

## **PENGUNAAN MODEL *FUNCTION POINT* DALAM ESTIMASI BIAYA DAN USAHA PROYEK PENGEMBANGAN *SOFTWARE* SISTEM INFORMASI BISNIS**

Suharjito\*, Budi Prasetyo\*\*

### **ABSTRAK**

**PENGUNAAN MODEL *FUNCTION POINT* DALAM ESTIMASI BIAYA DAN USAHA PROYEK PENGEMBANGAN *SOFTWARE* SISTEM INFORMASI BISNIS.** Tujuan penelitian ini adalah untuk pengembangan *software* sistem estimasi biaya dan usaha untuk proyek pengembangan *software* yang sesuai dengan kondisi di Indonesia. Metodologi yang digunakan dalam penelitian ini adalah metode analisis dan metode skema. Dalam tahap analisis dilakukan pengumpulan dan pengolahan informasi yang didapatkan dari penyebaran kuisioner ke beberapa perusahaan pengembang *software* di Indonesia. Tahap berikutnya adalah perancangan diagram *UML (Unified Modelling Language)*, perancangan antar muka, struktur menu dan fungsionalitas sistem. Hasil dari penelitian ini adalah paket aplikasi berbasis *Windows* yang dapat digunakan untuk mengestimasi biaya dan usaha pada proyek pengembangan *software* sistem informasi. Kesimpulan yang dapat ditarik dari penelitian ini adalah aplikasi berbasis *Windows* ini dapat diterapkan untuk mengestimasi biaya dan usaha proyek pengembangan *software* sistem informasi dengan menggunakan alat ukur (*metric*) *function points*.

Kata-kata kunci: Estimasi, Usaha, Biaya, Analisis *Function Point*, Analisis *Fuzzy Function Point* dari Proyek *Software*, Pemodelan Biaya.

### **ABSTRACT**

**COST AND EFFORT ESTIMATION FOR DEVELOPING OF BUSSINESS INFORMATION SYSTEM SOFTWARE USING FUNCTION POINT MODEL.** The purpose of this research is to devising a system estimate expense and effort development project of software windows based witch matching with condition in the country. The method of the research used is the analysis method and the scheme method. In analysis method taken is collecting and analyzing information, which got from questionnaires to some company of software development in Indonesia. While in scheme method taken is designing *UML (Unified Modeling Language)* diagrams, user interfaces, structure menus and the content functionality system. The result of this research is an application of windows bases, which can be used to estimate expense and effort that is needed in developing a business information system software project. The conclusion is this windows bases application applicable to estimate expense and effort project of software with characteristic of information system by using function points metric.

Keywords: Estimate, Effort, Expense, Function Point Analysis, Fuzzy Function Point Analysis of Software Project, Cost Modeling.

---

\* Staff PTA, TAB, BPP Teknologi, E-mail: harjito@webmail.bppt.go.id

\*\* Pusat Pengembangan Informatika Nuklir - BATAN, E-mail: budipras@batan.go.id

## PENDAHULUAN

### 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 [9]:

1. *Novel application of software* artinya dalam rekayasa proyek tradisional, suatu sistem dapat dikonstruksi dengan sistem 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.
2. *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.
3. *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.

Barry Boehm [2] telah mengidentifikasi beberapa metode estimasi biaya dan usaha proyek pengembangan *software* sebagai berikut: Model algoritmik, Analogi, Pendapat pakar, Parkinson, *Top-down*, dan *Bottom-up*. Dalam penelitian ini akan

dikembangkan metode estimasi parametrik berdasarkan karakteristik-karakteristik dari ukuran proyek *software* yaitu *function point* dan *object point* serta KLOC (*Kilo line of Code*). Metode estimasi parametrik yang sering digunakan saat ini adalah dengan menggunakan metode COCOMO yang tidak sesuai jika diterapkan untuk estimasi proyek *software* di Indonesia. Oleh karena itu dalam penelitian ini akan dikembangkan metode tersebut yang disesuaikan dengan data-data dan informasi pengembangan *software* dalam negeri, sehingga diharapkan dapat diperoleh parameter yang mempunyai tingkat validitas estimasi yang lebih tinggi.

## METHODOLOGI

Metode yang dilakukan dalam penelitian ini adalah:

- Metode Analisa dan pemodelan sistem
  1. Studi Lapangan  
Studi lapangan dilakukan dengan mengamati masalah-masalah aktual dalam proyek-proyek pengembangan piranti lunak (*software*) di Indonesia dan mengumpulkan data-data yang dibutuhkan untuk melakukan penelitian dari perusahaan-perusahaan pengembang *software*.
  2. Studi Kepustakaan  
Studi kepustakaan dilakukan dengan mencari sumber-sumber referensi yang berkaitan dengan penelitian yang dilakukan. Sumber-sumber referensi tersebut diperoleh melalui buku, artikel, dan jurnal *online* dari internet.
  3. Metode Survei  
Survei dilakukan dengan menggunakan kuesioner yang diberikan kepada responden secara langsung dan disediakan melalui situs *web*.
  4. Analisa dan pemodelan  
Data hasil survei dikodekan dan dianalisa untuk membuat model estimasi dengan metode regresi dan korelasi antar parameter.
- Metode Perancangan
  1. Perancangan Tampilan Layar  
Tampilan layar dirancang sesuai dengan input dan output dari sistem dan dikembangkan agar mudah digunakan dan dipelajari.
  2. Perancangan Basis Data  
Perancangan basis data dilakukan untuk membuat data *input* dan *output* terstruktur dengan baik dan memudahkan modifikasi dan pengaksesan data.

3. Perancangan Model Sistem dengan *Unified Modeling Language* (UML)
4. Sistem dikembangkan dengan menggunakan *methodology* pengembangan sistem yang berorientasi *object* dan dirancang dengan UML untuk merepresetasikan diagram statis dan dinamis dari sistem.

### **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*, faktor 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*.

### **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 diimplemenatsikan dengan kondisi pemrograman yang standar. 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 standar 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.

### **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 sistem. Pendekatan ini diusulkan oleh

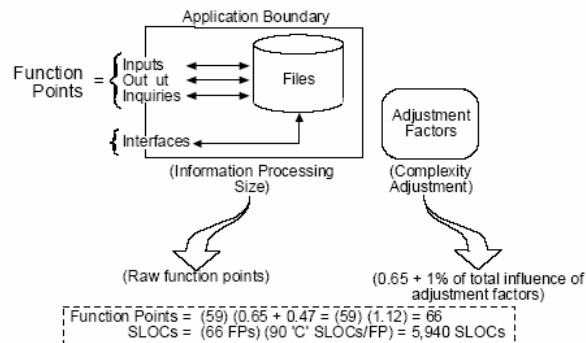
Albrecht yang disebut sebagai metrik *Function Points*. Metrik ini diperoleh dari keterhubungan dasar antara domain informasi *software* dan kompleksitas *software* (Gambar 1). *Function Points* biasanya digunakan dalam mengukur sistem informasi manajemen (SIM).

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 faktor kompleksitas sebagai berikut:

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



Gambar 1. Analisis *Function Point*

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

Table 1. Table analisis *Function Points*

FP Analysis	Simple	Average	Complex	Total
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:

$$FP = \text{count total} * [0.65 * 0.01 * \text{sum}(F_j)] \quad (1)$$

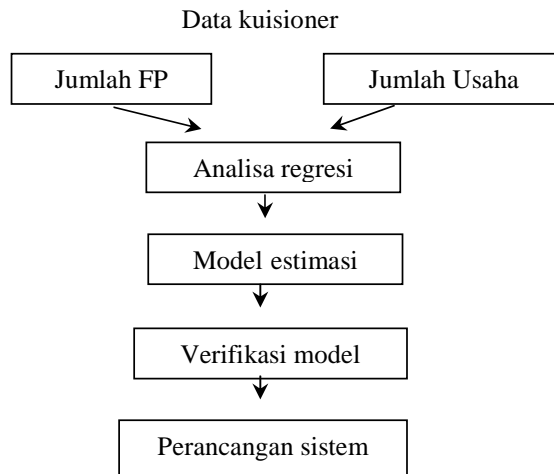
Di mana count total adalah total yang diperoleh dari table *function point* analisis sum( $F_j$ ) adalah jumlah dari 14 faktor kompleksitas yang bernilai 0 s/d 5.

## 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 subjektivitas 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 analisis 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. Analisis regresi linier selanjutnya dilakukan terhadap beberapa sampel jumlah FP terhadap jumlah usaha/*effort*. Dari analisa tersebut akan didapatkan suatu persamaan, yakni:

$$Effort = a + b * FP \quad (2)$$

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

## 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 (tidak melebihi ribuan) seperti dalam kajian ini, *exhaustive test* ini dapat dilakukan.

## **HASIL DAN PEMBAHASAN**

### **Pengembangan Model**

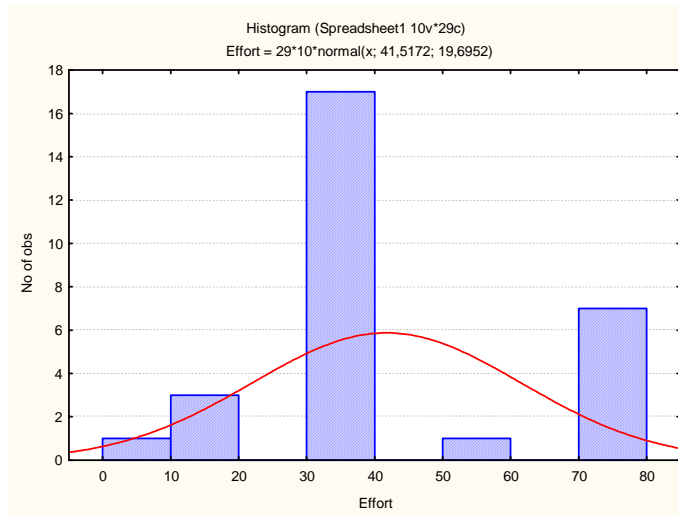
Model yang dikembangkan dalam kajian ini meliputi model estimasi besaran usaha pengembangan proyek dengan pendekatan *function point* dan alat bantu berupa *software* untuk memasukkan nilai parameter *function point* tersebut dan menampilkan model yang dihasilkan.

### **Analisis Data Hasil Observasi**

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

Untuk memastikan bahwa data yang telah diperoleh adalah data yang berdistribusi 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*. Adapun distribusi dari data *effort* adalah sebagai berikut:





Gambar 3. Distribusi data Usaha (Effort)

Data hasil observasi semua variabel yang digunakan untuk pembuatan model mempunyai nilai dekripsi statistik sebagai berikut: Data *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 rupiah 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.

### 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 dianalisa. Adapun tabel korelasi dari semua variabel hasil observasi adalah

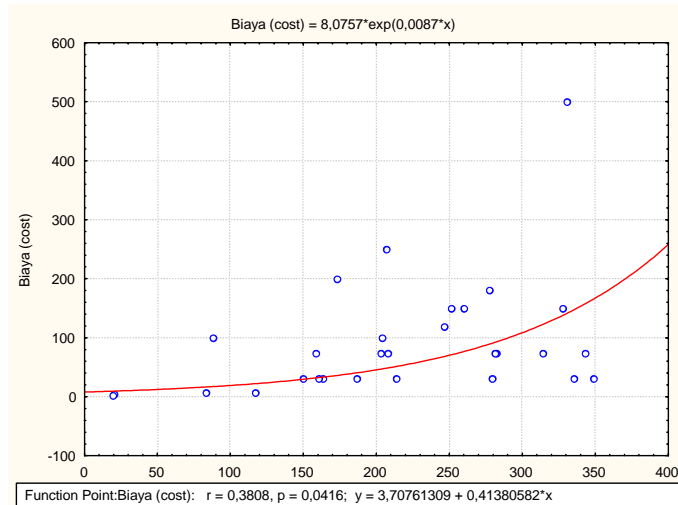
Tabel 2. Tabel Korelasi antar variabel

	<i>Effort</i>	Biaya ( <i>cost</i> )	<i>Function Point</i>	<i>Function Count</i>	Tot Fak Kompl	PCA
<i>Effort</i>	1,00	0,09	0,12	0,09	0,22	0,22
Biaya ( <i>cost</i> )	0,09	1,00	0,38	0,42	0,12	0,12
<i>Function Point</i>	0,12	0,38	1,00	0,97	0,70	0,70
<i>Function Count</i>	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 di atas 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.

Hal yang sama juga dapat dilihat tingkat keterkaitan antara variabel biaya dengan *function point* yang mempunyai nilai korelasi sebesar 0,38. artinya besaran *function point* dari suatu proyek pengembangan *software* akan sangat berpengaruh terhadap besaran biaya yang digunakan.

Adapun hasil pemodelan data biaya (*cost*) yang dikaitkan dengan *function point* (FP) adalah seperti gambar berikut:



Gambar 4. Grafik Model biaya (*cost*) dengan *Funtion Points*

Dari gambar di atas terlihat bahwa 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.

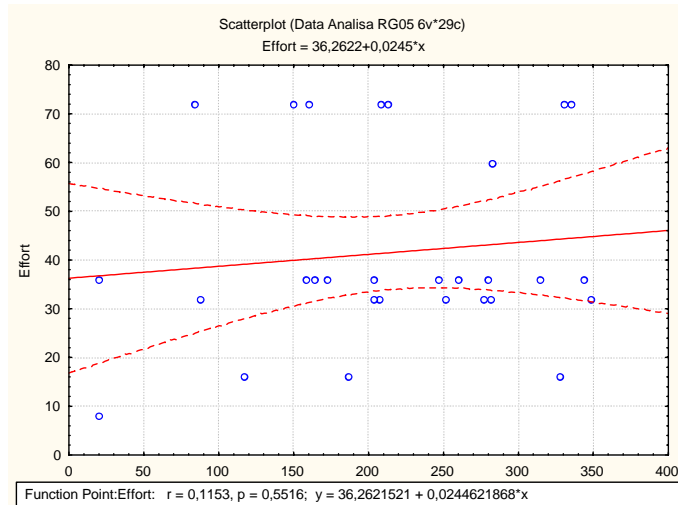
Adapun model eksponensial yang diperoleh dari analisa data hasil observasi adalah:

$$\text{Biaya (cost)} = 8,0757 \cdot \exp(0.0087 \cdot \text{FP}) \quad (3)$$

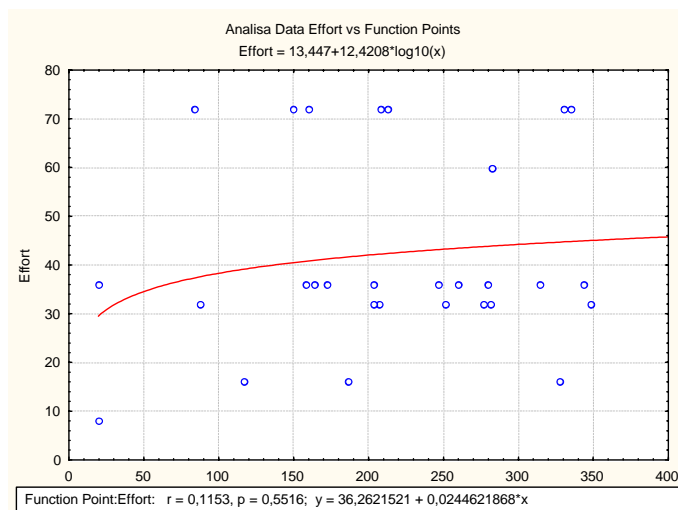
Di mana FP adalah *function point* dari proyek *software* yang akan dikembangkan. Secara linier regresi dapat direpresentasikan keterhubungan tersebut sebagai rumus:

$$\text{Biaya (cost)} = 3,7076 + 0,4138 \cdot \text{FP} \quad (4)$$

Sedangkan keterhubungan antara usaha (*effort*) dengan *function point* dapat diperlihatkan dengan beberapa model berikut:



Gambar 5. Grafik Model Usaha (*Effort*) dengan FP secara linier



Gambar 6. Grafik Model Usaha (*Effort*) dengan FP secara logaritmik

## 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* R<sup>2</sup>, standar deviasi estimasi dan prediksi pada tingkat L (Pred(L)). *Adjusted* R<sup>2</sup> adalah koefisien dari nilai R<sup>2</sup> yang diperoleh dari hasil observasi dan nilai dari hasil prediksi. 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 persentase 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:

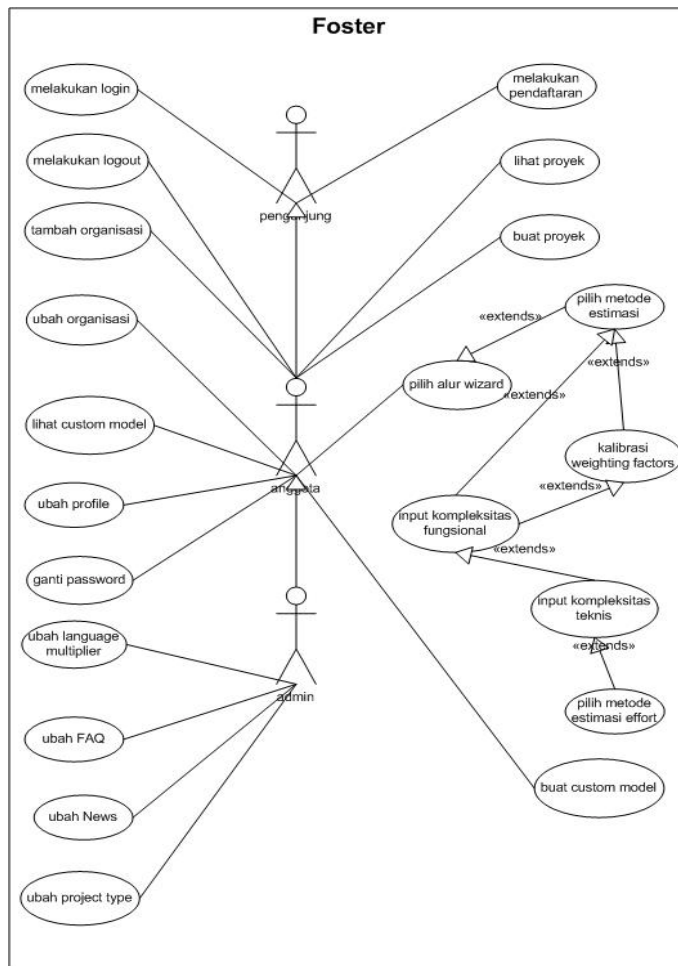
Tabel 3. validasi model estimasi Usaha (*Effort*)

Kriteria	Sebelum pemodelan	Setelah pemodelan
Adj_R <sup>2</sup>	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 standar deviasi model yang dikembangkan lebih kecil dari pada nilai standar 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.

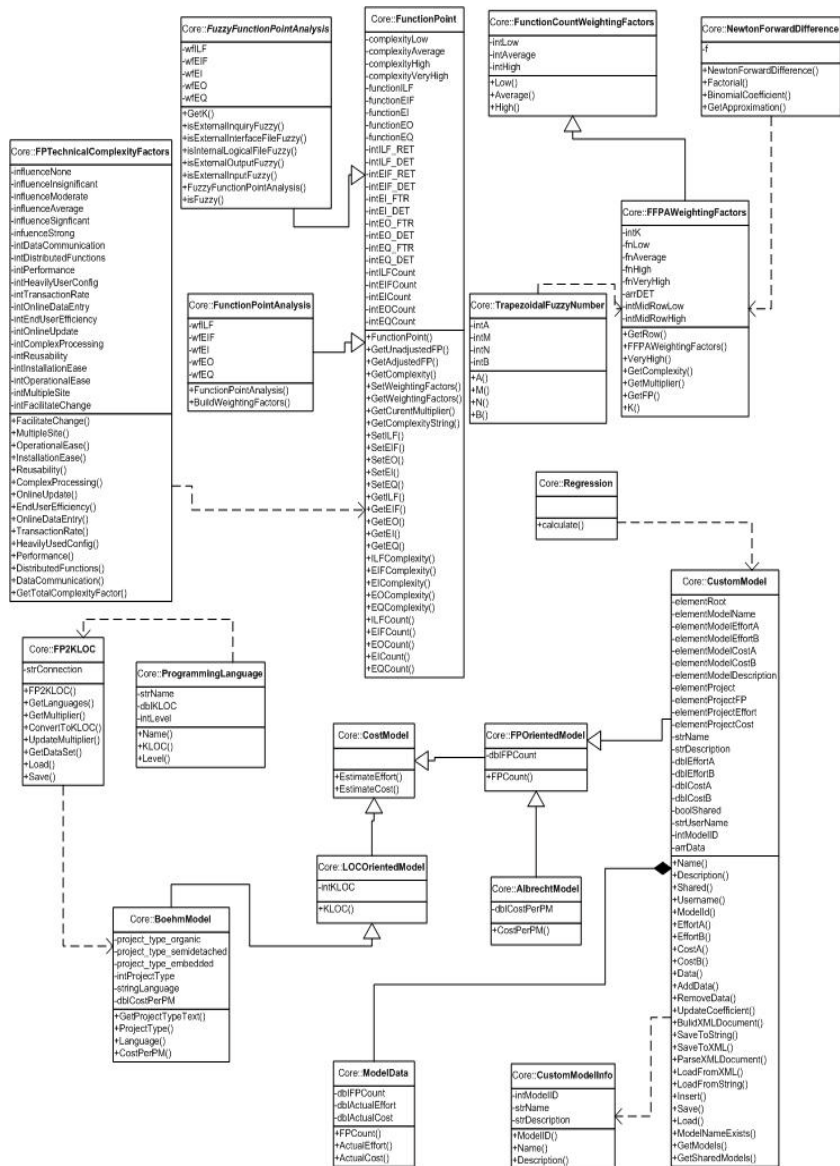
## Perancangan Sistem

Sistem yang dikembangkan dalam penelitian ini dirancang menggunakan pendekatan sistem berorientasi *object* dengan diagram *use case* dapat dilihat pada gambar 7:



Gambar 7. Use case diagram sistem

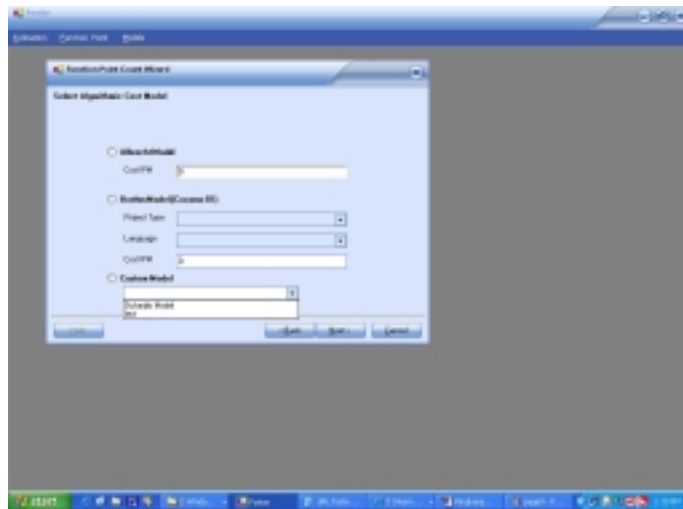
Sedangkan diagram statis sistem estimasi dapat digambarkan dengan *class diagram* berikut:



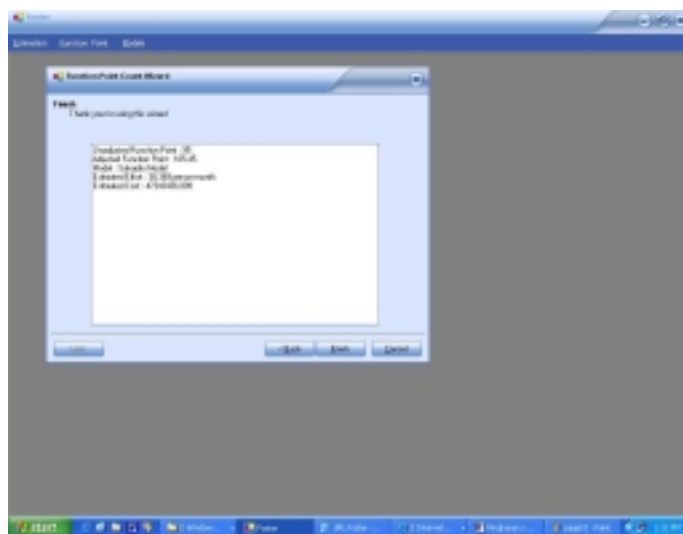
Gambar 8. Class diagram sistem







Gambar 11. Tampilan pilihan metode estimasi



Gambar 12. Tampilan hasil estimasi

## 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. Model estimasi biaya dan usaha pengembangan proyek *software* ini hanya sesuai digunakan untuk estimasi pengembangan *software* sistem informasi bisnis dengan metrik *function point*.
5. Model estimasi biaya pengembangan *software* yang diperoleh dari hasil observasi mempunyai bentuk model eksponensial, sedangkan model estimasi usaha modelnya cenderung berbentuk linier.
6. Sistem estimasi biaya proyek dapat digunakan bagi para pengembang *software* (*software developer*), manajer proyek, dan staf IT lainnya.
7. Sistem ini memungkinkan untuk melakukan estimasi suatu proyek secara kolaborasi baik dengan pengguna lain dari organisasi yang sama maupun dari luar organisasi dan dapat di akses secara *on-line* pada *website*: <http://www.proyeksoftware.com/fosterweb>

## 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. Sistem ini masih dikembangkan dengan metrik *function point* sehingga untuk pengembangan proyek *software* dengan pendekatan yang berorientasi *object* belum bisa digunakan. Oleh karena itu masih terbuka kemungkinan untuk mengembangkan sistem ini dengan pendekatan metrik yang berorientasi *object* seperti jumlah *class*, jumlah *object* dan lain-lain.
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.
4. Menganalisis hubungan antara tingkat kematangan suatu organisasi (CMM) dengan *effort* dan *cost* dari proyek yang dikerjakan sehingga lebih menajamkan hasil estimasi yang didapat.
5. Menambahkan pengali-pengali bahasa dengan konstanta pengali bahasa bahasa pemrograman baru.
6. Mengembangkan sistem yang memungkinkan estimasi proyek yang dapat melakukan estimasi tidak hanya dari secara empiris namun juga analogi.

## DAFTAR PUSTAKA

1. A.J.COWLING, *Lecture Note: Software Measurement and Testing*, <http://www.dcs.shef.ac.uk/ajc>
2. BARRY BOEHM, BRADFORD CLARK, ELLIS HOROWITZ, RAY MADACHY, RICHARD SHELBY, CHRIS WESTLAND, *Cost Models for Future Software Life Cycle Processes: COCOMO 2.0, Annals of Software Engineering*, (1995). [http://sunset.usc.edu/research/COCOMOII/Docs/C2ASE\\_submitted.pdf](http://sunset.usc.edu/research/COCOMOII/Docs/C2ASE_submitted.pdf)
3. BARRY W. BOEHM AND RICHARD E. FAIRLEY, *Software Estimation Perspectives*, <http://www.computer.org/software/so2000/pdf/s6022.pdf>
4. 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\\_improvement.htm](http://www.qaiindia.com/Resources_Art/journal_improvement.htm))

5. CAPERS, J., *What is Function Points*, <http://www.spr.com/library/0funcmet.htm>
6. CHOCKALINGAM, A., *Estimos: A Metrics Management and Schedule Planning Plug-in*, MSc Theses, University of Sheffield, (2004).
7. CHULANI, S., *Software Development Cost Estimation Approaches – A Survey*, PhD these, University of Southern California, (1998).
8. DAVE SRULOWITZ, MIKE BANDOR AND VIC HELBL, *Software Estimation*, [http://www.saspin.org/SASPIN\\_Apr2001\\_COCOMO.pdf](http://www.saspin.org/SASPIN_Apr2001_COCOMO.pdf)
9. DEVNANI, S., CLARK, B., BOEHM B., *Calibrating the COCOMO II Post-Architecture Model*, <http://sunset.usc.edu/publications/TECHRPTS/1998/uscscse98-502/CalPostArch.pdf>
10. GRAY, CLIFFORD F. AND LARSON, ERIK W., *Project Management: The Managerial Process*, McGraw-Hill, 1st Ed., Singapore, (2000)
11. HUGHES, BOB AND COTTERELL, MIKE, *Software Project Management*, 2nd eEd., McGraw-Hill, London, (1999)
12. J.P. LEWIS, *Large Limits to Software Estimation*, <http://www.idiom.com/~zilla/Work/kcsest.pdf>
13. KATHLEEN PETERS, *Software Project Estimation*, <http://www.spc.ca/downloads/resources/estimate/estbasics.pdf>
14. LEUNG, H., FAN, ZHANG, *Software Cost Estimation*, <http://www.paginaspersonales.deusto.es/cortazar/doctorado/articulos/leung-handbook.pdf>, (2000).
15. M. JORGENSEN, *A Review of Studies on Expert Estimation of Software Development Effort*, <http://www.simula.no/photo/expertssubmitnovember2002copy.pdf>
16. MENDES, E., MOSLEY, N. AND COUNSELL, S. *Web Metrics - Estimating Design and Authoring Effort*, <http://csdl.computer.org/comp/mags/mu/2001/01/u1050abs.htm>
17. PRESSMAN, ROGER S., *Software Engineering : A Practioner' s Approach*, 5th Ed., MC Graw - Hill, New York, (2001).

## DISKUSI

### UTAJA

1. Kalau program dibuat sendiri, bagaimana melakukan estimasi ?
2. Bagaimana menentukan harga jual ?

### SUHARJITO

1. Program atau software tersebut harus diestimasi sesuai dengan ukurannya (metriknya) yaitu dengan konsep FPA(*Function Point Analysis*) dengan mengukur banyaknya *field input*, *output*, entitas, *query* dan nilai kompleksitas dari *software* tersebut.
2. Untuk menentukan harga jual dapat dilakukan dengan beberapa cara salah satunya dengan menginputkan hasil pengukuran *metric function point* ke dalam model biaya yang diperoleh atau mengestimasi dengan biaya atau harga dari *software-software* yang sudah ada dan mempunyai kemiripan *functionalitas* dan diharapkan untuk melakukan estimasi harus mencoba atau memperhatikan beberapa metode estimasi yang ada sebagai validasi.

## DAFTAR RIWAYAT HIDUP

1. Nama : S U H A R J I T O, S.Si, MT
2. Tempat/Tanggal Lahir : Sleman, 26 Juli 1970
3. Instansi : Pusat T. Agroindustri, TAB, BPPT
4. Pekerjaan / Jabatan : Staff peneliti
5. Riwayat Pendidikan :
  - Universitas Gadjah Mada, lulus 1994, Jurusan Matematika (S1)
  - Institute Teknologi Sepuluh November, lulus Tahun 2000, Jurusan Teknik Informatika (S2)
6. Pengalaman Kerja :

- 1994 – 2000, peneliti di Direktorat Ilmu Kelautan, BPPT
  - 2000 – sekarang, peneliti di Pusat Teknologi Agroindustri, BPPT
  - 2001 – sekarang, Dosen Teknik Informatika, UBINUS
7. Makalah yang pernah disajikan :
- Suharjito dan Harianto, “Penggunaan B-Spline untuk Disain Kapal (Proceeding Seminar Nasional Teknologi Kelautan”, Suarabaya 13-14 Nopember 1995)
  - Suharjito “Image Compression Using Adaptive Segmentation” (Proceeding International Conference CECI’2001 7-8 March 2001)
  - Suharjito “Kompresi Lossless pada Citra Digital dengan metode segmentasi adaptif” (Media ISTA Volume II No. 2, Agustus 2001)
  - Suharjito “Perancangan sistem pakar Bidang Pertanian” (Buku paket teknologi agroindustri, juni 2002)
  - Suharjito, “Sistem Informasi Teknologi Agroindustri untuk menunjang Pengembangan agroindustri” (Proceding Teknologi Untuk Negeri, Mei 2003)
  - Suharjito dan M. Yusuf Jafar, “Pemanfaatan kulit split dan trimming untuk pembuatan gelatin dan prospek industri gelatin di Indonesia” Jurnal Sains dan Teknologi 2004
  - Suharjito dan Agus widodo, “perencanaan model estimasi biaya proyek perangkat lunak”, Jurnal Perencanaan Iptek BPPT Des 2005