

ANALISIS DAN PERANCANGAN MANAJEMEN PROYEK PERANGKAT LUNAK STUDI KASUS : PT. EKIOSKU

Kiki Kusumawati¹, Nurul Chafid², Sri Ariyani³

Dosen Teknik dan Mahasiswa Program Studi Sistem Informasi, Fakultas Teknik
Universitas Satya Negara Indonesia
Jl.Arteri Pondok Indah No.11 Kebayoran Lama – Jakarta Selatan
Email : aiyyarii26@gmail.com

ABSTRAK

Seiring dengan banyaknya proyek yang diterima dan harus diselesaikan tepat waktu oleh PT. Ekiosku, maka dibutuhkan sistem informasi yang dapat digunakan untuk memajemen proyek. Masalah yang sering terjadi pada pengerjaan proyek perangkat lunak adalah kurang optimalnya hasil estimasi waktu, sumber daya manusia, dan perkiraan biaya proyek perangkat lunak yang dilakukan oleh manajer proyek. Pembagian tugas yang dilakukan oleh manajer proyek dilakukan dengan cara lisan kurang efisien karena setiap tenaga kerja memiliki kemampuan yang berbeda. Tidak adanya gambaran pengerjaan proyek dan penjadwalan sebagai penunjang sehingga manajer proyek tidak mengetahui bagaimana perkembangan proyek. Oleh karena itu, sistem informasi manajemen proyek perangkat lunak ini dibangun untuk membantu dalam otomatisasi dan perhitungan menggunakan metode *Function Point* dan *COCOMO* sehingga dapat mempermudah manajer proyek dalam mengetahui lingkup kerja dan penjadwalan, serta untuk mengukur estimasi biaya dan waktu yang dibutuhkan untuk membangun suatu proyek. Mengurangi resiko kegagalan dalam pembangunan perangkat lunak dan untuk memonitor proyek yang akan berjalan, sedang berjalan, dan selesai, serta melihat sejauh mana perkembangan proyek yang dikerjakan oleh developer.

Kata Kunci: Manajemen Proyek, Estimasi Waktu, Estimasi Sumber Daya Manusia, Estimasi Biaya, *Function Point*, *COCOMO*

ABSTRACT

Along many projects that received and must be completed on time by PT. Ekiosku, requires information system that can be used to manage the project. The problem that often occurs in the implementation of software projects is the lack of optimal results from estimation of time, human resources, and estimation of project software costs carried out by the project manager. The division of tasks carried out by the project manager is done verbally inefficient because each workforce has different abilities. There is no image implementation and project scheduling as support so the project manager does not know how the project progresses. Therefore, this software project management information system is built to assist in automation and calculation using Function Point and COCOMO methods so that it can easier the project manager in knowing the scope of work and scheduling, and to measure the estimated costs and time needed to build a project. Reducing the risk of failure in software development and for monitoring projects that will run, running, and completed, and seeing the extent to which the project is being developed by the developer.

Keywords: *Project Management, Time Estimation, Human Resource Estimation, Cost Estimation, Function Point, COCOMO*

PENDAHULUAN

Perkembangan perangkat lunak yang pesat mengakibatkan banyaknya perusahaan menggunakan jasa konsultan yang menyediakan layanan pembuatan perangkat lunak (*software*) sesuai dengan permintaan klien salah satunya adalah PT. Ekiosku. PT. Ekiosku merupakan perusahaan yang bergerak di bidang industri teknologi informasi dan digital, dimana semakin bertambahnya jumlah klien semakin banyak pula proyek pembuatan perangkat lunak yang diterima dan harus diselesaikan tepat waktu.

Masalah yang sering terjadi pada pengerjaan proyek perangkat lunak adalah kurang optimalnya hasil estimasi waktu, sumber daya manusia, dan perkiraan biaya proyek perangkat lunak yang dilakukan oleh manajer proyek. Pembagian tugas yang dilakukan oleh manajer proyek dilakukan dengan cara lisan (berbicara langsung) kurang efisien karena setiap tenaga kerja memiliki kemampuan yang berbeda. Tidak adanya gambaran pengerjaan proyek dan penjadwalan sebagai penunjang sehingga manajer proyek tidak mengetahui bagaimana perkembangan proyek, akibatnya pada saat monitoring pengerjaan terhadap proyek sering terjadi kesalahan, sehingga proyek tidak selesai tepat waktu, kurangnya sumber daya manusia untuk menyelesaikan proyek, dan membengkaknya biaya pelaksanaan karena telah melebihi anggaran awal.

Oleh karena itu, sistem informasi manajemen proyek perangkat lunak ini dibangun untuk membantu dalam otomatisasi dan perhitungan menggunakan metode *Function Point* dan COCOMO dalam mengetahui lingkup kerja dan penjadwalan, serta untuk mengukur estimasi biaya dan waktu. Mempermudah dalam mengidentifikasi dan mengurangi resiko kegagalan dalam pembangunan perangkat lunak yang akan memberikan keuntungan bagi PT. Ekiosku sekaligus meningkatkan citra perusahaan. Untuk memonitor proyek yang akan berjalan, sedang berjalan, dan selesai, serta melihat sejauh mana perkembangan proyek yang dikerjakan oleh developer.

LANDASAN TEORI

a. Manajemen Proyek Perangkat Lunak

Perangkat Lunak adalah perintah (program komputer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan, struktur data yang memungkinkan program memanipulasi informasi secara proporsional, dan dokumen yang menggambarkan operasi dan kegunaan komputer. (Roger S. Pressman, 2002)

Menurut (Duncan, 1996) dalam bukunya yang berjudul *A guide to the project manajement body of knowledge* “Manajemen Proyek merupakan pengaplikasian pengetahuan, keterampilan, peralatan, dan teknik untuk kegiatan proyek dalam memenuhi persyaratan proyek”. Pengelola proyek tidak hanya harus berusaha untuk mengenalkan ruang lingkup yang spesifik, waktu, biaya dan kualitas proyek, tetapi mereka juga harus memfasilitasi semua kebutuhan melalui komunikasi antar orang dan pihak (stakeholder) yang terlibat dalam proyek (Munir, 2015).

b. Model Function Point

Matrik perangkat lunak *function-oriented* (berorientasi pada fungsi) menggunakan sebuah pengukuran fungsionalitas yang disampaikan oleh aplikasi sebagai suatu nilai normalisasi. Metrik berorientasi fungsi pertama kali diusulkan

oleh (Albrecht A. J., 1979) yang mengusulkan sebuah pengukuran yang disebut *function point* (FP). *Function point* ditarik menggunakan sebuah hubungan empiris berdasarkan pengukuran (langsung) domain informasi perangkat lunak yang dapat dihitung serta perkiraan kompleksitas perangkat lunak (Verdi Yasin, 2012). Lima karakteristik domain informasi ditentukan dan penghitungan diberikan di dalam lokasi tabel yang sesuai. Nilai domain informasi didefinisikan dengan cara sebagai berikut:

Parameter Pengukuran	Jumlah	Sederhana	Rata-Rata	Kompleks	Faktor Pembobotan
Jumlah Input Pemakai	<input type="text"/> x	3	4	6	= <input type="text"/>
Jumlah Output Pemakai	<input type="text"/> x	4	5	7	= <input type="text"/>
Jumlah Penyelidikan Pemakai	<input type="text"/> x	3	4	6	= <input type="text"/>
Jumlah File	<input type="text"/> x	7	10	15	= <input type="text"/>
Jumlah Interface Internal	<input type="text"/> x	6	7	10	= <input type="text"/>
Total	→				<input type="text"/>

Gambar 1 Perhitungan Metrik *Function Point*

Jumlah input pemakai. Setiap input pemakai yang memberikan data yang berorientasi pada aplikasi yang jelas pada perangkat lunak dihitung. Input ini harus dibedakan dari penelitian yang dihitung secara terpisah. Jumlah output pemakai. Setiap output pemakai yang memberikan informasi yang berorientasi pada aplikasi kepada pemakai dihitung. Pada konteks ini output mengacu pada laporan, layar, tampilan kesalahan, dan sebagainya. Jenis data individual pada sebuah laporan tidak dihitung secara terpisah. Jumlah penyelidikan pemakai. Sebuah penyelidikan didefinisikan sebagai input online yang mengakibatkan munculnya beberapa respon perangkat lunak yang cepat dalam bentuk sebuah output online. Setiap penyelidikan yang jelas dihitung.

Jumlah file. Setiap file master logika (yaitu pengelompokan data secara logis yang menjadi suatu bagian dari sebuah database yang besar atau sebuah file yang terpisah) dihitung. Jumlah *interface eksternal*. Semua *interface* yang dapat dibaca oleh mesin (contohnya, file data pada pita atau *disket*) yang digunakan untuk memindahkan informasi ke sistem yang lain dihitung. Sekali data tersebut dikumpulkan, maka sebuah nilai kompleksitas akan dihubungkan dengan masing-masing perhitungan. Organisasi yang menggunakan metode titik fungsi mengembangkan kriteria untuk menentukan apakah sebuah entri tertentu bersifat sederhana, rata-rata, atau kompleks. Meskipun demikian perkiraan kompleksitas tetap bersifat subyektif. Untuk menghitung titik-titik *function point* (FP) digunakan persamaan rumus sebagai berikut:

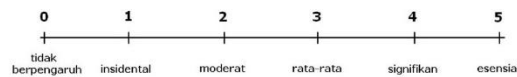
$$FP = \text{jumlah total} \times (0,65 + (0,01 \times \sum F_i)) \dots\dots(1)$$

Keterangan:

- Jumlah total adalah jumlah semua entri yang diperoleh dari gambar 1.
- $\sum F_i$ adalah nilai dari $i = 1$ sampai 14 adalah harga penyesuaian kompleksitas berdasarkan respon pada pertanyaan (Arthur, 1985) berikut ini:
 1. Apakah sistem membutuhkan *backup* dan *recovery* yang reliabel?

2. Apakah komunikasi data dibutuhkan?
3. Apakah fungsi pemrosesan didistribusikan?
4. Apakah kinerja penting?
5. Apakah sistem berjalan pada lingkungan operasional yang sudah ada yang paling banyak digunakan?
6. Apakah sistem membutuhkan entri data online?
7. Apakah entri data online membutuhkan ada transaksi input terhadap layar atau operasi ganda?
8. Apakah file master diperbarui secara online?
9. Apakah input, output, file atau penyelidikan kompleks?
10. Apakah pemrosesan internal kompleks?
11. Apakah kode didesain untuk dapat dipakai kembali?
12. Apakah desain melibatkan konversi dan instalasi?
13. Apakah sistem didesain untuk instalasi ganda dalam organisasi berbeda?
14. Apakah aplikasi didesain untuk memfasilitasi perubahan dan mempermudah pemakai untuk menggunakannya?

Untuk jawaban tiap faktor bernilai antara 0-5 yang dapat dilihat pada gambar 2 berikut ini:



Gambar 2 Arti Faktor pada *Function Point*

Untuk menghitung estimasi LOC (*Line of Code*) dan KLOC (*Kilo Line of Code*) menggunakan persamaan berikut ini:

$$LOC_{\text{estimasi}} = FP \times LOC \quad \dots\dots\dots (2)$$

$$KLOC = LOC_{\text{estimasi}} / 1000 \quad \dots\dots\dots (3)$$

Hubungan antara baris-baris kode dan function point tergantung pada Bahasa pemrograman yang digunakan untuk mengimplementasi perangkat lunak dan kualitas desain. Estimasi kasar terhadap rata-rata jumlah baris kode yang diperlukan untuk membangun satu *function point* dalam berbagai bahasa pemrograman (Albrecht & Gaffney, November 1983), dapat dilihat pada tabel 1 berikut ini:

Tabel 1 Konversi FP ke LOC

Bahasa Pemrograman	LOC/FP (rata-rata)
Bahasa assembly	320
C	128
Cobol	105
Fortran	105
Pascal	90
Ada	70
Bahasa berorientasi objek	30
Bahasa generasi keempat (4GLs)	20
Generator kode	15
Spreadsheets	6

Bahasa grafis (icon)	4
----------------------	---

c. Model Constructive Cost Model (CoCoMo)

Constructive *Cost Model* (COCOMO) adalah sebuah hirarki model estimasi perangkat lunak dikembangkan oleh (Boehm, 1981). Model ini menganggarkan jumlah biaya dengan istilah *personal-month* (PM) (Munir, 2015). Hirarki model berbentuk sebagai berikut:

- Model 1 : Model COCOMO Dasar menghitung usaha pengembangan perangkat lunak dan biaya sebagai fungsi dari ukuran program yang diekspresikan dalam baris kode yang diestimasi.
- Model 2 : Model COCOMO Intermediate menghitung usaha pengembangan perangkat lunak sebagai fungsi ukuran program dan serangkaian “pengendali biaya” yang menyangkut penilaian yang subyektif terhadap produk, perangkat keras personil, dan atribut proyek.
- Model 3 : Model COCOMO Advance menghubungkan semua karakteristik versi intermediate dengan penilaian terhadap pengaruh pengendali biaya pada setiap langkah (analisis, perancangan, dll) dari proses rekayasa perangkat lunak.

Model COCOMO ditetapkan untuk tiga kelas proyek perangkat lunak. Dengan menggunakan terminologi (Boehm, 1981), antara lain:

1. Mode organik adalah proyek perangkat lunak yang sederhana dan relatif kecil dengan pengalaman aplikasi yang baik mengerjakan serangkaian kebutuhan yang lebih tidak tegar (misalnya program analisis termal yang dikembangkan untuk kelompok transfer panas).
2. Mode semi-detached adalah proyek perangkat lunak menengah (dalam ukuran dan kompleksitas) dimana tim dengan pengalaman pada tingkat yang berbeda-beda harus memenuhi bauran yang kurang kuat dari syarat yang ketat (misalnya sistem pemrosesan transaksi dengan syarat tertentu untuk perangkat keras termal dan perangkat lunak database).
3. Mode embedded adalah proyek perangkat lunak yang harus dikembangkan ke dalam serangkaian perangkat keras, perangkat lunak dan batasan operasional yang ketat (seperti perangkat lunak control penerbangan untuk pesawat udara).

Untuk menghitung COCOMO dasar didapatkan persamaan sebagai berikut ini:

$$E = a_b (KLOC)^{b_b} \dots\dots\dots (4)$$

$$D = c_b (E)^{d_b} \dots\dots\dots (5)$$

$$P = E / D \dots\dots\dots (6)$$

Untuk nilai a_b , b_b , c_b , d_b adalah koefisien yang diberikan, dengan nilai yang tertera pada tabel 2 berikut ini:

Tabel 2 Model COCOMO Dasar

Proyek Perangkat Lunak	a_b	b_b	c_b	d_b
organik	2,4	1,05	2,5	0,38
semi-detached	3,0	1,12	2,5	0,35

embedded	3,6	1,20	2,5	0,32
----------	-----	------	-----	------

Untuk menghitung estimasi biaya proyek didapatkan persamaan sebagai berikut ini:

$$\begin{aligned} \text{Biaya FP} &= \text{Upah} / \text{LOC} \dots\dots\dots(7) \\ \text{Estimasi Biaya Proyek} &= \text{Biaya FP} \times \text{Estimasi FP} \dots\dots\dots(8) \end{aligned}$$

METODE PENELITIAN

a. Metodologi Pengumpulan Data (Primer dan Sekunder)

Pelaksanaan penelitian menggunakan beberapa metode untuk mengumpulkan data terkait dan dibutuhkan untuk bahan penelitian. Adapun metode pengumpulan tersebut adalah sebagai berikut:

1) Metode Pengumpulan Data Primer

- Wawancara, pada tahap ini dilakukan proses wawancara yang dilakukan pada setiap bagian yang terkait dalam proses manajemen proyek pada PT. Ekiosku untuk menggali data mengenai cara pengerjaan proyek IT.
- Observasi, proses pengamatan langsung dilakukan untuk mengetahui alur pengerjaan proyek IT pada masing-masing tim. Pada tahap ini akan dilakukan penggalian informasi dengan menganalisa tahap-tahap pengerjaan proyek IT pada setiap bagian yang terkait.
- Studi Literatur, pada tahap ini dilakukan pendalaman *e-book*, jurnal, karya tulis ilmiah, internet dan literatur lainnya yang berkaitan dengan implementasi manajemen proyek perangkat lunak.

2) Metode Pengumpulan Data Sekunder

Metode pengumpulan data sekunder sering disebut metode pengumpulan bahan dokumen, karena peneliti secara tidak langsung mengambil data sendiri tapi memanfaatkan data atau dokumen yang dihasilkan oleh pihak – pihak lain. Data sekunder yang digunakan oleh pihak peneliti pada umumnya untuk memberikan gambaran tambahan, gambaran pelengkap, ataupun diproses lebih lanjut.

Dalam metode pengumpulan data sekunder, obsevator tidak meneliti langsung, tetapi data didapatkan misalnya dari internet dalam hal untuk mengetahui berbagai pengetahuan dan karya yang pernah dicapai oleh para peneliti terdahulu.

b. Metodologi Perancangan Sistem

Untuk metode perancangan sistem penulis akan memilih metode *Waterfall* karena metode ini dinilai relevan dan cocok untuk sistem yang berbasis sekuensial, dimana setiap tahapan memiliki pengaruh yang terintegrasi, mulai dari proses yang paling penting yaitu analisa kebutuhan, desain sistem, penulisan kode program, pengujian program, hingga penerapan program dan pemeliharaan.

c. Analis Kebutuhan Sistem

Adapun kebutuhan dalam pembuatan aplikasi ini, pada tabel 3 menjelaskan spesifikasi yang digunakan untuk mendukung pembuatan aplikasi manajemen proyek perangkat lunak.

Tabel 3 Analisa Kebutuhan Sistem

Software	Hardware
Sistem Operasi Linux Ubuntu 17.10	Laptop Acer Aspire 4738 dengan spesifikasi RAM 6 GB, <i>Processor Intel Core i5</i> , SSD 120 GB
<i>Netbeans IDE</i>	
XAMPP (PHP, MySQL, Apache Server, Java)	
<i>Google Chrome (lastest edition)</i>	

HASIL DAN PEMBAHASAN

a. Perhitungan Biaya dengan Metode Function Point dan COCOMO

Langkah pertama dalam melakukan estimasi biaya dan waktu adalah dengan melakukan identifikasi pada fungsi-fungsi yang digunakan sebagai parameter perhitungan pada perangkat lunak. Ukuran ditentukan oleh jumlah input pemakai, jumlah output pemakai, jumlah penyelidikan pemakai, jumlah file, dan jumlah interface internal. Komponen ini memiliki kategori sederhana, rata-rata, dan kompleks tergantung pada karakteristik yang dimiliki. Berdasarkan data yang didapat dari proyek Pengadaan Sistem Qe Chain Supply PT. Pelita Indonesia Djaya (PT. PIDC).

Langkah kedua, jumlah kriteria dari tiap fungsi komponen yang dimiliki dikalikan dengan masing-masing bobot faktor, berdasarkan perhitungan metrik *function point* pada gambar 1 yang dapat dilihat pada tabel 4 dibawah ini.

Tabel 4 Perhitungan Metrik *Function Point* pada Proyek Pengadaan Sistem Qe Chain Supply

Parameter Pengukuran	Sederhana		Rata-Rata		Kompleks		Faktor Pembobotan
	Jml	Bbt	Jml	Bbt	Jml	Bbt	
Jumlah Input Pemakai	13	3	8	4	3	6	89
Jumlah Output Pemakai	10	4	7	5	3	7	96
Jumlah Penyelidikan Pemakai	20	3	11	4	3	6	122
Jumlah File	18	7	3	10	0	15	156
Jumlah Interface Internal	0	6	0	7	1	10	10
Total							473

Keterangan:

- Jml : Jumlah
- Bbt : Bobot

Berdasarkan perhitungan pada tabel 4 didapatkan total faktor pembobotan adalah 473. Langkah ketiga, melakukan perhitungan harga penyesuaian kompleksitas dengan melakukan penilaian pada 14 pertanyaan pada tabel 5 untuk jawaban tiap factor bernilai antara 0-5 berdasarkan pada gambar 2.

Tabel 5 Harga Penyesuaian Kompleksitas pada Proyek Pengadaan Sistem Qc Chain Supply

No	Faktor	Bobot
1	Apakah sistem membutuhkan <i>backup</i> dan <i>recovery</i> yang reliabel?	5
2	Apakah komunikasi data dibutuhkan?	4
3	Apakah fungsi pemrosesan didistribusikan?	4
4	Apakah kinerja penting?	4
5	Apakah sistem berjalan pada lingkungan operasional yang sudah ada yang paling banyak digunakan?	0
6	Apakah sistem membutuhkan <i>entry data online</i> ?	3
7	Apakah <i>entry data online</i> membutuhkan ada transaksi <i>input</i> terhadap layar atau operasi ganda?	2
8	Apakah <i>file master</i> diperbarui secara <i>online</i> ?	2
9	Apakah <i>input</i> , <i>output</i> , <i>file</i> atau penyelidikan kompleks?	3
10	Apakah pemrosesan internal kompleks?	3
11	Apakah kode didesain untuk dapat dipakai kembali?	3
12	Apakah desain melibatkan konversi dan instalasi?	2
13	Apakah sistem didesain untuk instalasi ganda dalam organisasi berbeda?	2
14	Apakah aplikasi didesain untuk memfasilitasi perubahan dan mempermudah pemakai untuk menggunakannya?	3
Total Kompleksitas		40

Setelah memperoleh nilai total faktor pembobotan dan total kompleksitas, maka langkah keempat adalah menentukan nilai dari *function point* dengan memasukan total faktor pembobotan dan total kompleksitas berdasarkan pada persamaan rumus pertama.

$$FP = \text{jumlah total} \times (0,65 + (0,01 \times \Sigma F_i))$$

$$FP = 473 \times (0,65 + (0,01 \times 40))$$

$$\begin{aligned} &= 473 \times 1,05 \\ &= 496,65 \end{aligned}$$

Setelah *function point* dihitung, maka langkah kelima adalah melakukan konversi FP ke LOC berdasarkan tabel 1. Proyek Pengadaan Sistem Qe Chain Supply dibangun menggunakan bahasa pemrograman berorientasi objek, maka didapatkan perhitungan konversi dimana nilai LOC adalah 30. Sehingga nilai LOC dapat dimasukkan pada persamaan rumus dua.

$$\begin{aligned} LOC_{\text{estimasi}} &= FP \times LOC \\ LOC_{\text{estimasi}} &= 496,65 \times 30 \\ &= 14.899,5 \end{aligned}$$

Langkah keenam, menghitung KLOC dengan membagi hasil estimasi LOC sesuai dengan persamaan rumus tiga.

$$\begin{aligned} KLOC &= LOC_{\text{estimasi}} / 1000 \\ KLOC &= 14.899,5 / 1000 \\ &= 14,8995 \end{aligned}$$

Langkah ketujuh, setelah semua tahap pada metode *Function Point* telah dilalui dilanjutkan ketahap perhitungan menggunakan metode COCOMO. Berdasarkan kategori proyek COCOMO, proyek Pengadaan Sistem Qe Chain Supply merupakan proyek kecil karena terdiri dari beberapa orang. Selain itu tipe proyeknya adalah semi detached, berdasarkan tabel 2 yang memiliki nilai a_b dan b_b masing-masing yaitu 3,0 dan 1,12. Untuk mengetahui berapa banyak usaha (*Effort*) untuk menyelesaikan proyek Pengadaan Sistem Qe Chain Supply, maka digunakan persamaan rumus empat.

$$\begin{aligned} E &= a_b (KLOC)^{b_b} \\ E &= 3,0 (14,8995)^{1,12} \\ &= 3,0 (20,6040) \\ &= 61,8121 \text{ PM} \end{aligned}$$

Berdasarkan perhitungan *Effort* dibutuhkan 62 orang/bulan untuk menyelesaikan proyek Pengadaan Sistem Qe Chain Supply. Langkah kedelapan, untuk menentukan waktu yang dibutuhkan untuk menyelesaikan proyek tersebut dengan menggunakan persamaan rumus lima.

$$\begin{aligned} D &= c_b (E)^{d_b} \\ D &= 2,5 (61,8120)^{0,35} \\ &= 10,588 \text{ M} \end{aligned}$$

Berdasarkan perhitungan diatas, waktu yang dibutuhkan adalah 10,6 bulan. Selanjutnya dapat ditentukan jumlah orang yang dibutuhkan adalah dengan menggunakan persamaan rumus enam.

$$\begin{aligned} P &= E / D \\ P &= 61,8120 / 10,588 \\ &= 5,8379 \text{ P} \end{aligned}$$

Sehingga diketahui jumlah orang yang dibutuhkan untuk menyelesaikan proyek Pengadaan Sistem Qe Chain Supply sebanyak 6 orang. Langkah kesepuluh untuk menghitung biaya FP maka digunakan persamaan rumus tujuh, dengan upah karyawan sebesar Rp. 4.500.000 tiap bulan.

$$\begin{aligned}\text{Biaya FP} &= \text{Upah} / \text{LOC} \\ \text{Biaya FP} &= 4.500.000 / 30 \\ &= 150.000\end{aligned}$$

Langkah kesebelas, untuk menghitung estimasi biaya proyek dapat digunakan persamaan rumus delapan dengan mengalikan biaya FP dan estimasi FP.
Estimasi Biaya Proyek = Biaya FP x Estimasi FP
Estimasi Biaya Proyek = 150.000 * 496,65
= 74.497.500

Berdasarkan perhitungan diatas didapatkan estimasi biaya pengerjaan proyek Pengadaan Sistem Qe Chain Supply sebesar Rp. 74.497.500 dengan waktu pengerjaan 11 bulan dengan 6 orang pekerja. Hasil analisis dari perbandingan estimasi dengan menggunakan metode *Function Point* (FP) dan COCOMO pada pengerjaan proyek Pengadaan Sistem Qe Chain Supply dengan metode perkiraan dapat dilihat pada tabel 6.

Tabel 6 Hasil Analisis Perbandingan

	Metode FP dan COCOMO	Metode Perkiraan
Sumber Daya Manusia	6 orang	7 orang
Durasi Pengerjaan	10,6 bulan	3 bulan
Biaya Total	Rp. 74.497.500	Rp. 81.125.000

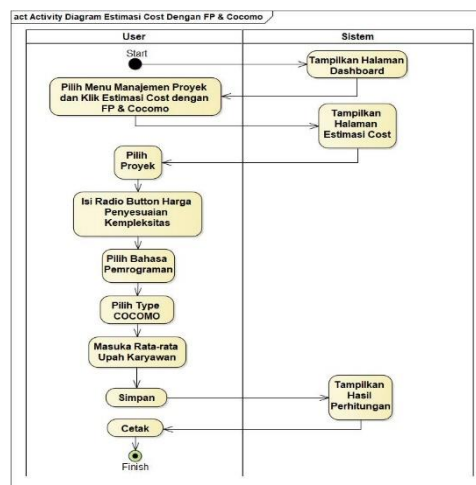
PT. Ekiosku memperkirakan waktu untuk menyelesaikan proyek Pengadaan Sistem Qe Chain Supply selama 3 bulan dengan melihat sumber daya manusia yang ada. Sedangkan dengan menggunakan metode *Function Point* (FP) dan COCOMO proyek diperkirakan akan selesai dalam waktu 10,6 bulan. Terdapat selisih yang cukup jauh pada perkiraan waktu untuk menyelesaikan proyek ini, karena metode *Function Point* (FP) dan COCOMO menilai setiap fungsi komponen yang dimiliki dengan masing-masing bobot faktor. Selain itu metode *Function Point* (FP) dan COCOMO juga menilai faktor penyesuaian kompleksitas yang diisi oleh proyek manajer dan analis. Jadi berdasarkan pada tabel 9 dapat disimpulkan perbedaan total biaya proyek disebabkan oleh perbedaan waktu penyelesaian proyek. Perbedaan tersebut dipengaruhi oleh beberapa faktor yang tidak dilakukan pada metode perkiraan yang digunakan PT. Ekiosku. Sehingga didapatkan hasil estimasi biaya yang lebih akurat dan dapat mengurangi masalah-masalah yang akan ditimbulkan pada saat pengerjaan proyek menggunakan metode *Function Point* (FP) dan COCOMO dibanding dengan metode perkiraan.

b. Perancangan Sistem

Pada sistem Manajemen Proyek Perangkat Lunak PT. Ekiosku, terdapat beberapa aktor dan akses menu yang terlibat di dalam sisem ini antara lain:

- a) Admin dapat mengakses menu karyawan, klien, provinsi, kota, kecamatan, kelurahan, kategori proyek, cetak invoice, cetak bukti serah terima, pembayaran, laporan rekapitulasi proyek per periode, laporan norminatif proyek per periode, laporan kinerja karyawan, dan laporan sejarah proyek per klien.
- b) Analis dapat mengakses menu metrik *function point*, bahasa pemrograman, proyek, pencapaian, agenda, modul, sub modul, dan isu.
- c) Designer dapat mengakses menu proyek, pencapaian, agenda, modul, sub modul, dan isu.
- d) Direktur dapat mengakses menu laporan norminatif proyek per periode, laporan sejarah proyek per klien, laporan rekapitulasi proyek per periode, laporan kinerja karyawan, log login, dan log aktivitas.
- e) Marketing dapat mengakses menu klien, cetak invoice, pembayaran, cetak bukti serah terima, estimasi *cost* fp dan cocomo, laporan rekapitulasi proyek per periode, laporan norminatif proyek, dan laporan sejarah proyek per klien.
- f) Programmer dapat mengakses menu proyek, pencapaian, agenda, modul, sub modul dan isu.
- g) Proyek Manajer dapat mengakses menu proyek, pencapaian, agenda, modul, sub modul, isu, estimasi *cost* fp dan cocomo, bahas pemrograman, metrik *function point*, laporan sejarah proyek per klien, laporan rekapitulasi proyek per periode, laporan norminatif proyek.
- h) Software tester dapat mengakses menu proyek, agenda, pencapaian, dan isu.

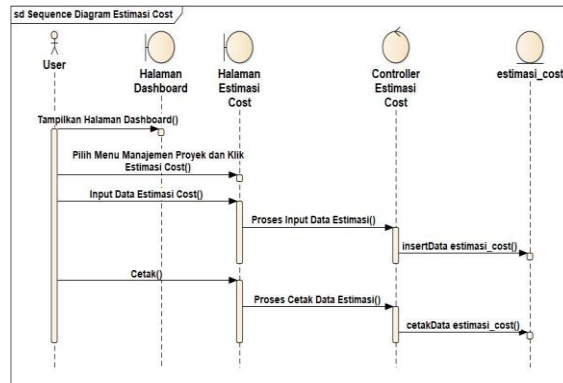
Activity Diagram Estimasi Cost dengan FP dan COCOMO



Gambar 3 Activity Diagram Estimasi Cost dengan FP dan COCOMO

Pada gambar 3, menunjukkan bahwa *user* dapat melakukan perhitungan estimasi proyek untuk mengetahui perkiraan biaya, waktu, dan jumlah tenaga kerja yang diperlukan untuk membangun suatu proyek perangkat lunak.

Sequence Diagram Estimasi Cost dengan FP dan COCOMO



Gambar 4 Sequence Diagram Estimasi Cost dengan FP dan COCOMO

Pada gambar 4, menjelaskan alur proses perhitungan estimasi. Proses dimulai ketika *user* memilih menu Estimasi Cost dengan FP dan COCOMO lalu *user* melakukan input dan menyimpannya yang kemudian akan direspon oleh sistem agar estimasi dapat dicetak.

KESIMPULAN DAN SARAN

a. Kesimpulan

Adapun kesimpulan dari penulisan ini adalah:

- Telah berhasil merancang sistem informasi manajemen perangkat lunak pada PT. Ekiosku yang dapat mempermudah manajer proyek untuk mengetahui lingkup kerja dan penjadwalan, serta untuk mengukur estimasi biaya dan waktu yang dibutuhkan untuk membangun suatu proyek.
- Mendapatkan hasil estimasi yang lebih akurat dengan melakukan perhitungan estimasi menggunakan metode *Function Point* dan COCOMO serta dapat mengurangi resiko kegagalan dalam pembangunan perangkat lunak yang akan memberikan keuntungan bagi PT. Ekiosku sekaligus meningkatkan citra perusahaan.

b. Saran

Untuk kemajuan sistem yang digunakan di masa yang akan datang maka penulis memiliki saran agar dilakukannya pelatihan kepada *user* dalam menggunakan aplikasi yang telah dibuat.

DAFTAR PUSTAKA

- Albrecht, A. J. (1979). *Measuring Application Development Productivity*. Monterey, CA: IBM Applications Development Symposium.
- Albrecht, A., & Gaffney, d. J. (November 1983). *Software Function, Source Lines of Code and Development Effort Production, A Software Science Validation* (Vol. VI). IEEE Trans. Software Engineering.
- Arthur, L. J. (1985). *Measuring Programmer Productivity dan Software Quality*. Wiley - Interscience.

- Boehm, B. (1981). *Software Engineering Economics*. Prentice-Hall.
- Duncan, W. R. (1996). *A guide to the Project Management body of knowledge - PMBoK. 1st ed. PMI Standards Committee, Upper Daryby, USA: PMBook.*
- Munir. (2015). *Manajemen Proyek Perangkat Lunak*. Bandung: UPI Press.
- Roger S. Pressman, P. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*. (C. Harnaningrum, Trans.) Yogyakarta: ANDI and McGraw-Hill Book Co.
- Verdi Yasin, S. M. (2012). *Rekayasa Perangkat Lunak Berorientasi Objek. Permodelan, Arsitektur dan Perancangan (Modeling, Architecture and Design)*. Jakarta: Mitra Wacana Media.