

PERANCANGAN SISTEM ESTIMASI BIAYA PROYEK PENGEMBANGAN SOFTWARE

Suharjito, Agus Widodo, Budi Prasetyo

Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Bina Nusantara

Jl. K.H. Syahdan No. 9, Kemanggis Palmerah, Jakarta

E-mail: harjito@yahoo.com

ABSTRACT

A project software is categorized as successful if that project is delivered on time, within the specified budget, and having the expected quality. To achieve those goals, a project manager must be able to estimate realistic cost and effort for the project. However, the complex and intangible characteristics of software project cause difficulties in estimating the software project. The purpose of this research is to devising a system estimate expense and effort development project of software have matching with condition in the country. The developed estimation model in this study is a parametric method which uses local data so that it could yield higher validity for local software projects. The steps undertaken in this study are literature study, data analysis, mathematic modeling, implementation and testing. The conclusion is this application applicable to estimate expense and effort project of software with characteristic of information system by using function points metric.

Keyword: project, software, Metrik Function Point, model, estimate realistic cost and effort

1. PENDAHULUAN

1.1 Latar Belakang

Dalam proyek fisik seperti pembangunan jembatan atau pembangunan jalan, estimasi biaya dan usaha proyek dapat dilakukan dengan lebih realistis karena semua komponen proyek dapat diestimasi dengan perkiraan secara fisik. Dalam proyek *software* estimasi biaya dan usaha proyek mempunyai kesulitan tersendiri karena karakteristik-karakteristik *software* yang lain dengan proyek fisik. Kesulitan-kesulitan yang sering dihadapi dalam estimasi proyek *software* sangat berkaitan dengan sifat alami *software* khususnya kompleksitas dan invisibilitas (keabstrakan). Selain itu pengembangan *software* merupakan kegiatan yang lebih banyak dilakukan secara intensif oleh manusia sehingga tidak dapat diperlakukan secara mekanistik murni. Kesulitan-kesulitan lainnya adalah (Hughes, 1999):

- a. *Novel application of software* artinya dalam rekayasa proyek tradisional, suatu system dapat dikonstruksi dengan system sebelumnya yang serupa tetapi dalam lokasi dan customer yang berbeda. Sehingga dapat dilakukan estimasi proyek berdasarkan pengalaman sebelumnya. Dalam proyek *software* akan mempunyai produk yang unik sehingga akan menimbulkan ketidakpastian estimasi.
- b. *Changing technology*, Untuk mengikuti perkembangan teknologi, maka suatu *software* aplikasi yang sama dapat diimplementasikan dalam lingkungan yang berbeda sehingga akan mempunyai estimasi proyek yang berbeda.
- c. *Lack of homogeneity of project experience*, untuk mendapatkan estimasi proyek yang efektif harus didasarkan pada informasi bagaimana proyek-proyek sebelumnya dilakukan.

Estimasi biaya dan usaha proyek merupakan suatu kegiatan pengaturan sumber daya dalam mencapai tujuan dan sasaran dari proyek, sehingga proyek dapat berjalan sesuai dengan tahapan dan target yang dikehendaki. Dalam usaha estimasi sering menghadapi dua permasalahan yaitu *over-estimates* dan *under-estimates*. *Over-estimates* (estimasi berlebihan) akan menimbulkan penambahan alokasi sumberdaya dari yang dibutuhkan sehingga akan meningkatkan penanganan managerial. Sedangkan estimasi yang kurang (*under-estimates*) akan mengurangi kualitas dari produk karena tidak sesuai dengan standar. Untuk itu perlu dilakukan langkah yang hati hati dalam melakukan estimasi suatu proyek *software* sehingga dapat dicapai keberhasilan proyek yaitu tepat waktu, sesuai budget dan terpenuhinya standar kualitas produk.

1.2 Tujuan

Penelitian ini dikembangkan dengan tujuan:

- a. Membuat model system estimasi biaya dan usaha proyek pengembangan *software* yang sesuai dengan kondisi dalam negeri.
- b. Tersedianya alat bantu manajemen proyek *software* dalam melakukan estimasi biaya dan usaha pengembangan *software* yang dapat diterapkan oleh *software* manager.

1.3 Manfaat

Manfaat dari penelitian ini adalah:

- a. Mendapatkan gambaran umum dan permasalahan yang dihadapi pengembang *software* dalam melakukan manajemen proyek *software*
- b. Tersedianya model estimasi biaya dan usaha pengembangan *software*.

2. STUDI PUSTAKA

2.1 Manajemen proyek

Secara umum, management dapat dipandang sebagai proses untuk mencapai tujuan dari sebuah system dan kemudian memonitor system untuk melihat kebenaran dari performa. Walaupun kebanyakan proyek manajemen secara umum dapat diaplikasikan pada pengembangan perangkat lunak, menurut Fred Brook terdapat karakteristik tertentu yang membuat manajemen pengembangan perangkat lunak berbeda [Hughes, B. Cotterell, M (1999)].

2.1.1 Aktifitas dalam manajemen proyek

Cukup sulit untuk menggambarkan semua aktifitas yang dilakukan dalam manajemen proyek karena perbedaan yang ada pada setiap proyek, akan tetapi aktifitas umum yang sering dilakukan dalam manajemen proyek menurut Sommerville adalah:

1. Sebelum implementasi (Pre Implementation)
 - Pengumpulan kebutuhan dan studi kelayakan proposal
 - Perencanaan dan penjadwalan
 - Estimasi biaya proyek
 - Seleksi personel dan evaluasi
2. Setelah implementasi (Post Implementation):
 - Penulisan laporan dan presentasi
3. Selama proyek berjalan (*Through out the Project - Umbrella Activity*):
 - Pengawasan aktifitas proyek dan review tindak lanjut

2.2 Metrik Software dan Pengukuran

Pengukuran merupakan dasar dari setiap disiplin rekayasa dan berlaku juga dalam perkerayaan perangkat lunak. Untuk mengevaluasi performa suatu system atau proses diperlukan suatu mekanisme untuk mengamati dan menentukan tingkat efisiensinya. Melalui pengukuran, maka akan diperoleh tingkat pencapaian di dalam proyek perangkat lunak yang sedang diamati.

Untuk setiap pengukuran yang dilakukan dibutuhkan tersedianya suatu ukuran kuantitatif yang disebut metrik. Istilah ukuran, pengukuran dan metrik sering digunakan secara bergantian walaupun sebenarnya masing-masing mempunyai pengertian yang berbeda. Metrik berdasarkan istilah rekayasa perangkat lunak didefinisikan sebagai sebuah ukuran kuantitatif yang dimiliki oleh suatu system, komponen atau proses tertentu dengan *attribute-attribute* yang diberikan.

2.2.1 Metrik Software

Ukuran merupakan faktor utama untuk menentukan biaya, penjadwalan, dan usaha. Kegagalan dari perkiraan ukuran yang tepat akan mengakibatkan penggunaan biaya yang berlebih atau keterlambatan penyelesaian proyek.

Estimasi ukuran *software* merupakan suatu aktifitas yang kompleks dan sukar berdasarkan pada beberapa alasan seperti kemampuan programmer,

factor lingkungan dan sebagainya. Tetapi karena tindakan ini harus dilakukan dan untuk mendapatkannya dengan mengukur ukuran proyek menggunakan ukuran seperti jumlah baris program (*Source lines of code/SLOC*) dan *Function Points*.

a. Ukuran jumlah baris program (SLOC)

SLOC merupakan ukuran yang kurang akurat dan merupakan sebuah topik yang menimbulkan perdebatan selama bertahun-tahun, dipandang sebagai sebuah ukuran untuk mengestimasi biaya dan waktu, tidak dapat dipastikan bahwa dua program yang mempunyai SLOC sama akan membutuhkan waktu implementasi yang sama walaupun keduanya diimplementasikan dengan kondisi pemrograman yang standard. Meskipun metode ini kurang akurat dan merupakan metodologi yang belum diterima secara luas, tetapi metrik dengan orientasi ukuran ini merupakan kunci pengukuran dan banyak estimasi *software* yang menggunakan model ini.

Secara virtual tidak mungkin untuk menghitung SLOC dari dokumen requirement awal. SLOC pengukurannya didasarkan pada bahasa pemrograman tertentu, oleh karena itu muncul banyak masalah dalam membuat standard pengukuran dengan teknik SLOC. Ukuran lain yang ada untuk mengukur besaran *software* adalah ukuran yang berorientasi fungsi dan ukuran yang berorientasi object. Metode ini merupakan metode yang lebih konsisten dan diterima secara luas.

b. Metrik yang berorientasi fungsi (*Function Point*)

Pendekatan yang berorientasi fungsi mengukur fungsionalitas aplikasi untuk mengestimasi ukuran *software* dan selanjutnya digunakan untuk estimasi biaya dan usaha yang diperlukan untuk mengembangkan system. Pendekatan ini diusulkan oleh Albrecht yang disebut sebagai metrik *Function Points*. Metrik ini diperoleh dari keterhubungan dasar antara domain informasi *software* dan kompleksitas *software* (Gambar 1). *Function Points* biasanya digunakan dalam mengukur system informasi manajemen (SIM).

Pada metodologi ini *software* dapat diklasifikasikan menjadi 5 domain yaitu:

- Jumlah data input pengguna
- Jumlah data output pengguna
- Jumlah data permintaan pengguna
- Jumlah file
- Jumlah file interface luar

Kemudian hitung nilai fungsi proyek yang mungkin pada setiap katagori dan kemudian setiap nilai perhitungan dikalikan dengan factor kompleksitas sebagai berikut:

- Sederhana (*simple*)
- Rata-rata (*average*)
- Komplek (*complex*)

Untuk menghitung *Unadjusted Function Points* digunakan tabel berikut berdasarkan kriteria dari setiap kategori.

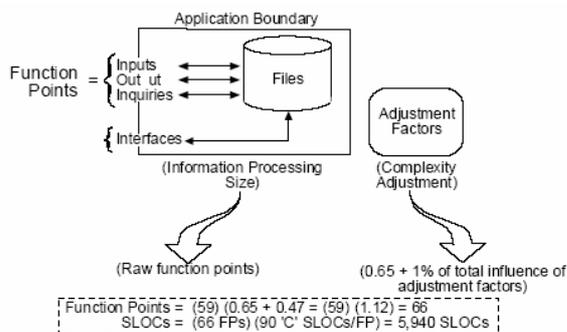
Tabel 1. Tabel analisis *Function Points*

<i>FP Analysis</i>	<i>Simple</i>	<i>Average</i>	<i>Complex</i>	<i>Total</i>
Inputs	3x` `	4x` `	6x` `	
Outputs	4x` `	5x` `	7x` `	
Inquiries	3x` `	4x` `	6x` `	
Files	7x` `	10x` `	15x` `	
Interfaces	5x` `	7x` `	10x` `	

Untuk menghitung function point digunakan persamaan berikut:

$$FP = \text{count total} * [0.65 * 0.01 * \text{sum}(F_j)]$$

dimana count total adalah total yang diperoleh dari table *function point* analisis sum(Fj) adalah jumlah dari 14 faktor kompleksitas yang bernilai 0 s/d 5.



Gambar 1. Analisis *Function Point*

3. PENDEKATAN DAN VERIFIKASI MODEL/SISTEM

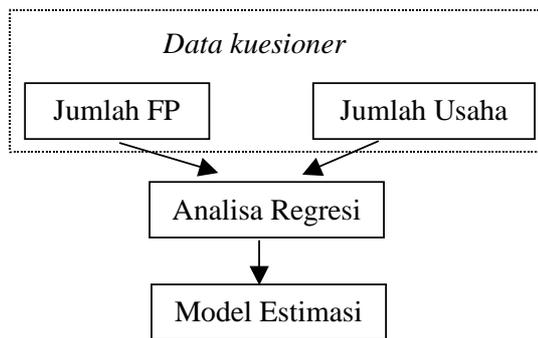
3.1 Pendekatan Model

Pendekatan model yang digunakan dalam menghitung besaran proyek adalah model function point (FP). Dibandingkan dengan pendekatan berbasis ukuran baris (LOC/Line Of Code). Pendekatan FP lebih independen terhadap bahasa pemrograman sehingga bisa diterapkan pada jenis aplikasi yang berbeda baik aplikasi database yang non-procedural, sistem informasi berbasis web, maupun aplikasi penghitungan, misalnya payroll. Pendekatan ini juga lebih mudah diprediksi daripada LOC karena parameternya dihitung berdasarkan data yang lebih diketahui, misalnya prediksi jumlah input dan output.

Meskipun FP dianggap memiliki kelemahan dalam subyektifitas data yang dimasukkan tetapi beberapa kriteria, misalnya untuk menentukan kategori sederhana atau kompleks, telah ditetapkan secara numerik untuk lebih memastikan obyektivitas data. Disamping itu, hasil perhitungan FP juga sering dianggap tidak memiliki arti yang mudah dipahami dibandingkan dengan LOC yang besarnya menunjukkan jumlah ukuran coding. Akan tetapi,

hasil akhir FP dapat dikonversikan ke dalam LOC berdasarkan jenis bahasa pemrograman yang dipakai.

Untuk mendapatkan model estimasi (Gambar 2), dilakukan analisa regresi linier terhadap sample hasil survey yang menghasilkan jumlah FP dan jumlah usaha dari suatu proyek pengembangan *software* yang dilakukan oleh beberapa *software* house. Kuesioner diisi dengan cara in-depth interview maupun dengan menyediakan fasilitas pengisian secara online (<http://www-proyeksoftware.com>). Dari kuesioner tersebut didapatkan jumlah function point dan jumlah usaha untuk mengerjakan suatu proyek *software*.



Gambar 2. Langkah pemodelan

Jumlah usaha didapatkan dari waktu pengerjaan dikalikan dengan jumlah personel yang terlibat dalam pengerjaan proyek. Analisa regresi linear selanjutnya dilakukan terhadap beberapa sampel jumlah FP terhadap jumlah usaha/effort. Dari analisa tersebut akan didapatkan suatu persamaan, yakni:

$$\text{Effort} = a + b * \text{FP}$$

Konstanta a dan b yang didapat tersebut akan menjadi model bagi penentuan usaha (i.e jumlah biaya dan personel yang diperlukan) jika diketahui jumlah function point yang dapat dihitung dari kebutuhan pengguna (requirement definition).

3.2 Verifikasi Model

Verifikasi terhadap validitas model yang dihasilkan dapat diketahui dari sampel data yang masuk, tingkat kesalahan dalam regresi tingkat kesesuaian dengan model yang sudah ada. Perbedaan lingkungan pengembangan *software*, misalnya di negara maju dan di Indonesia (khususnya Jakarta), tentu akan menyebabkan persamaan yang berbeda. Kemungkinan perbedaan itu jugalah yang mendorong dilakukannya kajian ini sehingga diperoleh suatu model yang paling sesuai untuk daerah/lingkungan tertentu.

Verifikasi dengan berdasar pada jumlah sampel sering disebut sebagai exhaustive testing. Cara ini dilakukan dengan mengambil sampel atau kasus sebanyak mungkin untuk menguji disain dan implementasi sebuah sistem. Pendekatan ini diharapkan dapat mendeteksi kesalahan lebih akurat.

Pendekatan ini membutuhkan semua kombinasi (*exhaustive test*) agar disain atau implementasi dapat dijamin benar. Untuk sistem yang besar dan kompleks, hal ini tentu saja tidak dapat dilakukan. Pendekatan formal method, yang menggunakan pembuktian secara matematis, biasanya digunakan untuk menguji sistem dengan tingkat kemungkinan yang tinggi. Akan tetapi untuk sample yang kecil (i.e tidak melebihi ribuan) seperti dalam kajian ini, *exhaustive test* ini dapat dilakukan.

4. PENGEMBANGAN DAN KINERJA MODEL

4.1 Pengembangan Model

Model yang dikembangkan dalam kajian ini meliputi model estimasi besaran usaha pengembangan proyek dengan pendekatan function point dan alat bantu berupa *software* untuk memasukkan nilai parameter function point tersebut dan menampilkan model yang dihasilkan.

4.1.1 Analisa Data Hasil Observasi

Dari hasil pengumpulan data selama empat bulan, diperoleh data sebanyak 34 data observasi. Dari data observasi ini kemudian dianalisa untuk membuat model estimasi dengan berdasarkan pada metrik *Function Points*.

Untuk memastikan bahwa data yang telah diperoleh adalah data yang berisitribusi normal, maka dilakukan analisa descriptive terhadap semua variabel data hasil observasi. Hasil analisa

deskriptive untuk data effort atau usaha pengembangan *software* diperoleh dari hasil perkalian antara lama pengembangan dalam bulan dengan jumlah orang yang digunakan dalam pengembangan *software*.

Data hasil observasi semua variabel yang digunakan untuk pembuatan model mempunyai nilai dekripsi statistik sebagaimana terlihat pada Tabel 2.

Dari Tabel 2 terlihat bahwa effort mempunyai nilai rata-rata 41,51 OB dengan effort minimum 8 OB dan effort maksimum 72 OB. Sedangkan biaya (cost) yang digunakan untuk pengembangan *software* dari semua data hasil observasi mempunyai nilai rata-rata 92,6 juta dengan nilai minimum biaya proyek yang dihabiskan adalah 1,5 juta dan nilai maksimum biaya yang diperoleh adalah 500 juta. Ukuran metrik function point dari data hasil observasi mempunyai nilai rata-rata 214,85 dengan nilai minimum function point yang dikembangkan adalah sebesar 19,55 dan nilai maksimum function point hasil observasi adalah 348,48.

4.1.2 Pembuatan Model Estimasi

Untuk pembuatan model estimasi biaya dan usaha proyek pengembangan *software* pertama-tama dilakukan analisa parameter yang berpengaruh terhadap kedua variabel tersebut. Untuk menguji keterkaitan atau pengaruh dari variabel, digunakan perhitungan nilai korelasi dari setiap variabel yang di analisa. Adapun tabel korelasi dari semua variabel hasil observasi adalah:

Tabel 2. deskripsi statistik semua variabel data hasil observasi

<i>Variabel</i>	<i>Valid N</i>	<i>Mean</i>	<i>Minimum</i>	<i>Maximum</i>	<i>Std.Dev.</i>	<i>Standard error</i>
Effort	29	41,51	8,00	72,00	196,952	365,731
Biaya (cost)	29	92,60	1,50	500,0	1,008,569	1,872,866
Function Point	29	214,85	19,55	348,48	928,059	1,723,363
Function Count	29	185,86	22,00	264,00	736,291	1,367,259
Total Faktor Kompleksitas	29	48,00	20,00	67,00	126,152	234,258
AFC	29	1,13	0,85	1,32	0,1262	0,02343

Tabel 3. Nilai korelasi antar variabel

<i>Variabel</i>	<i>Effort</i>	<i>Biaya (cost)</i>	<i>Function Point</i>	<i>Function Count</i>	<i>Tot Fak Kompl</i>	<i>PCA</i>
Effort	1,00	0,09	0,12	0,09	0,22	0,22
Biaya (cost)	0,09	1,00	0,38	0,42	0,12	0,12
Function Point	0,12	0,38	1,00	0,97	0,70	0,70
Function Count	0,09	0,42	0,97	1,00	0,54	0,54
TotFakt Kompleks	0,22	0,12	0,70	0,54	1,00	1,00
PCA	0,22	0,12	0,70	0,54	1,00	1,00

Dari Tabel 3 terlihat bahwa nilai korelasi antara effort dan function point bernilai 0,12, sedangkan korelasi antara effort dengan total faktor kompleksitas bernilai 0,22. Dari nilai korelasi ini dapat disimpulkan bahwa nilai usaha (effort) proyek pengembangan *software* dipengaruhi oleh nilai besaran function point dan tingkat kompleksitas proyek *software*. Artinya semakin tinggi nilai

function point dan tingkat kompleksitas proyek *software* akan membutuhkan effort yang semakin tinggi pula.

Hubungan antara biaya (cost) dapat dimodelkan dengan grafik eksponensial. Artinya nilai peningkatan biaya yang dibutuhkan proyek pengembangan *software* bertambah secara eksponensial terhadap penambahan besaran function

point dari proyek *software* yang akan dikembangkan. Adapun model eksponential yang diperoleh dari analisa data hasil observasi adalah:

$$\text{Biaya (cost)} = 8,0757 * \exp(0.0087 * \text{FP})$$

Dimana FP adalah function point dari proyek *software* yang akan dikembangkan. Secara linier regresi dapat direpresentasikan keterhubungan tersebut sebagai rumus:

$$\text{Biaya (cost)} = 3,7076 + 0,4138 * \text{FP}$$

4.2 Kinerja Model

Model yang telah diperoleh perlu diuji coba dengan data-data kasus proyek yang serupa untuk mendapatkan tingkat kesahihan analisa data dan penggunaan model. Untuk menguji validitas model yang dibuat digunakan metode uji adjusted R2, standard deviasi estimasi dan prediksi pada tingkat L (Pred(L)). Adjusted R2 adalah koefisien dari nilai R2 yang diperoleh dari hasil observasi dan nilai dari hasil prediksi. Sedangkan standard deviasi estimasi adalah variasi nilai yang menunjukkan tingkat kesalahan prediksi, yang dihitung berdasarkan selisih antara usaha yang digunakan secara real dalam proyek *software* dengan nilai estimasi usaha dari model. PRED(L) adalah persentasi nilai estimasi pada L persen nilai aktual. Sebagai contoh PRED(0,25) adalah presentasi estimasi dalam 25% nilai aktual.

Adapun hasil pengujiannya dapat diperlihatkan dengan tabel sebagai berikut:

Tabel 4. validasi model estimasi Usaha (Effort)

<i>Kriteria</i>	<i>Sebelum pemodelan</i>	<i>Setelah pemodelan</i>
Adj_R ²	0,115	0,115
Standard Deviasi	19,762	19,564
PRED(0,20)	-2%	26%
PRED(0,25)	-2%	33%
PRED(0,30)	-3%	39%

Dari tabel di atas terlihat bahwa nilai korelasi antara model estimasi yang dikembangkan dengan kondisi lokal sama dengan model yang diperoleh dari literatur yaitu model FP-Albrecht. Namun nilai standard deviasi model yang dikembangkan lebih kecil dari pada nilai standard deviasi dari model Albrecht, selain itu nilai prosentase tingkat prediksi model yang dikembangkan dari kondisi lokal lebih tinggi dari pada prosentase nilai model dari literatur. Oleh karena itu dapat disimpulkan bahwa tingkat akurasi estimasi usaha proyek pengembangan *software* dengan model yang dikembangkan berdasarkan kondisi lokal lebih tinggi dari pada model yang diperoleh dari literatur.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil analisa data dan pembuatan model estimasi biaya dan estimasi usaha proyek pengembangan *software* dapat disimpulkan hal-hal sebagai berikut:

1. Proyek *software* hasil observasi mempunyai tingkat kompleksitas yang relatif tinggi serta menggunakan ukuran metrik function point yang cenderung besar dibandingkan dengan biaya yang dialokasikan.
2. Proyek *software* hasil observasi menggunakan dana atau biaya penyelesaian proyek yang relatif kecil atau cenderung kecil jika dibandingkan dengan besaran ukuran *software* yang dikembangkan, hal ini menunjukkan bahwa *software* house hasil observasi belum mengestimasi biaya pengembangan *software* secara real sesuai ukuran *software*.
3. Kesulitan yang sering dihadapi oleh *software* house dalam pelaksanaan proyek *software* adalah sering adanya perubahan user requirement atau kebutuhan user yang sering berubah-ubah.
4. Dari hasil perhitungan nilai korelasi menunjukkan bahwa ada keterkaitan atau hubungan antara biaya atau usaha proyek pengembangan *software* dengan function point metrik ukuran besaran *software*.
5. Model estimasi biaya pengembangan *software* yang diperoleh dari hasil observasi mempunyai bentuk model eksponensial, sedangkan model estimasi usaha modelnya cenderung berbentuk linier.
6. Telah dilakukan perancangan dan implementasi *software* estimasi biaya dan usaha proyek *software* dengan dasar perhitungan ukuran metrik function point, sehingga dapat digunakan untuk estimasi biaya dan usaha proyek *software* pada tahap requirement analisis.

5.2 Saran

Adapun saran-saran yang perlu ditindak lanjuti adalah sebagai berikut:

1. Pada tahap pengumpulan data dengan penyebaran kuisioner ke berbagai *software* house yang ada sering menemui hambatan yaitu *software* house tidak mau mengisi kuisioner yang berkaitan dengan biaya proyek pengembangan *software* karena data tersebut menyangkut rahasia perusahaan, oleh karena itu perlu dilakukan pemahaman yang baik akan pentingnya penelitian manajemen proyek *software* yang tidak digunakan untuk hal-hal yang bersifat komersial
2. Data hasil observasi yang diperoleh masih sangat terbatas sehingga diperoleh model yang kurang optimal, maka untuk mengoptimalkan dan meningkatkan keakuratan model estimasi

yang dibuat perlu dilakukan updating data terus menerus sehingga model menggunakan data yang cukup banyak dan sesuai dengan teknologi informasi yang terus berkembang.

3. Untuk setiap *software house* mempunyai karakteristik-karakteristik tertentu yang spesifik dalam pengembangan *software*, sehingga dalam melakukan estimasi biaya proyek *software* membutuhkan data masa lalu yang spesifik pula. Oleh karena itu perlu dikembangkan *software* estimasi yang dapat dibuat untuk memodelkan estimasi biaya proyek dengan inputan data dari pengalaman masa lalu *software house* tertentu sehingga akan diperoleh model yang sesuai dengan karakteristiknya dan mendapatkan model estimasi yang lebih sesuai dengan kebutuhannya dan mempunyai tingkat keakuratan yang lebih tinggi.

http://www.simula.no/photo/expertsubmitnoveember2002_copy.pdf

- [13] Pressman, Roger S., *Software Engineering: A Practioner's Approach*, 5th Ed., MC Graw - Hill, New York, 2001
- [14] Sommerville, I., *Software Engineering*, sixth edition, Addison Wesley, 2001.

DAFTAR PUSTAKA

- [1] A.J.Cowling, *Lecture Notee: Software Measurement and Testing*, <http://www.dcs.shef.ac.uk/ajc>
- [2] Barry W. Boehm and Richard E. Fairley, *Software Estimation Perspectives*, <http://www.computer.org/software/so2000/pdf/s6022.pdf>
- [3] Callahan, John and Sabolish, George A., *Process Improvement Model for Software Verification and Validation*, IV&V Facility West Virginia University (http://www.qaiindia.com/Resources_Art/journal_improvment.htm)
- [4] Capers, J., *What is Function Points*, <http://www.spr.com/library/0funcmet.htm>
- [5] Chockalingam, A., *Estimos: A Metrics Management and Schedule Planning Plug-in*, *M.Sc. Thesis*, University of Sheffield, 2004.
- [6] Chulani, S, *Software Development Cost Estimation Approaches—A Survey*, *PhD Thesis*, University of Southern California, 1998.
- [7] Gray, Clifford F. and Larson, Erik W., *Project Management: The Managerial Process*, McGraw-Hill, 1st Ed., Singapore, 2000.
- [8] Hughes, Bob and Cotterell, Mike, *Software Project Management*, 2nd eEd., McGraw-Hill, London, 1999.
- [9] J.P. Lewis, *Large Limits to Software Estimation*, <http://www.idiom.com/~zilla/Work/kcsest.pdf>
- [10] Kathleen Peters, *Software Project Estimation*, <http://www.spc.ca/downloads/resources/estimate/estbasics.pdf>
- [11] Leung, H., Fan, Zhang, *Software Cost Estimation*, <http://paginaspersonales.deusto.es/cortazar/doctorado/articulos/leung-handbook.pdf>, 2000.
- [12] M. Jorgensen, *A Review of Studies on Expert Estimation of Software Development Effort*,