

**Makalah Seminar Tugas Akhir**  
**ESTIMASI BIAYA PEMBUATAN PERANGKAT LUNAK**  
**MENGGUNAKAN METODE COCOMO II PADA**  
**SISTEM INFORMASI PELAPORAN KEGIATAN PEMBANGUNAN**

**Eko Handoyo**  
**R. Rizal Isnanto**  
**Aderian Primaraka**

**Abstract :** Nowadays, software is absolutely important for individual or company in many matters. Software design or development is done based on future or certain condition. So, it is important to understand these conditions to calculate cost and duration of a software project.

This research discusses on cost estimation method of software project, this is COCOMO II (Constructive Cost Model). COCOMO II has three submodels i.e. Application Composition, Early Design, and Post Architecture, which has a possibility to estimate in less or complete condition of information. By using this COCOMO II estimation model, total efforts being needed to complete a software project in person month and total duration of processing or developing in month can be known. By adapting project standard value in the area with certain time, nominal value of a software project can be determined.

Study case completed in Development Activity Report Information System, gives a conclusion that COCOMO II method is suitable to calculate cost (effort) and schedule (time) estimation. Size used as basic calculation is SLOC (Source Line Of Code). In this research, SLOC is determined by calculating UFP (Unadjusted Function Points), while SLOC is determined by calculating the number of source code line of the former project.

**Keywords :** Software Cost Estimation, COCOMO II, Post Architecture

Keuntungan kompetitif semakin bergantung pada pengembangan produk dan pelayanan yang cerdas dan saling terkait, serta pada kemampuan dalam mengembangkan dan mengadaptasi produk dan pelayanan tersebut dengan lebih cepat dibanding waktu adaptasi pesaing.

Perencanaan, analisis dan kontrol yang efektif dari suatu proyek perangkat lunak diatasi dengan diformulasikannya versi terbaru dari *Constructive Cost Model* (COCOMO) untuk estimasi usaha, biaya dan jadwal pembuatan perangkat lunak atau yang lebih sering disebut COOMO II.

#### **Batasan Masalah**

1. Pada penelitian ini hanya akan dibahas mengenai aplikasi model estimasi pembiayaan pada perangkat lunak Sistem Informasi Pelaporan Kegiatan Pembangunan yang telah selesai dibuat.
2. Model yang akan diaplikasikan adalah COCOMO II dan hanya pada submodel *Post Architecture*.
3. Perangkat lunak yang digunakan untuk simulasi, tidak dibahas pembuatannya, karena hanya sebagai pembandingan.

#### **Tujuan Penelitian**

Tujuan dari tugas akhir ini adalah untuk mengaplikasikan model estimasi pembiayaan perangkat lunak COCOMO II untuk submodel *Post Architecture* pada Sistem Informasi Pelaporan Kegiatan Pembangunan dan tercapainya profesionalisme dalam menyusun dan melaksanakan suatu proyek perangkat lunak.

#### **Pendahuluan COCOMO II**

Pada tahun 1981, Barry Boehm mendesain COCOMO untuk memberikan estimasi/perkiraan jumlah *person-months* untuk mengembangkan suatu produk perangkat lunak. Referensi pada model ini dikenal dengan nama COCOMO 81. Model estimasi COCOMO telah digunakan oleh ribuan manajer proyek suatu proyek perangkat lunak, dan berdasar pada pengalaman dari ratusan proyek sebelumnya.

Secara umum, referensi COCOMO sebelum 1995 merujuk pada model original COCOMO yaitu COCOMO 81, kemudian setelah itu merujuk pada COCOMO II. COCOMO II adalah suatu usaha untuk memperbaiki model estimasi biaya perangkat lunak COCOMO yang dipublikasikan dalam *Software Engineering Economics* oleh Dr. Barry Boehm pada tahun 1981.

---

*Eko Handoyo, R. Rizal Isnanto, adalah dosen di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro (Undip) Semarang Jl. Prof. Sudharto, S.H. Tembalang Semarang 50275*

*Aderian Primaraka adalah mahasiswa di Jurusan Teknik Elektro Fakkultas Teknik Universitas Diponegoro (Undip) Semarang Jl. Prof. Sudharto, S.H. Tembalang Semarang 50275*

Usaha penelitian COCOMO dilakukan oleh Direktur Pusat Rekayasa Perangkat Lunak di *University Of Southern California*, Dr. Barry Boehm dan beberapa peneliti lainnya.

### Submodel COCOMO II

1. Komposisi Aplikasi  
Model Komposisi Aplikasi mendukung tahapan ini dan beberapa aktivitas pembuatan prototipe lain selanjutnya akan muncul dalam siklus hidup.
2. *Early Design*  
Model ini menggunakan *function points* untuk pengukuran, dan satu set dari lima penggerak biaya yang masih kasar.
3. *Post Architecture*  
Pada saat proyek siap untuk dikembangkan, proyek harus memiliki arsitektur siklus hidup yang memberikan informasi yang lebih akurat pada masukan-masukan penggerak biaya dan memungkinkan estimasi biaya untuk lebih akurat.

### Source Line Of Code (SLOC)

COCOMO II menggunakan pernyataan *logical source* sebagai standar SLOC. *Logical source* disebut *Logical Line Of Code* (LLOC). LLOC adalah jumlah dari baris logik. Baris logik dapat digolongkan sebagai *Line Of Code* bila mengandung tidak hanya komentar atau baris kosong. Jadi baris eksekusi dan baris deklarasi dihitung sebagai LLOC. Sedangkan baris komentar dan baris kosong tidak termasuk *Line Of Code*.

Perintah pengkompilasi (`#const`, `#if`) dihitung sebagai kode. Sedangkan kode yang dikeluarkan oleh perintah kondisional `False` dalam blok `#if .. #then .. #elseif .. #else .. #end if` tidak dihitung sebagai kode. Secara ringkas dapat dijelaskan bahwa LLOC menghitung semua baris logik, kecuali :

- a. Baris komentar (LLOC')
- b. Baris kosong (LLOW)
- c. Baris yang dikeluarkan oleh perintah pengkompilasi kondisional

### Function Points (FP)

FP mengukur proyek perangkat lunak dengan mengkuantisasi kegunaan pemrosesan informasi yang berhubungan dengan tipe berkas, keluaran, masukan data atau kontrol eksternal. Lima tipe fungsi pengguna perlu untuk diidentifikasi terlebih dahulu. Tabel 1. menjelaskan kelima tipe fungsi pengguna dalam estimasi berdasarkan FP.

Tabel 1. Tipe Fungsi Pengguna

<i>External Input</i> (EI)	Jumlah setiap tipe data atau masukan kontrol pengguna unik yang memasuki batas luar dari sistem perangkat lunak yang sedang diukur dan menambah atau mengubah data dalam berkas internal logik.
<i>External Outputs</i> (EO)	Jumlah setiap tipe data atau keluaran kontrol unik yang meninggalkan batas luar dari sistem perangkat lunak yang sedang diukur.

Tabel 1. Tipe Fungsi Pengguna (Lanjutan)

<i>Internal Logical File</i> (ILF)	Jumlah setiap grup logik utama dari data atau informasi kontrol pengguna dalam sistem perangkat lunak sebagai tipe berkas internal logik. Meliputi setiap berkas logik yang dibuat, digunakan atau dirawat oleh sistem perangkat lunak.
<i>External Interface Files</i> (EIF)	Berkas yang dilewatkan atau dibagikan diantara sistem perangkat lunak seharusnya dihitung sebagai tipe berkas antarmuka eksternal dalam setiap sistem.
<i>Extenrnal Inquiry</i> (EI)	Jumlah setiap kombinasi masukan-keluaran, dimana masukan menyebabkan dan menimbulkan keluaran yang seketika, sebagai tipe <i>inquiry</i> eksternal.

Sumber : Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., and Selby, R., *Cost Models for Future Life Cycle Processes: COCOMO II*, Science Publisher, Amsterdam, 1995

Setiap bagian dari tipe fungsi kemudian diklasifikasikan oleh tingkat kerumitan. Tingkat kerumitan menentukan bobot yang akan diaplikasikan pada jumlah fungsi untuk menentukan kuantitas *Unadjusted Function Points* (UFP).

UFP yang telah diperoleh harus diubah kedalam SLOC supaya dapat disubstitusikan ke dalam persamaan estimasi usaha. Tabel 2. menunjukkan aturan konversi UFP ke SLOC.

Tabel 2. Daftar Konversi UFP ke SLOC dalam berbagai bahasa

BAHASA	SLOC / UFP
C++ default	53
COBOL default	107
Delphi 5	17
HTML	14
Visual Basic 6	24
SQL default	13
Java 2 default	1

Sumber : Sarno, R., Biliati, J., L., Maimunah, S., *Pengembangan Metode Analogy Untuk Estimasi Biaya Rancang Bangun Perangkat Lunak*, Makara Teknologi Vol. 6 No. 2, 2002.

### Estimasi Biaya

Penghitungan usaha berbeda untuk setiap submodel COCOMO II. Untuk submodel *early design* dan *post architecture* menggunakan Persamaan 1.

$$PM_{NOMINAL} = A \times (Size)^E \dots \text{(Pers. 1)}$$

Persamaan di atas menggunakan input *Size* dari pengembangan perangkat lunak, A konstan dan E adalah eksponen faktor skala (SF). Ukuran yang digunakan adalah dalam *kilo source line of code* (KSLOC). KSLOC diperoleh dari perkiraan ukuran modul perangkat lunak yang menyusun program aplikasi. Ukuran tersebut dapat juga diperkirakan dari *unadjusted function points* (UFP), kemudian diubah kedalam SLOC dibagi 1000.

Penggerak skala (B) digunakan untuk menghitung skala ekonomis dan nonekonomis yang ditemukan untuk proyek perangkat lunak dari ukuran yang berbeda-beda. Konstan A digunakan untuk menangani pengaruh multiplikatif pada usaha dengan proyek peningkatan ukuran.

### Estimasi Jadwal

COCOMO II memberikan sebuah kemampuan estimasi jadwal yang sederhana, sama halnya pada COCOMO dan Ada COCOMO. Persamaan 2. berikut

digunakan untuk menghitung estimasi jadwal untuk semua submodel COCOMO II.

$$TDEV = \left[ 3,67 \times (PM)^{0,28 + 0,2 \times (B - 1,01)} \right] \times \frac{SCED\%}{100} \dots \text{(Pers. 2)}$$

TDEV (*Time Development*) adalah waktu dalam bulan dari penentuan dari garisbesar kebutuhan produk untuk penyelesaian dari suatu penerimaan aktivitas yang menjamin kebutuahn kepuasan akan produk, PM adalah *person month* yang telah diestimasi tanpa melibatkan EM SCED, dan SCED% adalah persentase kompresi atau perluasan dalam EM SCED.

**Scale Drivers (B)**

Dalam COCOMO II *scale drivers* memiliki tingkat rating yang dapat dipilih berdasar pada aturan bahwa terdapat *source* yang signifikan dari variasi eksponensial pada usaha proyek atau variasi produktivitas. Setiap skala rating mempunyai rentang dari *Very Low* hingga *Extra High*. Setiap tingkat rating memiliki bobot yang disebut *W*, nilai spesifik dari bobot disebut faktor skala. Skala faktor proyek ditambahkan untuk seluruh faktor, dan digunakan untuk menentukan eksponen skala B, sesuai dengan Persamaan 3.

$$E = B + 0,01 \times \sum W_i \dots \dots \dots \text{(Pers. 3)}$$

Faktor skala ada lima, yaitu *Precedentedness* (PREC), *Development Flexibility* (FLEX), *Risk Resolution* (RESL), *Team Cohesion* (TEAM), *Project Maturity* (PMAT)..

**Effort Multipliers (EM)**

Berikut ini akan dijabarkan 17 EM yang digunakan dalam submodel *Post Architecture* COCOMO II untuk menentukan usaha nominal yang menggambarkan proyek perangkat lunak yang sedang dikembangkan. Masing-masing EM dibagi ke dalam 6 kelas, mulai dari *very low* hingga *extra high*. 17 EM tersebut dikelompokkan ke dalam 4 kelompok yaitu : *product*, *platform*, *personnel*, dan *project*.

Tabel 3 Faktor Biaya Dan Pengelompokkannya

Kelompok	Effort Multipliers
Product	RELY, DATA, CPLX, RUSE, DOCU
Platform	TIME, STOR, PVOL
Personnel	ACAP, PCAP, PCON, AEXP, PEXP, LTEX
Project	TOOL, SITE, SCED

Sumber : Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., and Selby, R., *Cost Models for Future Life Cycle Processes: COCOMO II*, Science Publisher, Amsterdam, 1995

Setelah dilakukan wawancara dengan tim pengembang, maka akan terkumpul informasi berupa data-data tentang faktor skala, penggerak biaya, dan faktor fungsi pengguna *function points*.

**Faktor Skala**

Data mengenai faktor skala dikumpulkan berdasarkan informasi yang terkumpul menggunakan faktor skala. Tabel 4., menunjukkan salah satu contoh tabel faktor skala yaitu PREC.

Tabel 4. Pengumpulan Data PREC

Fitur	Very Low	Nominal/High	Extra High
Pemahaman tim tentang tujuan produk.	Umum	Lengkap	Terperinci
Pengalaman bekerja dengan perangkat lunak yang berhubungan.	Sedang	Lengkap	Luas
Pengembangan bersama dari perangkat keras dan prosedur operasional yang berhubungan.	Luas	Sedang	Beberapa
Kebutuhan akan pemrosesan data, arsitektur dan algoritma yang inovatif.	Lengkap	Beberapa	Minimal

Khusus untuk faktor skala PMAT, sebelumnya perlu dilakukan wawancara untuk mengisi tabel *Key Proses Area* (KPA) yang mempunyai bobot mulai dari *Almost Always* hingga *Does Not Apply* adalah 100%, 75%, 50%, 25%, 1%, dan 0%.

Tabel 5. Rata-rata bobot KPA

KPA	Kejadian	Bobot
<i>Requirement Management</i>	<i>Rarely if ever</i>	0,01
<i>Software Project Planning</i>	<i>Rarely if ever</i>	0,01
<i>Software Project Tracking and Oversight</i>	<i>About Half</i>	0,50
<i>Software Subcontract Management</i>	<i>Does Not Apply</i>	0,00
<i>Software Quality Assurance</i>	<i>Does Not Apply</i>	0,00
<i>Software Configuration Management</i>	<i>Does Not Apply</i>	0,00
<i>Organization Process Focus</i>	<i>Occasionally</i>	0,25
<i>Organization Process Definition</i>	<i>Rarely if ever</i>	0,01
<i>Training Program</i>	<i>Does Not Apply</i>	0,00
<i>Integrated Software Management</i>	<i>Rarely if ever</i>	0,01
<i>Software Product Engineering</i>	<i>Rarely if ever</i>	0,01
<i>Intergroup Coordination</i>	<i>About Half</i>	0,50
<i>Peer Reviews</i>	<i>Frequently</i>	0,75
<i>Quantitative Process Management</i>	<i>Does Not Apply</i>	0,00
<i>Software Quality Management</i>	<i>Does Not Apply</i>	0,00
<i>Defect Prevention</i>	<i>Rarely if ever</i>	0,01
<i>Technology Change Management</i>	<i>Does Not Apply</i>	0,00
<i>Process Change Management</i>	<i>Does Not Apply</i>	0,00

Dengan menggunakan persamaan EPML. dan disesuaikan dengan tabel CMM. , maka akan didapatkan nilai PMAT sebesar **6,24**. Sedangkan nilai faktor skala yang lain adalah PREC = **4,65**, FLEX = **1,69**, RESL = **5,66**, TEAM = **3,56**.

Dengan menggunakan persamaan faktor skala, maka dapat dihitung faktor pangkat E sebesar **1,13**.

### Penggerak Biaya

Data mengenai faktor skala dikumpulkan berdasarkan informasi yang terkumpul menggunakan faktor skala. Tabel 6. menunjukkan seluruh hasil pengumpulan data faktor skala.

Tabel 6. Hasil Pengumpulan Data Faktor Skala

Penggerak Biaya	Level	Effort Multiplier
RELY	Nominal	1.00
DATA	High	1.14
RUSE	Low	0.95
DOCU	Very low	0.81
TIME	Nominal	1.00
STOR	Low	n/a
PVOL	Low	0.87
ACAP	Low	1.19
PCAP	Nominal	1.00
PCON	Very high	0.81
APEX	Nominal	1.00
PLEX	High	0.91
LTEX	Very low	1.20
TOOL	Nominal	1.00
SITE	High (Low dan Extra high)	$(1.09 \times 0.5) + (0.80 \times 0.5) = 0.95$
SCED	Very high	1.00
CPLX	Nominal	1.00

Dengan menggunakan persamaan faktor penggerak biaya, maka didapatkan nilai EM sebesar **0,76**.

### Faktor Fungsi Pengguna

Pengumpulan data mengenai faktor fungsi pengguna, selain dilakukan melalui analisis perangkat lunak SIPKP. Gambar 1. menunjukkan analisis dari salah satu proses dalam SIPKP untuk faktor fungsi pengguna *external inquiry* (EQ).



Gambar 1. EO pada proses Satker

Kemudian dari analisis proses dapat dikumpulkan data mengenai faktor fungsi pengguna EQ, yang dapat dilihat pada Tabel 7.

Tabel 7. Daftar seluruh proses EO beserta tipe elemen data, rujukan tipe berkas, dan bobot

Proses	Tipe Elemen Data	Rujukan Tipe Berkas	Bobot
Subbidang	1	1	Low
Program	1	2	Low
Kegiatan	2	4	Average
Indikator	11	2	Average
Sub Indikator	8	3	Average
Unitkerja	2	1	Low
Login Satker	1	1	Low
Konfirmasi Password	2	1	Low

### Penghitungan SLOC Manual

Perangkat lunak SIPKP menggunakan bahasa pemrograman Visual Basic 6.0, baris kodenya terdiri dari kode program dan kode tersembunyi. Kode tersembunyi adalah kode yang tidak tampak dalam penyunting kode, tetapi sebenarnya merupakan kode pembangun antarmuka, dan kode untuk menambahkan komponen yang diperlukan untuk komunikasi ke basis data atau yang lain seperti MSADODC. Oleh karena itu penulis menggunakan *tool* tambahan yang bersifat *shareware*, yang dapat menunjukkan secara rinci kode sumber dari setiap antarmuka ataupun pemanggil komponen yaitu Code Counter Pro v1.32. Tabel 8. menunjukkan hasil penghitungan SLOC manual.

Tabel 8. Perincian penghitungan SLOC

Berkas	Kode Program	Kode Tersembunyi	SLOC
Module1.bas	11	0	11
frmLogin.frm	76	172	248
frmSplash.frm	7	30	37
frm_akumulasi.frm	44	204	248
frm_bidang.frm	162	708	870
frm_ik06.frm	1483	4693	6176
frm_ik_06.frm	54	92	146
frm_ingat.frm	37	156	193
frm_kegiatan.frm	449	1518	1967
frm_lapik.frm	62	225	287
frm_pilihan.frm	58	149	207
frm_program.frm	273	601	874
frm_rencana fisik.frm	102	740	842
frm_rfk_1.frm	36	247	283
frm_rfk_2.frm	21	120	141
frm_rfk_3.frm	34	204	238
frm_subbidang.frm	202	522	724
frm_sub_benefit.frm	196	490	686
frm_sub_impact.frm	196	490	686
frm_sub_outcome.frm	196	490	686
frm_sub_output.frm	196	490	686
frm_unitkerja.frm	96	364	460
<b>TOTAL</b>	<b>3991</b>	<b>12705</b>	<b>16696</b>

### Penghitungan UFP Dan Konversi SLOC

Setelah data dari kelima faktor fungsi pengguna didapatkan, kemudian dapat dihitung UFP sesuai dengan langkah-langkah penghitungan UFP. Sehingga didapatkan hasil akhir total UFP, dan dapat dilihat pada Tabel 9.

Tabel 9. Total UFP dari seluruh fungsi pengguna

Fungsi Pengguna	Bobot	Nilai Bobot	Jumlah	Total
ILF	Low	7	11	77
	Average	10	1	10
	High	15	-	-
EIF	Low	5	-	-
	Average	7	-	-
	High	10	-	-
EI	Low	3	6	18
	Average	4	2	8
	High	6	5	30
EO	Low	4	5	20
	Average	5	3	15
	High	7	-	-
EQ	Low	3	13	39
	Average	4	1	4
	High	6	1	6
<b>Total UFP</b>			<b>227</b>	

Total UFP kemudian diubah kedalam bentuk SLOC, sesuai dengan tabel konversi UFP ke SLOC dan didapatkan hasil sebesar **5.448 SLOC** atau **5,448 KSLOC**.

**Penghitungan SLOC Menggunakan SLOC Metrics v3.0.6**

Penghitungan jumlah SLOC menggunakan bantuan program *freeware* yaitu SLOC Metrics v3.0.6. Gambar 2. menunjukkan tampilan hasil penghitungan SLOC menggunakan *freeware*.

File	SLOC	Comments	Total
File: E:\In\w\CAMPUS\research\CORE\Program_2.kj	100.00%	18.670	1846
File: _006.frm	35.42%	7.173	73
File: _016.frm	11.85%	2.412	5
File: _020.frm	3.10%	633	0
File: _021.frm	5.91%	120	42
File: _022.frm	4.92%	101	5
File: _023.frm	4.35%	893	0
File: _024.frm	4.20%	754	0
File: _025.frm	4.20%	754	0
File: _026.frm	4.18%	751	0
File: _027.frm	4.18%	751	0
File: _028.frm	2.62%	490	0
File: _029.frm	1.54%	297	11
File: _030.frm	1.48%	277	12
File: _031.frm	1.42%	266	0
File: _032.frm	1.25%	254	12
File: _033.frm	1.25%	253	12
File: _034.frm	1.15%	233	0
File: _035.frm	1.07%	200	0
File: _036.frm	0.92%	183	0
File: _037.frm	0.70%	133	12
File: _038.frm	0.20%	37	0
File: _039.frm	0.06%	12	0
Total: E:\In\w\CAMPUS\research\CORE\Program_2.kj	100.00%	18.670	1846

Gambar 2. Snapshot hasil penghitungan SLOC menggunakan SLOC Metrics v3.0.6

Dengan menggunakan SLOC Metrics v3.0.6, didapatkan total SLOC sebesar **18.670 SLOC** atau **18,67 KSLOC**.

**Estimasi Usaha Manual**

Usaha suatu proyek perangkat lunak pada tahap submodel *post architecture* dihitung menggunakan Persamaan 4 berikut.

$$PM = A \times Size^E \prod_{i=1}^{17} EM_i \dots\dots (Pers. 4.)$$

Diketahui bahwa nilai **A = 2,94** (menurut COCOMO II 2000), **Size = 16,696 KSLOC**, **E = 1,13** dan perkalian **EM<sub>1-17</sub> = 0,76**. Setelah semua variabel diketahui nilainya, maka didapatkan PM sebesar **53,79 PM**.

**Estimasi Jadwal Manual**

Jadwal disini dapat diartikan sebagai durasi proyek. Parameter yang mendasari penghitungan jadwal adalah penggerak biaya SCED. Untuk menghitung jadwal diperlukan total usaha PM tanpa melibatkan penggerak biaya SCED. Karena penggerak biaya SCED adalah 1.00 sehingga PM yang tidak melibatkan penggerak biaya SCED sama besarnya dengan total PM yang sebenarnya. Durasi proyek dapat dihitung melalui Persamaan 5. berikut.

$$TDEV = \left[ 3,67 \times (PM)^{(0,28 + 0,2 \times (E - 1,01))} \right] \times \frac{SCED\%}{100} \dots\dots (Pers. 5.)$$

Diketahui **PM = 53,79**; **E = 1,13**; dan **SCED% = 160**. setelah melalui penghitungan didapatkan durasi proyek sebesar **19,72** atau kalau dibulatkan selama **20 bulan**.

**Perbedaan SLOC Antara Metode UFP Dengan Metode Manual**

Perbandingan hasil estimasi SLOC dengan metode UFP dengan hasil total penghitungan kode program dengan metode manual akan ditampilkan pada Tabel 10.

Tabel 10 Hasil Estimasi SLOC Dengan Metode UFP Manual (Kode Program)

SLOC	
UFP	Manual (Kode Program)
5.448	3.991

Perbedaan hasil penghitungan estimasi SLOC antara metode UFP dengan manual dikarenakan UFP tidak memasukkan perkiraan antarmuka yang akan terjadi dalam aturan penghitungannya. Jadi perbandingan yang tepat bagi hasil estimasi SLOC dengan metode UFP adalah total penghitungan kode program manual, tanpa mempertimbangkan total penghitungan kode tersembunyi.

**Perbedaan SLOC Antara Metode Manual Dengan Hasil Penghitungan Freeware**

Tabel 11. menunjukkan perbedaan hasil penghitungan antara metode manual dengan *freeware*.

Tabel 11. Hasil Estimasi SLOC Dengan Metode Freeware Dan Dengan Metode Manual

SLOC	
Freeware	Manual
18.690	16.696

Perbedaan tersebut disebabkan oleh persepsi SLOC yang digunakan oleh *freeware* berbeda dengan yang digunakan penulis dalam melakukan penghitungan manual. Persepsi logik yang digunakan oleh *freeware* adalah semua baris kode yang terjadi tanpa baris komentar dan baris kosong. Sedangkan persepsi logik yang dipakai oleh penulis adalah semua baris kode tanpa baris komentar, baris kosong, dan hanya salah satu cabang dalam algoritma percabangan. Jadi perbedaannya terletak pada dasar penghitungan SLOC saat menjumpai kode sumber dengan algoritma percabangan.

**Pengujian Menggunakan Online Freeware Dan Software Modifikasi Dengan Masukan Size, Faktor Skala, Dan Penggerak Biaya Dari Hasil Penghitungan Manual**

*Freeware* yang digunakan sebagai penguji estimasi usaha dan jadwal diambil dari situs <http://sunset.usc.edu/research/COCOMOII/> yang dibuat oleh [madachy@usc.edu](mailto:madachy@usc.edu).

Sebagai perbandingan estimasi *freeware*, saya memodifikasi *freeware* tersebut sesuai dengan dasar teori yang saya dapatkan. Secara garis besar sama, hanya terdapat perbedaan dalam pemasukan data *scale drivers* dan *cost drivers multisite development*.

Pengujian berikut ini dilakukan dengan menghitung estimasi usaha dan jadwal, menggunakan 2 perangkat lunak, yaitu *online freeware* dan *software Modifikasi*. Data faktor skala dan penggerak biaya diambil dari hasil pengumpulan data pada bab 3. Setelah kesemua data dimasukkan, maka dengan menekan tombol proses, semua data akan dihitung, dan didapatkan hasil estimasi *effort* dan *schedule* seperti dapat dilihat pada Tabel 4.8 di bawah.

Tabel 12. Hasil Estimasi Usaha Dan Jadwal Menggunakan 2 Perangkat Lunak

Estimasi	Perangkat Lunak	
	Online Freeware	Software Modifikasi
Usaha	53,8	53,7
Jadwal	21,8	19,7

Perbedaan hasil estimasi usaha dan jadwal antara menggunakan *online freeware* dengan *software modifikasi* disebabkan oleh dasar teori yang berbeda yang digunakan sebagai dasar dalam pembuatan perangkat lunaknya. Perbedaan tersebut dapat dilihat pada subbab deskripsi *online freeware* dan *software modifikasi* di atas. Salah satu contohnya adalah pada masukan faktor skala. Masukan faktor skala pada *online freeware* dilakukan dengan memilih salah satu opsi bobot yang masing-masing opsinya mempunyai nilai yang tetap. Sedangkan pada *software modifikasi* masukan berupa *textboxes* yang harus diisi nilainya sesuai dengan penghitungan dengan cara merata-rata setiap faktor skala sesuai data yang terkumpul.

### Kesimpulan

1. Dengan metode COCOMO II dapat dihitung perkiraan atau estimasi usaha atau biaya dan jadwal atau durasi waktu suatu proyek perangkat lunak.
2. Pada proyek perangkat lunak yang baru pertama kali dilakukan, *size* yang digunakan berasal dari konversi hasil penghitungan *Unadjusted Function Point* (UFP) ke dalam bentuk *Source Line Of Code* (SLOC).
3. Pada proyek perangkat lunak yang sudah pernah dikerjakan dan dimaksudkan untuk pengembangan, *size* yang digunakan berasal dari penghitungan baris kode sumber secara manual atau menggunakan bantuan *freeware*.
4. Hasil konversi UFP ke SLOC sangat jauh bedanya dengan penghitungan SLOC melalui penghitungan manual ataupun dengan bantuan *freeware*. Hal disebabkan oleh UFP kurang memperhatikan kode sumber untuk pembentukan antarmuka.
5. Hasil penghitungan SLOC secara manual berbeda dengan hasil yang didapatkan melalui *freeware*. Hal ini disebabkan oleh dasar yang digunakan oleh *freeware* adalah *Line Of Code* (LOC), sedangkan penghitungan manual dasarnya adalah *Logical Line Of Code* (LLOC).

### Saran

1. Dengan telah dikemukakannya metode estimasi biaya dan jadwal proyek perangkat lunak COCOMO II, maka diharapkan penelitian atau perencanaan proyek dapat menggunakan COCOMO II sebagai dasar estimasi.
2. Untuk mempermudah proses estimasi COCOMO II, sangat perlu dilakukan pengembangan *software* terpadu memfasilitasi penghitungan usaha dan jadwal, UFP, dan SLOC (LLOC).

### Daftar Rujukan

- [1]. J. Baik, B. Boehm, and B.M. Steece, *Disaggregating and Calibrating CASE Tool Variable in COCOMO II*, IEEE vol 28 no 11, 2002.
- [2]. R. Banker, R. Chang, and R. Kemerer, *Evidence on Economies of Scale in Software Development*, Information and Software Technology, 1994.
- [3]. B. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy, and R. Selby, *Cost Models for Future Life Cycle Processes: COCOMO II*, Science Publisher, Amsterdam, 1995.
- [4]. Centre of Software Engineering, *COCOMO II Model Definition Manual*, Software Engineering Department of USC, California, 1997.
- [5]. B. Clark, S.D. Chulani, and B.Boehm, *Calibrating the COCOMO II Post- Architecture Model*, IEEE, 1998.
- [6]. IFPUG, *IFPUG Function Point Counting Practices: Manual Release 4.0*, International Function Point Users' Group, Westerville, 1994.
- [7]. R. Kauffman, and R. Kumar, *Modeling Estimation Expertise in Object Based ICASE Environments*, Stern School of Business Report, New York University, 1993.
- [8]. D. Milicic, *A Case Study-Appling COCOMO II*, Institut Teknologi Blackinge, Swedia, 2004.
- [9]. R. Park, *Software Size Measurement: A Framework for Counting Source Statements*, CMU/SEI-92-TR-20, Software Engineering Institute, Pittsburg, 1992.
- [10]. J.W. Paul, and T. Shepard, *Post-Architecture COCOMO II*, IEEE, Canada, 2005.
- [11]. I. Sommerville, *Software Cost Estimation*, Software Engineering 7<sup>th</sup> chapter 26, 2004.
- [12]. R. Sarno, J.L. Biliiali, and S. Maimunah, *Pegembangan Metode Analogy Untuk Estimasi Biaya Rancang Bangun Perangkat Lunak*, Makara Teknologi Vol. 6 No. 2, 2002.
- [13]. -, *Line Of Code Metrics (LOC)*, <http://www.aivosto.com>, November 2008.
- [14]. -, *COCOMO*, <http://www.matabumi.com>, November 2008.
- [15]. R. Madachy, *COCOMO II With Heuristic Risk Assesment*, <http://www.sunset.usc.edu>, November 2008.

**BIOGRAFI PENULIS**

Aderian Primaraka, lahir di Surakarta, Jawa Tengah, 29 Juni 1984. Menempuh pendidikan di SD Negeri Sompok 03 Semarang, SLTP Negeri 3 Semarang, dan SMU Negeri 3 Semarang. Saat ini sedang menyelesaikan pendidikan program Strata 1 Jurusan Teknik Elektro Universitas Diponegoro

dengan mengambil konsentrasi teknik informatika dan komputer. Topik Tugas Akhir yang diambil tentang Estimasi biaya dan jadwal pranglat lunak.

Menyetujui dan mengesahkan,

Dosen Pembimbing I

Eko Handoyo, S.T., M.T.  
NIP. 132 309 142  
Tanggal.....

Dosen Pembimbing II

R. Rizal Isnanto, S.T., M.M., M.T.  
NIP. 132 288 515  
Tanggal.....