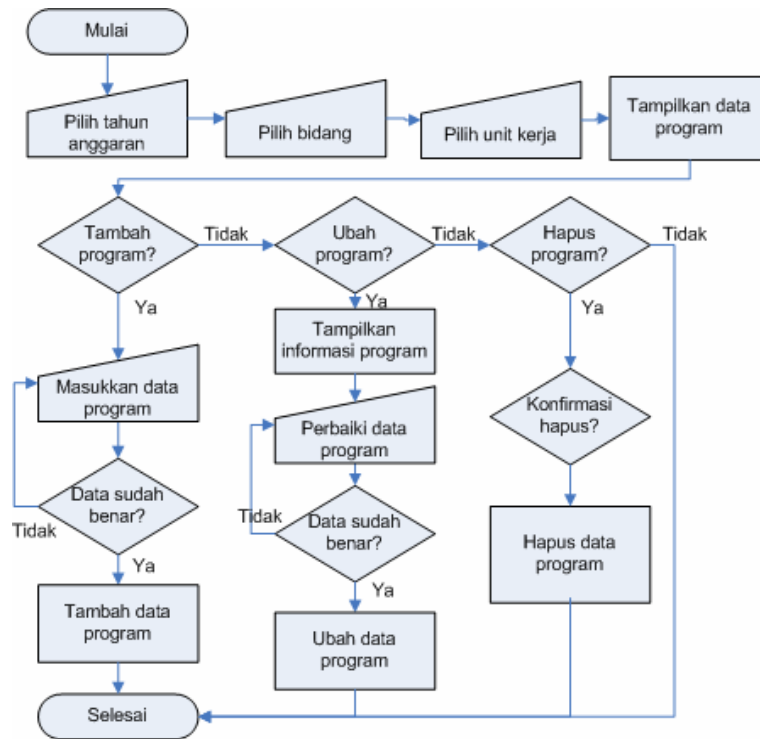


Gambar 3.15 Diagram alir proses administrasi data pengguna

Administrasi Data Program

Identifikasi	<i>Folder setting_program</i>
Fungsi	Administrasi data program
Hak Akses	Administrator, operator
Antarmuka	<ul style="list-style-type: none"> • Tampilan data program • Tampilan masukan dan perubahan data program
Pemrosesan	<ul style="list-style-type: none"> • Penambahan, perubahan, dan penghapusan data program

Diagram alir proses administrasi data program terdapat pada gambar 3.16.

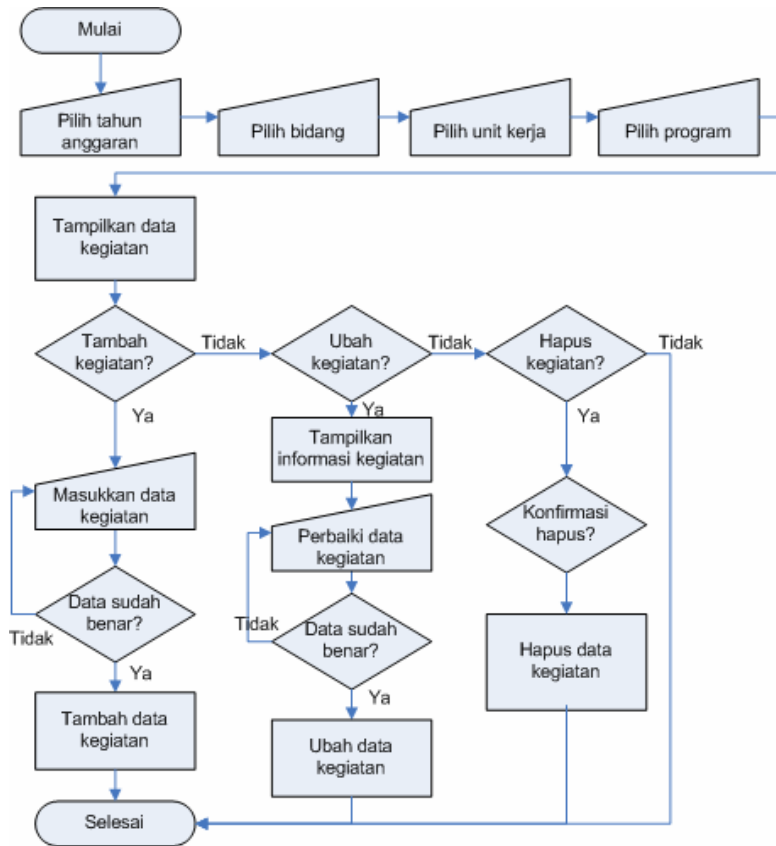


Gambar 3.16 Diagram alir proses administrasi data program

Administasi Data Usulan Kegiatan

Identifikasi	<i>Folder setting_kegiatan</i>
Fungsi	Administrasi data usulan kegiatan
Hak Akses	Administrator, operator
Antarmuka	<ul style="list-style-type: none"> • Tampilan data usulan kegiatan • Tampilan masukan dan pengubahan data usulan kegiatan
Pemrosesan	<ul style="list-style-type: none"> • Penambahan, pengubahan, dan penghapusan data usulan kegiatan

Diagram alir proses administrasi data usulan kegiatan terdapat pada gambar 3.17.



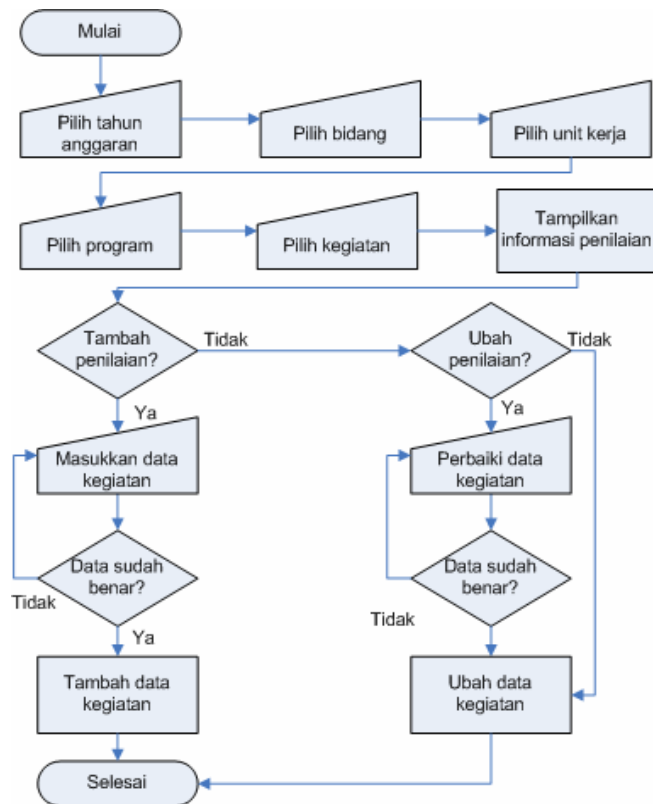
Gambar 3.17 Diagram alir proses administrasi data usulan kegiatan

Penilaian

Identifikasi	<i>Folder setting_penilaian</i>
Fungsi	Administrasi data penilaian usulan kegiatan
Hak Akses	Penilai
Antarmuka	<ul style="list-style-type: none"> • Tampilan data penilaian usulan kegiatan • Tampilan masukan dan perubahan data penilaian usulan kegiatan

Pemrosesan	<ul style="list-style-type: none"> • Penambahan, perubahan, dan penghapusan data penilaian usulan kegiatan
------------	---

Diagram alir proses penilaian terdapat pada gambar 3.18.



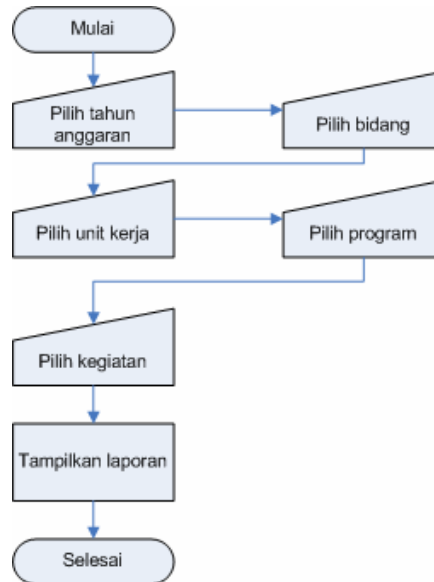
Gambar 3.18 Diagram alir proses penilaian

Pelaporan

Identifikasi	<i>Folder setting_penilaian</i>
Fungsi	Menampilkan laporan untuk SIMANCA
Hak Akses	Semua pengguna
Antarmuka	Tampilan data hasil penilaian kegiatan
Pemrosesan	Penampilan data berdasarkan kriteria-kriteria yang

	ditentukan oleh pengguna
--	--------------------------

Diagram alir proses pelaporan terdapat pada gambar 3.19.



Gambar 3.19 Diagram alir proses pelaporan

Transfer Data

Identifikasi	<i>Folder ekstrak_data</i>
Fungsi	Melakukan ekstrak data untuk dapat digunakan oleh aplikasi lain
Hak Akses	Administrator
Antarmuka	Tampilan teks hasil ekstrak data
Pemrosesan	Penampilan data berupa teks untuk di- <i>download</i>

Diagram alir proses transfer data terdapat pada gambar 3.20.

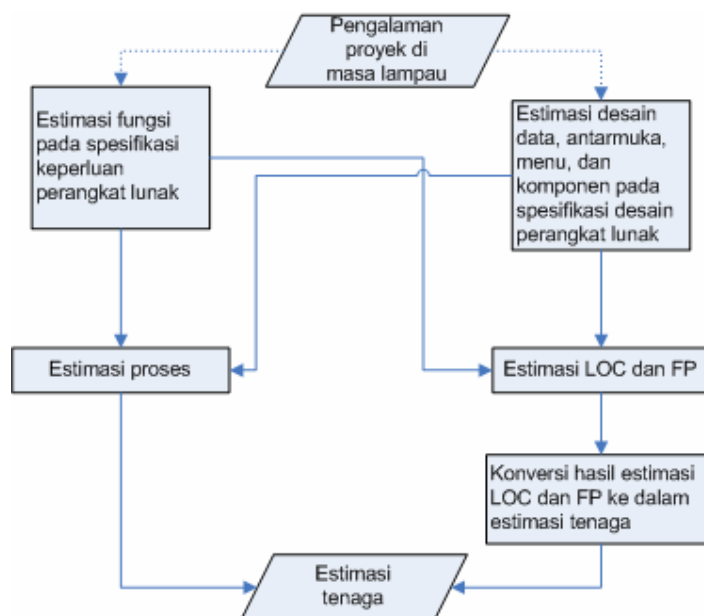


Gambar 3.20 Diagram alir proses transfer data

BAB IV IMPLEMENTASI DAN PEMBAHASAN

4.1 Petunjuk Umum Estimasi Perangkat Lunak

Berdasarkan pengalaman yang diperoleh dari proyek pengembangan SIMANCA, diperoleh diagram alur kerja estimasi proyek pengembangan perangkat lunak yang terdapat pada gambar 4.1.



Gambar 4.1 Alur kerja estimasi perangkat lunak

Estimasi dilakukan pada analisis fungsi dan desain yang diperoleh dari spesifikasi keperluan perangkat lunak dan spesifikasi desain perangkat lunak. Estimasi ini lebih mengandalkan intuisi yang dikombinasikan dengan pengalaman proyek di masa lampau sebagai bahan pertimbangan dalam melakukan estimasi. Estimasi fungsi diperoleh dari spesifikasi keperluan perangkat lunak, sedangkan estimasi desain data, antarmuka, menu, dan komponen diperoleh dari spesifikasi

desain perangkat lunak. Estimasi fungsi dan desain digunakan sebagai acuan untuk estimasi proses maupun estimasi LOC dan FP.

Pada estimasi berbasis proses, dilakukan estimasi pada waktu yang diperlukan untuk melakukan analisa tiap fungsi, desain, pengerjaan (*coding*), dan ujicoba (*testing*). Seluruh estimasi usaha dan tenaga yang diperlukan untuk tiap fungsi dan desain dijumlahkan, sehingga langsung dapat diketahui estimasi jumlah usaha dan tenaga yang diperlukan untuk menyelesaikan proyek pengembangan perangkat lunak.

Selain estimasi berbasis proses, terdapat estimasi berbasis LOC dan FP. Estimasi LOC dan FP juga menggunakan pengalaman proyek di masa lalu sebagai salah satu bahan pertimbangan dalam estimasi. Pada estimasi berbasis LOC, Sistem dibagi kedalam modul-modul yang sesuai, dan modul dibagi ke dalam fungsi, lalu dilakukan estimasi jumlah LOC untuk tiap fungsi di dalam modul tersebut. Estimasi jumlah LOC untuk tiap fungsi kemudian dijumlahkan sehingga diperoleh estimasi LOC pada tiap modul. Estimasi LOC pada tiap modul kemudian dijumlahkan sehingga diperoleh estimasi jumlah LOC perangkat lunak secara keseluruhan.

Estimasi FP dilakukan berdasarkan jumlah masukan, jumlah keluaran, jumlah permintaan (*inquiries*), jumlah berkas, dan jumlah antarmuka eksternal. Masing-masing jumlah estimasi tersebut dikalikan dengan faktor pemberat berdasarkan kompleksitas fungsi perangkat lunak, yang menghasilkan estimasi FP untuk tiap komponen. Estimasi FP tersebut dijumlahkan untuk memperoleh

estimasi FP total. Estimasi FP total kemudian dikalikan dengan faktor peubah kompleksitas untuk memperoleh estimasi FP akhir.

Hasil estimasi LOC dan FP kemudian dimasukkan ke dalam persamaan-persamaan yang sudah ditentukan, untuk mengubah estimasi LOC dan FP menjadi estimasi tenaga kerja dan usaha yang diperlukan untuk menyelesaikan proyek pengembangan perangkat lunak.

4.2 Implementasi Berdasarkan Teori

Implementasi berdasarkan teori dilakukan dengan menggunakan sumber tulisan mengenai metrik dan pengukuran perangkat lunak. Teori yang digunakan untuk melakukan perhitungan mengacu pada Pressman [2000].

4.2.1 Estimasi Berbasis LOC

Untuk melakukan penghitungan LOC dilakukan dekomposisi dengan menggunakan persamaan (2-2). Dalam proyek SIMANCA diasumsikan beberapa fungsi perangkat lunak yang diidentifikasi, di antaranya adalah: antarmuka pengguna dan fasilitas kendali (UICF), manajemen basis data (DBM), dan modul analisis desain (DAM). Menurut Pressman, perkiraan LOC dapat dilakukan dengan menggunakan data historis dari proyek-proyek serupa di masa lalu, dan apabila data-data tersebut tidak ada, digunakan metode intuisi dengan memanfaatkan perkiraan mengenai nilai optimis, nilai mendekati (sedang), dan nilai pesimis, lalu diambil reratanya. Pada analisa LOC di dalam sub bab ini, digunakan analisa berdasarkan perkiraan data yang diperoleh dari beberapa proyek yang sudah ada sebelumnya.

Antarmuka pengguna dan fasilitas kendali (UICF)

Antarmuka pengguna dirancang dengan menggunakan tampilan berbasis web. Kendali fungsi dilakukan melalui pranala-pranala yang disediakan untuk tiap modul yang terdapat pada SIMANCA.

Tabel 4.1 Analisa estimasi LOC UICF

Modul	S_{opt}	S_m	S_{pess}	S
Login	20	25	30	25
Pengubahan sandi	20	25	30	25
Administrasi data tahun anggaran	60	70	80	70
Administrasi data bidang	70	85	100	85
Administrasi data unit kerja	75	90	105	90
Administrasi data penilai	70	85	100	85
Administrasi data kriteria penilaian	70	85	100	85
Administrasi data subkriteria penilaian	75	90	105	90
Administrasi data pengguna	80	100	120	100
Administrasi data program	70	85	100	85
Administrasi data usulan kegiatan	75	90	105	90
Penilaian	80	100	120	100
Pelaporan	125	160	200	161
Transfer data	20	30	40	30
Total Estimasi LOC_{UICF}				1121

Manajemen Basis Data (DBM)

Dalam estimasi desain basis data digunakan analisa berdasarkan rancangan diagram E-R yang terdapat pada spesifikasi desain perangkat lunak. Perkiraan dilakukan dengan menjumlahkan atribut dan entitas-entitas yang terlibat pada desain basis data. Perkiraan tersebut kemudian dijumlahkan dengan perkiraan jumlah *query* yang akan dilakukan. Perkiraan LOC DBM ditampilkan dalam tabel berikut:

Tabel 4.2 Analisa estimasi LOC DBM

Modul	S_{opt}	S_m	S_{pess}	S
Perkiraan baris DDL (<i>Database Definition Language</i>) pada basis data	75	85	105	87
Login	1	2	3	2
Pengubahan sandi	2	3	4	3
Administrasi data tahun anggaran	3	5	7	5
Administrasi data bidang	5	7	9	7
Administrasi data unit kerja	6	8	10	8
Administrasi data penilai	5	7	9	7
Administrasi data kriteria penilaian	5	7	9	7
Administrasi data subkriteria penilaian	6	8	10	8
Administrasi data pengguna	8	10	12	10
Administrasi data program	5	7	9	7
Administrasi data usulan kegiatan	6	8	10	8
Penilaian	8	12	16	12
Pelaporan	10	14	18	14
Transfer data	10	12	14	12
Total Estimasi LOC DBM				197

Modul Analisis Desain (DAM)

Modul analisis desain digunakan untuk memperkirakan jumlah LOC yang diperlukan pada fungsi yang ada pada SIMANCA.

Tabel 4.3 Analisa LOC DAM

Modul	S_{opt}	S_m	S_{pess}	S
Login	40	50	60	50
Pengubahan sandi	40	50	60	50
Administrasi data tahun anggaran	40	50	60	50
Administrasi data bidang	50	60	70	60
Administrasi data unit kerja	60	70	80	70
Administrasi data penilai	50	60	70	60
Administrasi data kriteria penilaian	50	60	70	60
Administrasi data subkriteria penilaian	60	80	70	80
Administrasi data pengguna	70	80	90	80
Administrasi data program	50	60	70	60
Administrasi data usulan kegiatan	60	70	80	70
Penilaian	90	110	130	110
Pelaporan	100	110	120	110

Transfer data	20	30	40	30
Total Estimasi LOC_{DAM}				940

Analisa LOC

Berdasarkan indikasi data historis yang diambil dari proyek-proyek yang telah dikerjakan, produktifitas organisasi untuk sistem adalah 620 LOC/orang-bulan.

Jumlah total estimasi LOC pada SIMANCA adalah:

$$\begin{aligned}
 \text{LOC} &= \text{LOC}_{\text{UICF}} + \text{LOC}_{\text{DBM}} + \text{LOC}_{\text{DAM}} \\
 &= 1121 + 197 + 940 \\
 &= 2258 \text{ LOC} \\
 &= 2,258 \text{ KLOC}
 \end{aligned}$$

Estimasi tenaga yang diperlukan untuk mengembangkan *prototype* SIMANCA adalah:

$$\begin{aligned}
 E_{\text{OB}} &= \text{LOC} / 620 \\
 &= 2258 / 620 \\
 &= \mathbf{3,64 \text{ orang-bulan}}
 \end{aligned}$$

Estimasi dalam orang jam:

$$\begin{aligned}
 E_{\text{OJ}} &= E_{\text{OB}} \times 173,33 \\
 &= 3,64 \times 173,33 \\
 &= \mathbf{630,92 \text{ orang-jam}}
 \end{aligned}$$

Jadi, berdasarkan perhitungan yang diperoleh dari estimasi LOC berdasarkan konstanta yang diambil dari proyek-proyek yang telah dikerjakan,

perkiraan jumlah tenaga yang diperlukan untuk mengembangkan SIMANCA adalah 3,64 orang-bulan atau setara dengan 630,92 orang-jam.

4.2.2 Estimasi Berbasis FP

Dekomposisi estimasi berbasis FP lebih dekat ke arah kawasan nilai informasi daripada pada fungsi dari perangkat lunak. Oleh karena itu, faktor pemberat untuk kompleksitas perangkat lunak SIMANCA diasumsikan bernilai sedang.

Tabel 4.4 Perhitungan estimasi FP

Informasi nilai kawasan	S_{opt}	S_m	S_{pess}	S	W	Σ FP
Jumlah masukan	10	12	14	12	4	48
Jumlah keluaran	6	8	10	8	5	40
Jumlah permintaan	7	9	11	9	6	54
Jumlah berkas	28	30	32	30	10	300
Jumlah fungsi eksternal	2	3	4	3	7	21
Jumlah total						463

Sesuai dengan teori, hasil dari penjumlahan FP kemudian dikalikan konstanta dan faktor pemberat yang dijabarkan dalam perhitungan berikut:

Tabel 4.5 Perhitungan faktor peubah kompleksitas

Faktor	Nilai
Backup dan recovery	2
Komunikasi data	5
Pemrosesan terdistribusi	0
Pentingnya performa sistem	4
Lingkungan operasi yang sudah ada	3
Pemasukan data secara online	4
Pemasukan transaksi dengan banyak layar	5
Berkas utama data di- <i>update</i> online	3
Kompleksitas kawasan informasi	3
Kompleksitas pemrosesan internal	3
Desain kode dapat digunakan kembali	3

Konversi dan instalasi	3
Instalasi lebih dari satu	1
Aplikasi didesain untuk perubahan	4
Faktor peubah kompleksitas (0,65 + 0,01 x $\sum F_i$)	1,07

Estimasi FP setelah dikalikan faktor peubah kompleksitas:

$$\begin{aligned}
 FP_{\text{estimasi}} &= \text{Jumlah total} \times \text{Faktor peubah kompleksitas} \\
 &= 463 \times 1,07 \\
 &= 495,41
 \end{aligned}$$

4.2.3 Estimasi Berbasis Proses

Estimasi berbasis proses mengacu pada analisis estimasi berbasis permasalahan. Estimasi ini menggunakan asumsi waktu yang diperlukan untuk menyelesaikan proses-proses tertentu, mulai dari awal proyek, sampai evaluasi hasil proyek.

Tabel 4.6 Perhitungan estimasi berbasis proses

Aktivitas	CC	Peren- canaan	Analisis resiko	Rekayasa		Rilis		CE	Total
				Analisis	Desain	Kode	Tes		
Proses →									
Fungsi ↓									
UICF				0,25	0,5	0,25	0,25		1,00
DBM				0,25	0,5	0,25	0,25		1,00
DAM				0,25	0,5	0,25	0,25		1,00
Total	0,25	0,25	0,25	0,75	1,5	0,75	0,75		3,75
% tenaga	1%	1%	1%	8%	45%	10%	30%		
<i>CC = customer communication, CE = customer evaluation</i>									

Konversi waktu:

$$E_{OJ} = E_{OB} \times 173,33$$

$$= 3,75 \times 173,33$$

$$= \mathbf{649,99 \text{ orang-jam}}$$

Dari estimasi berbasis proses diperoleh perkiraan tenaga kerja yang diperlukan adalah 3,75 orang bulan atau setara dengan 649,99 orang-jam.

4.2.4 Model Estimasi Empiris

Estimasi empiris dilakukan berdasarkan estimasi LOC can FP. Dalam estimasi ini digunakan persamaan (2-3), dan menggunakan berbagai kombinasi konstanta yang diambil dari beberapa model yang diajukan oleh para ahli di bidang rekayasa perangkat lunak.

Estimasi Empiris Berbasis LOC

Model Waltson-Felix

$$E_{OB} = 5,2 \times (KLOC)^{0,91}$$

$$= 5,2 \times 2,258^{0,91}$$

$$= 10,91 \text{ orang-bulan}$$

$$E_{OJ} = E_{OB} \times 173,33$$

$$= 10,91 \times 173,33$$

$$= 1891,32 \text{ orang-jam}$$

Model Bailey-Basili

$$E_{OB} = 5,5 + 0,73 \times (KLOC)^{1,16}$$

$$= 5,5 + 0,73 \times 2,258^{1,16}$$

$$= 7,38 \text{ orang-bulan}$$

$$E_{OJ} = E_{OB} \times 173,33$$

$$= 7,38 \times 173,33$$

$$= 1278,79 \text{ orang-jam}$$

Model Sederhana Boehm

$$\begin{aligned} E_{OB} &= 3,2 \times (\text{KLOC})^{1,05} \\ &= 3,2 \times 2,258^{1,05} \\ &= 7,53 \text{ orang-bulan} \end{aligned}$$

$$\begin{aligned} E_{OJ} &= E_{OB} \times 173,33 \\ &= 7,53 \times 173,33 \\ &= 1304,47 \text{ orang-jam} \end{aligned}$$

Estimasi Empiris Berbasis FP

Model Albreth dan Gaffney

$$\begin{aligned} E_{OB} &= -13,39 + 0,0545 \text{ FP} \\ &= -13,39 + (0,0545 \times 495,41) \\ &= 13,61 \text{ orang-bulan} \end{aligned}$$

$$\begin{aligned} E_{OJ} &= E_{OB} \times 173,33 \\ &= 13,61 \times 173,33 \\ &= 2358,99 \text{ orang-jam} \end{aligned}$$

Model Kemerer

$$\begin{aligned} E_{OB} &= 60,62 \times 7,728 \times 10^{-8} \text{ FP}^3 \\ &= 60,62 \times 7,728 \times 495,41^3 \times 10^{-8} \\ &= 569,61 \text{ orang-bulan} \end{aligned}$$

$$\begin{aligned} E_{OJ} &= E_{OB} \times 173,33 \\ &= 569,61 \times 173,33 \\ &= 98730,44 \text{ orang-jam} \end{aligned}$$

Model Matson, Barnett, dan Mellichamp

$$\begin{aligned}E_{OB} &= 585,7 + 15,12 \text{ FP} \\ &= 585,7 + (15,12 \times 495,41) \\ &= 8076,30 \text{ orang-bulan}\end{aligned}$$

$$\begin{aligned}E_{OJ} &= E_{OB} \times 173,33 \\ &= 8076,30 \times 173,33 \\ &= 1399864,94 \text{ orang-jam}\end{aligned}$$

Model COCOMO

Dalam model COCOMO, tingkat kompleksitas data diasumsikan bernilai sedang, sehingga diperoleh $a_b = 3,0$ dan $b_b = 1,12$. Model COCOMO yang diimplementasikan untuk memperoleh estimasi adalah model COCOMO dasar, dengan asumsi bahwa pada model COCOMO tingkat menengah, EAF memiliki nilai 1,00.

$$\begin{aligned}E_{OB} &= a_b (\text{KLOC})^{b_b} \\ &= 3,0 \times 2,258^{1,12} \\ &= 7,47 \text{ orang-bulan}\end{aligned}$$
$$\begin{aligned}E_{OJ} &= E_{OB} \times 173,33 \\ &= 7,47 \times 173,33 \\ &= 1294,69 \text{ orang-jam}\end{aligned}$$

4.3 Hasil Pengamatan Lapangan

Hasil pengamatan yang dilakukan pada proses perancangan sampai proyek pembuatan *prototype* SIMANCA selesai dilakukan. Pengamatan dilakukan dengan menghitung jumlah hari dan jam yang diperlukan berdasarkan tabel

presensi yang disediakan untuk masing-masing anggota tim yang telah disiapkan sebelumnya.

Sumber daya manusia yang terlibat dalam pengembangan *prototype* SIMANCA berjumlah 4 (empat) orang dengan komposisi analis sistem atau basis data berjumlah 1 (satu) orang, dan programmer berjumlah 3 (tiga) orang. Hasil akumulasi presensi ditampilkan dalam tabel 4.7.

Tabel 4.7 Hasil akumulasi presensi tim pengembang

Orang	Jumlah Hari	Jumlah Jam
A	18	79
B	18	81
C	16	76
D	17	83
Total orang-jam		319

4.4 Perbandingan Hasil Estimasi

Dari perhitungan yang dilakukan, maka nilai estimasi E_{OJ} yang diperoleh dari beberapa model ditampilkan pada tabel berikut:

Tabel 4.8 Perbandingan hasil estimasi antar model

Model	Estimasi E_{OJ}
Estimasi Berbasis LOC	
Analisis LOC umum	630,92
Model Waltson-Felix	1891,32
Model Bailey-Basili	1278,79
Model Sederhana Boehm	1304,47
Model COCOMO	1294,69
Estimasi berbasis FP	
Model Albrecth dan Gaffney	2358,99
Model Kemerer	98730,44
Model Matson, Barnett, dan Mellichamp	1399864,94
Estimasi berbasis proses	
Estimasi berbasis proses	649,99

Dari perhitungan diperoleh bahwa tiap model memiliki hasil perhitungan yang berbeda untuk LOC dan FP yang sama. Hasil yang berbeda diperoleh karena masing-masing model menggunakan indikator yang berbeda pada saat pembuatan estimasi. Selain itu, masing-masing hasil estimasi tersebut diperoleh berdasarkan pengalaman dari masing-masing pencipta model estimasi. Apabila data hasil estimasi dibandingkan dengan hasil pengamatan pada proyek, maka diperoleh selisih, dimana hasil estimasi tenaga lebih besar dari kenyataan yang terjadi di lapangan. Salah satu faktor yang dapat diterima sebagai alasan adalah tim yang direkrut untuk melakukan pengembangan SIMANCA telah memiliki pengalaman dalam mengerjakan proyek serupa, sehingga jumlah waktu pengerjaan sistem dapat menjadi lebih kecil dari estimasi.

Hasil estimasi tenaga terkecil diperoleh pada model estimasi LOC umum. Dari hasil pengamatan pada estimasi proyek pengembangan SIMANCA, hasil analisis LOC yang diimplementasikan pada beberapa model memiliki rerata E_{OJ} yang lebih kecil dibandingkan pada analisis serupa yang menggunakan FP. Hasil estimasi yang paling besar diperoleh pada model Kemerer dan model Matson, Barnett, dan Mellichamp. Hasil perhitungan menunjukkan bahwa hasil estimasi pada model Kemerer dan model Matson, Barnett, dan Mellichamp menghasilkan jumlah E_{OJ} yang jauh lebih besar dari rerata estimasi dengan menggunakan model lainnya.

Pada estimasi berbasis proses, digunakan analisis yang independen, dengan membagi proses menjadi aktifitas-aktifitas yang lebih kecil. Penghitungan

E_{OJ} dilakukan secara langsung pada tiap aktifitas, kemudian dijumlahkan, sehingga total E_{OJ} langsung dapat diketahui. Pada estimasi proyek pengembangan SIMANCA diperoleh bahwa jumlah E_{OJ} pada estimasi berbasis proses mendekati jumlah E_{OJ} pada estimasi berbasis LOC.

Dari hasil implementasi terhadap proyek SIMANCA diperoleh bahwa model analisis yang paling mendekati hasil di lapangan adalah analisis berbasis LOC yang menggunakan indikasi data historis. Hasil analisis yang paling jauh dari hasil yang diperoleh dari lapangan adalah estimasi berbasis FP yang menggunakan model Matson, Barnett, dan Mellichamp. Namun demikian, jika masing masing-model estimasi diambil reratanya, maka model estimasi berbasis proses memiliki rerata yang paling mendekati hasil di lapangan, dilanjutkan dengan rerata estimasi berbasis LOC dan yang paling jauh dari hasil pengamatan di dalam pelaksanaan proyek SIMANCA adalah rerata estimasi berbasis FP.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan implementasi dan pembahasan yang terdapat di dalam tugas akhir ini, dapat disimpulkan bahwa:

1. Estimasi perangkat lunak digunakan untuk melakukan perkiraan terhadap sumber daya dan biaya yang diperlukan untuk menyelesaikan suatu proyek pengembangan perangkat lunak.
2. Kebenaran hasil estimasi perangkat lunak baru dapat diketahui ketika suatu proyek pengembangan perangkat lunak telah diselesaikan.
3. Untuk memperoleh perkiraan yang tepat perlu dilakukan pembelajaran dengan melakukan kalibrasi pada perumusan konstanta estimasi, sesuai dengan kondisi yang berlaku pada proyek pengembangan perangkat lunak yang sedang dijalankan.

5.2 Saran

Sebagai penutup untuk tugas akhir ini, pembelajaran yang berkelanjutan pada proyek-proyek perangkat lunak sangat diperlukan untuk memperoleh suatu acuan perkiraan tenaga dan biaya yang paling efektif, sehingga dapat diperoleh suatu angka biaya yang bisa diterima oleh pihak pengembang maupun pihak klien atau rekanan dalam suatu proyek pengembangan perangkat lunak.

DAFTAR PUSTAKA

- B. Hetzel, *Making Software Measurement Work: Building an Effective Measurement Program*, QED Technical Publishing Group, Boston, Massachusetts, 1993, ISBN: 0471565687
- C. Jones, *Programming Productivity*, McGraw-Hill Book Company, New York, New York, 1986, ISBN: 0070328110
- Grady, Robert B., *Practical Software Metrics for Project Management and Process Improvement*, Prentice-Hall, Inc., 1992, ISBN: 0137203845
- Lowell Jay Arthur, *Measuring Programmer Productivity and Software Quality*, John Wiley & Sons, Inc., New York, NY, 1985, ISBN: 0471887137
- Pressman, R. S. *Software Engineering A Practitioner's Approach*, 5th Edition, New York, McGraw Hill, 2000, ISBN: 0073655783
- Albrecht, A. J., *Measuring Application Development Productivity*, IBM Applications Development Symposium, Monterey, CA , 1979 , pp. 83-92
- D.J. Paulish and A.D. Carleton, *Case Studies of Software-Process-Improvement Measurement*, IEEE Computer, Vol. 27, No. 9, September 1994, pp. 50 – 57
- J.E. Matson, B.E. Barret, and J.M. Mellichamp, *Software Development Cost Estimation Using Function Points*, IEEE Trans. Software Eng., vol. 20, no. 4, pp. 275–287, April 1994
- Wikipedia, <http://www.wikipedia.org/>
- ConvertAll, <http://www.bellz.org/convertall/>