

# PERANGKAT LUNAK UNTUK MENGESTIMASI BIAYA PENGEMBANGAN PERANGKAT LUNAK DENGAN MENGUNAKAN REFERENSI *COCOMO II 2000*

**Khakim Ghozali, Febriliyan Samopa, Nina Yuliasuti**

Program Studi Sistem Informasi,

Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember

Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Telp. + 62 31 5939214, Fax. + 62 31 5913804

Email : khakim@its-sby.edu

## ABSTRAK

Sifat dasar dari perangkat lunak selalu berkembang sehingga sulit untuk memprediksikan secara akurat biaya pengembangan dari sebuah produk perangkat lunak yang sedang dikembangkan tersebut. Oleh karena itu diperlukan proses untuk mengembangkan model yang dapat secara konstruktif menjelaskan siklus perkembangan perangkat lunak dan menyediakan sebuah dasar rekayasa untuk pertimbangan mengenai prediksi biaya untuk pengembangan sebuah produk perangkat lunak.

Pada makalah ini dibuat aplikasi yang memprediksikan biaya pengembangan perangkat lunak dengan menggunakan referensi model *COCOMO II*. Terdapat model yang memprediksi biaya pengembangan pada tahap awal proyek ketika sangat sedikit pengetahuan tentang ukuran dari produk yang sedang dikembangkan, karakter dari platform, karakter dari personal yang terlibat pada proyek, spesifikasi detail dari proses yang digunakan (*Early Design*) dan model yang lebih detail, yang digunakan ketika arsitektur life cycle perangkat lunak telah dikembangkan (*Post Architecture*). Pada model ini terdapat kemampuan yang merupakan rumpun penyesuaian dari model pengukuran perangkat lunak, melibatkan poin fungsi (*Function Point*) dan baris kode (*Source Lines of Code*). Terdapat juga model nonlinear untuk penggunaan kembali perangkat lunak, rekayasa ulang dan perubahan kebutuhan. Model pengembangan yang digunakan untuk perangkat lunak adalah *Waterfall* dan *MBase*. Prediksi biaya dan jadwal yang dihasilkan selain estimasi *most likely* juga pesimistik dan optimistik, sehingga kisaran yang diberikan lebih besar.

**Kata Kunci** : *COCOMO II 2000*, rekayasa, perangkat lunak, model.

## 1. PENDAHULUAN

Dengan adanya sifat dasar dari perangkat lunak yang selalu berkembang menyebabkan sulitnya memprediksikan secara akurat biaya pengembangan dari sebuah produk perangkat lunak yang sedang dikembangkan tersebut. Oleh karena itu diperlukan solusi yang optimal untuk mengatasi permasalahan tersebut. Sehingga diperlukan proses untuk mengembangkan model yang dapat secara konstruktif menjelaskan siklus perkembangan perangkat lunak dan menyediakan sebuah dasar rekayasa untuk pertimbangan mengenai prediksi biaya dan jadwal untuk pengembangan sebuah produk perangkat lunak, yang dalam hal ini digunakan konsep model *COCOMO II 2000* berdasarkan perhitungan poin fungsi, baris kode atau penggunaan ulang.

Aplikasi yang dibuat ini melakukan estimasi biaya pengembangan perangkat lunak baik *Post Architecture* maupun *Early Design*, yang terbagi atas biaya tiap modul dan biaya setiap instruksi dalam satu modul, perangkat lunak yang dibuat ini juga

melakukan perhitungan usaha nominal (*nominal effort*), usaha yang dibutuhkan (*estimation effort*), produktivitas, jumlah staf yang diperlukan. Perhitungan tersebut dilakukan untuk setiap modul dan juga total keseluruhan. *Effort* adalah jumlah kesatuan tenaga kerja yang dibutuhkan untuk menyelesaikan aktivitas atau elemen proyek lainnya, biasanya digambarkan sebagai jam kerja (*staffhours*), hari kerja (*staffdays*) atau minggu kerja (*staffweeks*). Dalam proses estimasi, aplikasi yang dibuat ini diharapkan dapat memberikan fasilitas kemudahan bagi pengguna.

Melihat tujuan dari aplikasi ini maka permasalahan yang dihadapi :

1. Bagaimana merancang aplikasi yang mampu menangani penghitungan dan pengolahan data dari parameter-parameter yang terlibat dalam pembuatan perangkat lunak dengan secara akurat.
2. Bagaimana aplikasi dapat digunakan untuk estimasi biaya pengembangan berbagai jenis

perangkat lunak yang dibuat berdasarkan perhitungan parameter-parameter penilaian yang jelas.

3. Bagaimana merancang basisdata dimana parameter dari data yang diperlukan bisa ditambah atau dimodifikasi apabila ada proses pembaharuan, dimungkinkan dengan adanya penelitian lebih lanjut.

Dari permasalahan-permasalahan di atas, maka aplikasi ini mempunyai batasan :

1. Informasi-informasi yang dihasilkan dari aplikasi berupa:
  - Informasi yang ditampilkan berupa rincian *effort*, biaya, jadwal dari data parameter perangkat lunak yang dimasukkan.
  - Informasi yang ditampilkan berdasarkan metode pembuatan yang dipilih untuk ditampilkan.
2. Pemodelan sistem ini dilakukan dengan menggunakan PAM (*Process Analyst Model*).
3. Estimasi biaya pengembangan perangkat lunak yang dibuat ini sesuai dengan referensi model yang dipakai, yaitu *COCOMO II 2000*.
4. Penentuan ukuran perangkat lunak yang dapat digunakan dalam penghitungan adalah dengan metode baris kode, poin fungsi dan penggunaan kembali.
5. Parameter yang digunakan adalah yang terdapat pada model *COCOMO II 2000*.

## 2. KONSEP COCOMO II 2000

Pada *COCOMO II 2000* secara umum terdapat dua metode pembuatan yaitu Early Design dan Post Architecture. Untuk masing-masing metode pembuatan mempunyai rumusan metode tersendiri.

### 1. Early Design

$$PM = A \times \text{Size}^E \times \prod_{i=1}^7 EM_i + PM_{\text{AUTO}}$$

### 2. Post Architecture

$$PM = A \times \text{Size}^E \times \prod_{i=1}^{17} EM_i + PM_{\text{AUTO}} \text{ dimana}$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

Terdapat 3 macam penghitungan ukuran modul, yaitu:

#### 1. Ukuran SLOC

$$\text{size} = \left( 1 + \frac{\text{REVL}}{100} \right) \times \text{size} \triangleright$$

#### 2. Ukuran Poin Fungsi

$$\text{Size} = \left( 1 + \frac{\text{REVL}}{100} \right) \times \left( \sum_{i=1}^5 \left[ \sum_{j=1}^3 (\text{input} \times \text{parameter}) \right] \right) \times \text{multiplier}$$

### 3. Ukuran Adaptasi dan Ruse

$$\text{AAM} = \left\{ \begin{array}{l} \frac{\text{AA} + \text{AAF} \times (1 + [0.02 \times \text{SU} \times \text{UNFM}])}{100}, \text{ Untuk AAF} \leq 50 \\ \frac{\text{AA} + \text{AAF} + (\text{SU} \times \text{UNFM})}{100}, \text{ Untuk AAF} > 50 \end{array} \right\}$$

dimana  $\text{AAF} = (0.4 \times \text{DM}) + (0.3 \times \text{CM}) \times (0.3 \times \text{IM})$

$$\text{Equivalent KSLOC} = \text{AdaptedKSLOC} \times \left( 1 - \frac{\text{AT}}{100} \right) \times \text{AAM}$$

$$\text{Size} = \left( 1 + \frac{\text{REVL}}{100} \right) \times (\text{New KSLOC} + \text{Equivalent KSLOC})$$

### Maintenance

Terdapat pertimbangan khusus pada maintenance software, yaitu :

1. Faktor biaya *SCED* tidak digunakan pada estimasi *effort* untuk *maintenance*.
2. Faktor biaya *RUSE* tidak digunakan pada estimasi *effort* untuk *maintenance*.
3. Faktor biaya *RELY* mempunyai seperangkat pengali *effort* yang berbeda untuk *maintenance*.

$$(\text{SIZE})_M = [(\text{Base Code Size}) \times \text{MFC}] \times \text{MAC}$$

$$\text{MFC} = \frac{\text{Size Added} + \text{Size Modified}}{\text{Base Code Size}}$$

dapat disederhanakan menjadi :

$$(\text{SIZE})_M = (\text{Size Added} + \text{Size Modified}) \times \text{MAF}$$

$$\text{dimana } \text{MAF} = 1 + \left( \frac{\text{SU}}{100} \times \text{UNFM} \right)$$

$$PM_M = A \times (\text{Size}_M)^E \times \prod_{i=1}^{15} EM_i$$

dimana faktor pengali *RELY* berubah, sedangkan untuk *SCED* dan *RUSE* tidak diikutkan.

$$\text{KSLOC}_{\text{THNKEN}} = \text{KSLOC}_{\text{THNKEN}-1} + (\% \text{size add} \times \text{KSLOC}_{\text{THNKEN}-1})$$

atau dapat disederhanakan menjadi

$$\text{KSLOC}_{\text{THNKEN}} = \text{KSLOC}_{\text{THNKEN}-1} \times (1 + \% \text{size add})$$

$$\text{SIZE}_M = [(\% \text{SIZE ADDED} + \% \text{SIZE MODIFIED}) \times \text{KSLOC}_{\text{THNKEN}}] \times \text{MAF}$$

$$PM_{\text{maintenance}_M} = A \times \text{SIZE}_M^E \times \prod_{i=1}^{15} EM_i$$

$$\text{FSPM} = \frac{PM_M}{TM} \text{ dimana FSPM adalah staff yang}$$

dibutuhkan untuk *maintenance* dan TM adalah waktu yang dibutuhkan untuk *maintenance*. TM bisa ditentukan sesuai dengan keinginan

Hal-hal yang akan menjadi keluaran dari aplikasi adalah :

1. Biaya

$$BIAYA = PM_{EST} \times BIAYA \text{ TENAGA KERJA}$$

2. Waktu

$$TDEV = \left[ C \times (PM_{NS})^F \right] \times \frac{SCED\%}{100}$$

3. Biaya per instruksi

$$BIAYA_I = \frac{BIAYA}{SLOC}$$

4. Produktivitas

$$PROD = \frac{SLOC}{PM_{EST}}$$

5. Staf

$$STAFF = \frac{PM_{EST}}{TDEV}$$

### 3. PERANCANGAN APLIKASI

Pada sistem aplikasi akan dijelaskan mengenai perancangan sistem penyimpanan yang menjadi bagian yang paling banyak ditangani dan juga penghitungan yang menjadi bagian inti dari Makalah ini.

1. Penyimpanan sementara mempunyai tujuan agar nilai parameter, khususnya, dapat digunakan kembali dan dilakukan perhitungan kapan pun setelah form di-unload. Seperti yang dikemukakan diatas bahwa penyimpanan sementara nilai-nilai dari parameter yang ada akan dilakukan dengan menggunakan variabel *array*.
2. Nilai-nilai yang dimasukkan ke dalam penyimpan tetap dalam basisdata sebagian besar merupakan nilai yang sebelumnya telah disimpan dalam *array* sebagai penyimpan sementara.

Perancangan dan pembuatan aplikasi penghitung biaya pembuatan perangkat lunak ini diimplementasikan menggunakan Prosesor Intel Pentium 4 1,8 GHz, RAM (Memori) 256 MHz., VGA card SiS 650\_651\_M650\_M652\_740, Windows 2000 sebagai sistem operasi, Microsoft Visual Basic 6.0 sebagai bahasa pemrograman dari aplikasi, Microsoft Access 2000 sebagai *database engine*.

Implementasi aplikasi ini meliputi implementasi antarmuka dan implementasi proses. Adapun beberapa implementasi proses yang adalah yaitu :

1. Proses penyimpanan sementara

Proses pemasukan nilai baru terjadi pada parameter model, metode pembuatan, pengali jadwal, faktor skala, data modul, data *maintenance*. Penyimpanan sementara ini disebabkan karena banyak jenis nilai harus dimasukkan pada *form* yang berbeda dimana nilai-nilai tersebut akan dilakukan perhitungan.

2. Proses perhitungan

Proses perhitungan disini meliputi perhitungan *effort* nominal, *effort* estimasi, produktivitas, biaya, biaya per instruksi, staf. Perhitungan ini meliputi baik tiap modul maupun total keseluruhan satu proyek.

### 4. UJI COBA DAN ANALISA HASIL

Pada sub bab ini akan dijelaskan mengenai perhitungan yang dilakukan pada uji coba. Setelah dilaksanakan uji coba kemudian dilakukan analisa dari hasil yang didapatkan. Uji coba ini menggunakan data Aplikasi Keuangan [4]. Aplikasi tersebut merupakan sistem informasi yang mengotomatisasi mekanisme pada suatu Pemerintah Kabupaten di Jawa Timur.

Perhitungan pada uji coba ini menggunakan metode pengembangan *Early Design*. Aplikasi Keuangan ini mempunyai karakteristik proyek sebagai berikut :

Tabel 1. Data Proyek

Faktor Skala	Level	Keterangan
<i>PREC(Precedentness)</i>	<i>High</i> (2.48)	Tim yang terlibat memahami tujuan dari proyek dan berpengalaman dengan sistem semacam itu.
<i>FLEX(Development Flexibility)</i>	<i>High</i> (2.03)	Secara umum ada kesesuaian terhadap kebutuhan dan penyelesaian produk sesuai waktu dan biaya.
<i>RESL(Architecture/Risk Resolution)</i>	<i>Low</i> (5.65)	Tahap identifikasi atau pengurangan resiko dilakukan pada beberapa bagian.
<i>TEAM(Team Cohesion)</i>	<i>Very High</i> (1.1)	Terdapat kerjasama yang tinggi antara tim <i>customer</i>

		dan <i>developer</i> .
<i>PMAT(Proccess Maturity)</i>	<i>Low</i> (6.24)	Penjadwalan, keuangan, fungsionalitas dan kualitas secara umum sudah mulai dimanajemen tetapi belum konsisten.

Aplikasi ini mempunyai empat modul yaitu Anggaran, Pembukuan, Perbendaharaan, Verifikasi. Dari rancangan menu, diagram ER dan aktivitas-aktivitas dari setiap modul, maka dapat dilakukan estimasi ukuran modul sebagai berikut :

- Estimasi ukuran modul dari aplikasi ini menggunakan perhitungan poin fungsi
- Bahasa pemrograman untuk semua modul menggunakan ASP sehingga pengali bernilai 15. Data pengolahan data (detailnya dijelaskan di buku), didapatkan ringkasan ukuran tiap modul, yaitu pada Tabel 2 dan Tabel 3.

**Tabel 2. Data 1 Ukuran Modul**

Modul	<i>ILF</i>	<i>EIF</i>	<i>EI</i>	<i>EO</i>	<i>EI Q</i>
Anggaran	6	0	22	6	0
Pembukuan	9	0	25	8	0
Perbendaharaan	4	0	0	2	0
Verifikasi	4	0	2	5	0

**Tabel 3. Data 2 Ukuran Modul**

Modul	Pengali	<i>REVL</i> (%)
Anggaran	15	0
Pembukuan	15	0
Perbendaharaan	15	0
Verifikasi	15	0

**Tabel 4. Data 1 Modul**

Modul	Ukuran ( <i>SLOC</i> )	Biaya Tenaga Kerja (US \$)

Anggaran	2670	400
Pembukuan	3450	400
Perbendaharaan	750	400
Verifikasi	1095	400

**Tabel 5. Data 2 Modul**

Modul	<i>EAF</i> <i>COCOMO</i> II 2000	<i>EAF</i> <i>COCOMO</i> II 1997	<i>EAF</i> <i>COCOMO</i> 1981
Anggaran	0.85	0.71	0.77
Pembukuan	0.85	0.71	0.8
Perbendaharaan	0.74	0.61	0.72
Verifikasi	0.74	0.61	0.63

Hasil estimasi biaya dengan menggunakan model *COCOMO* 1981 dapat dilihat pada Tabel 6.

**Tabel 6. Hasil Estimasi yang Didapatkan dari Model *COCOMO* 1981**

Modul	<i>Effort</i> (PM)	Biaya (US \$)
Anggaran	6.73	2692
Pembukuan	8.81	3524
Perbendaharaan	1.774	709.6
Verifikasi	2.64	1056
TOTAL		7981.6

Hasil estimasi dari aplikasi dengan menggunakan model *COCOMO* II 2000 bisa dilihat pada tabel 7. Hasil estimasi yang didapatkan dari aplikasi dengan menggunakan model *COCOMO* II 1997 bisa dilihat pada tabel 8. Menggunakan cara perhitungan yang sama dengan Aplikasi yang menggunakan model *COCOMO* II 2000, didapatkan hasil estimasi dari *Software COCOMO* 2000 dan bisa dilihat pada Tabel 9.

**Tabel 7. Hasil Estimasi Aplikasi dengan Model *COCOMO* II 2000**

Modul	<i>Effort</i> Nom (PM)	<i>Effort</i> Est (PM)	Prod ( <i>SLOC</i> /PM)	Biaya (US \$)	Biaya tiap instr (US \$)	Staf ( <i>person</i> )
Anggaran	8.09	6.9	386.85	2760.75	1.03	1.09
Pembukuan	10.45	8.92	386.85	3567.26	1.03	1.41
Perbendaharaan	2.27	1.69	444.66	674.68	0.9	0.27
Verifikasi	3.32	2.46	444.66	985.03	0.9	0.39
TOTAL		19.97	398.86	7987.71	1	3.16

**Tabel 8. Hasil Estimasi Aplikasi dengan COCOMO II 1997**

Modul	Effort Nom (PM)	Effort Est (PM)	Prod (SLOC/PM)	Biaya (US \$)	Biaya tiap instr (US \$)	Staf (person)
Anggaran	5.78	4.11	649.05	1645.49	0.62	0.99
Pembukuan	7.47	5.32	649.05	2126.2	0.62	1.28
Perbendaharaan	1.62	1.01	746.03	402.13	0.54	0.24
Verifikasi	2.37	1.47	746.03	587.11	0.54	0.35
TOTAL		11.9	669.2	4760.92	0.6	2.86

**Tabel 9. Hasil Estimasi Software COCOMO II 2000**

Modul	Effort Nom (PM)	Effort Est (PM)	Prod (SLOC/PM)	Biaya (US \$)	Biaya tiap instr (US \$)	Staf (person)
Anggaran	8.1	6.9	386.9	2760.75	1.0	1.1
Pembukuan	10.4	8.9	386.9	3567.26	1.0	1.4
Perbendaharaan	2.3	1.7	444.7	674.68	0.9	0.3
Verifikasi	3.3	2.0	535.7	817.57	0.7	0.3
TOTAL		19.6	407.4	7820.25	1	3.1

#### Uji Kebenaran

Pada uji kebenaran ini, dilakukan perbandingan antara Aplikasi yang melakukan estimasi dengan menggunakan model COCOMO II 2000 dengan Software COCOMO II 2000.

**Tabel 10. Perbandingan Hasil**

Aplikasi dengan model COCOMO II 2000 (US \$)	Software COCOMO II 2000 (US \$)	SELISIH (US \$)	%
7987.71	7987.71	0	0

#### Uji Data

Pada uji data ini, dilakukan perbandingan antara Aplikasi yang melakukan estimasi dengan menggunakan model COCOMO II 2000 dengan model COCOMO II 1997 dan COCOMO 1981. Untuk mengkonversikan biaya pengembangan yang diestimasi, maka US \$ 1 = Rp 10000,00.

**Tabel 11. Perbandingan Hasil**

	MODE L COCOMO (US \$)	KENYATAAN (Rp)	SELISIH (Rp)	%
Aplikasi dengan model COCOMO II 2000	7987.71	82800000	2922900	3.53

(\$)				
Aplikasi dengan model COCOMO II 1997 (\$)	4760.92	82800000	35190800	42.5
Model COCOMO 1981 (\$)	7981.6	82800000	2984000	3.6

#### Analisa Hasil

Sesuai dengan tujuan dibuatnya aplikasi yaitu untuk mendapatkan informasi mengenai proyek, seperti biaya, effort, waktu, produktivitas, staf yang mendekati kenyataan.

Analisa mengenai uji coba diatas adalah:

1. Dengan SLOC  
Perhitungan dengan pengukuran SLOC lebih sulit untuk dilakukan karena harus menghitung secara langsung, dimana ini membutuhkan waktu dan tenaga.
2. Dengan Poin Fungsi  
Pengukuran dengan cara ini agak lebih mudah dilakukan, karena tidak menghitung LOC (Line of Code) secara langsung, melainkan dengan pembobotan dan nilai konversi.
3. Dengan Adaptasi dan Penggunaan Kembali  
Pengukuran ini sangat bermanfaat terutama apabila menggunakan komponen yang reuse, yaitu biaya pengembangan lebih bisa dikurangi, karena effort yang dibutuhkan lebih sedikit.

Untuk komponen yang *reuse*, semakin komponen tersebut bisa *reuse* di berbagai macam produk atau aplikasi maka parameter pengali *RUSE* secara umum lebih tinggi. Ini berpengaruh terhadap effort dan biaya yang lebih besar pula .

## 5. KESIMPULAN

Setelah melakukan uji coba dan evaluasi pada makalah ini maka dapat diambil kesimpulan sebagai berikut:

1. Hasil estimasi biaya perangkat lunak dengan menggunakan model *COCOMO II 2000* dibandingkan dengan *COCOMO II 1997* maupun *COCOMO 1981* lebih akurat yaitu dengan tingkat kesalahan 3.53%.
2. Pengukuran dengan poin fungsi lebih mudah daripada *SLOC* karena tidak menghitung secara langsung tetapi menggunakan nilai konversi dan pembobotan.
3. Pembuatan struktur basisdata yang memasukkan faktor pengali maupun faktor skala sebagai *record* dan bukan *field*, menjadikan basisdata lebih fleksibel apabila ada penambahan faktor pengali maupun faktor skala tersebut

## 6. KEMUNGKINAN PENGEMBANGAN

Kemungkinan pengembangan untuk aplikasi penghitung biaya pembuatan perangkat lunak ini antara lain:

1. Pengumpulan data bisa menambah atau mengurangi nilai parameter yang mempengaruhi penghitungan dimana data tersebut disesuaikan dengan kondisi pengguna aplikasi, agar hasil yang diharapkan lebih akurat.
2. Untuk memudahkan pengukuran dengan penghitungan *SLOC* secara langsung maka diperlukan aplikasi penghitung *SLOC* otomatis untuk berbagai bahasa pemrograman, yang dihubungkan langsung dengan aplikasi pengestimasi biaya, sehingga bisa secara langsung didapatkan ukuran dari suatu komponen.

3. Menentukan karakteristik dari nilai parameter agar pengguna lebih mudah untuk memasukkan data.

## 7. DAFTAR PUSTAKA

1. W. Barry Boehm, "Software Engineering Economics". New Jersey-Prentice Hall PTR, 1981.
2. Barry, W. Boehm, Chris Abts, Winsor A. Brown, Sunita Chulani, K. Clark Bradford; Ellis Horowitz, Ray Madachy, J. Donald Madachy, Bert Steece, "Software Cost Estimation with COCOMO II", New Jersey-Prentice Hall PTR, 2000.
3. Sunita Chulani, "Bayesian Analysis of Software Cost and Quality Models".
4. Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, "Rencana Pengembangan Software Aplikasi Keuangan", Proyek Penerapan Sistem Komputerisasi Kabupaten Malang Tahun Anggaran 2002 : 2002.
5. Ir. Fathansyah, "Basisdata". Informatika Bandung, 1999.
6. Vajde Horvat Romana, Ivan Rozman, Vesna Lešnik, "SoPCoM – Model for Evaluation of the Software Processes Complexity".
7. Björn Kjellsson, Berglie Kjellsson, "A Resource Estimation Method Recommendation".
8. Ario Kusumo, Drs. Suryo, "Buku Latihan Microsoft Visual Basic 6.0". Jakarta-PT Elex Media Komputindo, 2001.
9. Raj Mukherji, Ph.D., "Business Case for A Common Services Based Architecture".
10. Roger S. Pressman, Ph.D., "Software Engineering (A Practitioner's Approach) 4<sup>th</sup> Edition", Singapore-The McGraw-Hill Companies, Inc., 1997.
11. Aris Tjahyanto, "Diktat Kuliah Basisdata".
12. Wahana Komputer Semarang, Andi Yogyakarta, "Pemrograman Visual Basic 6.0". Yogyakarta-Andi Offset, 2000.