

## ESTIMASI PENENTUAN HARGA PERKIRAAN SENDIRI MENGUNAKAN USE CASE POINT UNTUK PENGEMBANGAN PERANGKAT LUNAK SKALA KECIL MENENGAH DENGAN MODEL AGILE

Aris Juliyono<sup>1)</sup> Teguh Sutanto<sup>2)</sup> Sholiq<sup>3)</sup>

S1 / Jurusan Sistem Informasi Kekhususan Komputerisasi Akuntansi

Institut Bisnis dan Informatika Stikom Surabaya

Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email: 1)[11410110035@stikom.edu](mailto:11410110035@stikom.edu), 2)[teguh@stikom.edu](mailto:teguh@stikom.edu), 3)[sholiq@is.its.ac.id](mailto:sholiq@is.its.ac.id)

**Abstrak:** Each application of the procurement of goods / services required the manufacture of Self-Estimated price derived from the approximate (estimated) price of a job (goods / services) to be held. This above is in accordance with Indonesian Presidential Regulation No. 70 Year 2012 on Procurement of Goods / Services, Article 66 digits (5) points a. But what is happening now in the procurement of many software companies individuals or businesses, government officials and developers of software that performs supply and demand for software procurement, had difficulty in making the report HPS. So it can be deduced to solve the problems occurred is needed to set the benchmark in the determination Cost Self-Estimated (HPS) method Use Case Point in software development projects by activity agile models. model testing is done using 5 software development project data gets deviation between the estimated level of HPS with the actual cost by 10%, which indicate that the models estimated in this research has a good degree of accuracy.

**Keywords:** *Use Case Points (UCP), Self-Estimated Price, Software Development Project, Effort Estimation*

Pada tahun 2015 belanja TI (teknologi informasi) di Indonesia telah meningkat menjadi Rp 176,3 T, naik 15,1% dari tahun 2014 (BMI research, 2015). Dan salah satu kebutuhan belanja TI tersebut digunakan untuk pengadaan perangkat lunak (*software*). Pada dasarnya bahwa untuk setiap pelaksanaan pengadaan barang/jasa perlu untuk dibuatkan Harga Perkiraan Sendiri yang merupakan hasil perkiraan (estimasi) harga suatu pekerjaan (barang/jasa) yang akan diadakan, hal ini sesuai dengan Peraturan Presiden Republik Indonesia No. 70 Tahun 2012 Tentang Pengadaan Barang/Jasa, pada Pasal 66 Angka (5) Butir a. Adapun maksud dan tujuan disusunnya HPS adalah supaya harga atau nilai proyek tersebut dalam batas kewajaran dan untuk menetapkan besaran tambahan nilai jaminan pelaksanaan bagi penawaran yang dinilai terlalu rendah.

Namun yang terjadi saat ini dalam pelaksanaan pengadaan perangkat lunak banyak perusahaan perorangan maupun badan usaha, pejabat pemerintah dan pengembang perangkat lunak yang melakukan permintaan dan penawaran untuk pengadaan perangkat lunak kesulitan dalam pembuatan HPS, sehingga apabila penetapan harga dalam pengadaan perangkat lunak terlalu mahal dari harga wajar maka akan menimbulkan potensi kerugian, akan tetapi apabila ditetapkan lebih rendah dari harga wajar maka berpotensi terjadinya kegagalan dalam pengadaan perangkat lunak karena pengembang perangkat lunak tidak akan berminat. Sehingga dapat disimpulkan untuk mengatasi permasalahan yang terjadi tersebut maka perlu dibuat acuan dalam penentuan Harga Perkiraan Sendiri (HPS) dalam proyek pengembangan perangkat lunak menggunakan

metode yang tepat. Metode yang akan digunakan dalam penelitian ini adalah metode use case point (UCP) karena metode UCP memiliki kemampuan untuk memberikan estimasi *effort* dan biaya yang diperlukan untuk membuat suatu proyek perangkat lunak berdasarkan jumlah dan kompleksitas *use case* yang dimiliki oleh proyek tersebut (Karner, 1993).

Dalam metode UCP, estimasi *effort* didapatkan dari perkalian antara nilai UCP dengan nilai *Effort Rate* (ER). Nilai *effort rate* sebesar 20 man-hours dengan menggunakan tiga data proyek pengembangan perangkat lunak (Karner, 1993). Dengan diketahuinya nilai dari estimasi *effort* tersebut, maka dapat dilanjutkan untuk perhitungan selanjutnya, yaitu perhitungan estimasi biaya. Dimana untuk perhitungan estimasi biaya diperlukan persentase aktivitas *effort*, dimana dalam aktivitas *effort* 3 kelompok aktivitas yaitu *software development*, *ongoing activity*, *quality and testing*. Masing-masing kelompok tersebut mempunyai segmentasi peran dan presentase nilai *effort* yang berbeda (Shaleh, 2011).

Pengelompokan sub aktivitas di aktivitas *software development* dalam penelitian ini akan menggunakan model *agile*, dimana dalam *agile software development* mempunyai kelebihan yaitu :

1. 82% menambah produktivitas tim,
2. 77% menambah kualitas perangkat lunak,
3. 78% menambah kepuasan klien,
4. 37% menghemat biaya (Silvelburg, 2002).

Setelah diketahui persentase aktivitas *effort* maka bisa dilakukan perhitungan estimasi biaya dengan mengkalikannya dengan estimasi *effort*. Langkah terakhir yaitu dapat dilakukan perhitungan Total Biaya HPS dalam proyek pengembangan perangkat lunak.

Selain belum adanya acuan dalam penentuan HPS untuk pengembangan perangkat lunak, permasalahan lain yang timbul dalam penelitian ini yaitu belum adanya penentuan sub aktivitas yang terjadi dan besarnya persentase nilai *effort* dalam pengembangan perangkat lunak skala kecil menengah di Indonesia. Dalam beberapa penelitian yang terjadi menggunakan dasar penentuan sub aktivitas pengembangan perangkat lunak berdasarkan penelitian milik Shaleh, dimana penelitian milik Shaleh subaktivitas yang digunakan untuk pengembangan perangkat lunak adalah skala menengah ke besar. Berdasarkan penjabaran

tersebut perlunya dibuat penentuan sub aktivitas dan Harga Perkiraan Sendiri (HPS) pengembangan perangkat lunak skala kecil menengah di Indonesia dengan aktivitas utama mengacu pada penelitian milik Shaleh.

Sehingga dapat diketahui tingkat akurasi dari model perhitungan estimasi harga perkiraan sendiri pada penelitian ini dengan aktual cost, dan model perhitungan dapat dijadikan acuan dalam menghitung estimasi harga perkiraan sendiri untuk pengembangan perangkat lunak.

## LANDASAN TEORI

### Harga Perkiraan Sendiri (HPS)

HPS merupakan total harga yang diperkirakan cukup untuk membiayai pekerjaan yang akan dilaksanakan dalam pengadaan barang/jasa, dan ditetapkan oleh PPK, kecuali untuk Kontes/Sayembara.

Komponen Harga Perkiraan Sendiri (HPS) untuk pengembangan perangkat lunak terdiri dari biaya-biaya, sebagai berikut :

1. Biaya Lagsung Personil (Remunerasi)
2. Biaya Langsung Non Personil, yang terdiri dari :
  - a. *Reimbusable*
  - b. *Fixed Unit Rate*
  - c. *Lump Sum*
3. Pajak Pertambahan Nilai (PPN)

### Estimasi *Effort*

*Effort* (usaha) dari sebuah proyek pengembangan perangkat lunak dapat didefinisikan sebagai waktu yang dikonsumsi oleh proyek yang dinyatakan dengan hitungan orang dalam jam, hari, bulan atau tahun tergantung pada ukuran proyek, sebagai contoh adalah  $effort = people * time$  (Chatters, 1999) dalam (Haapio, 2011).

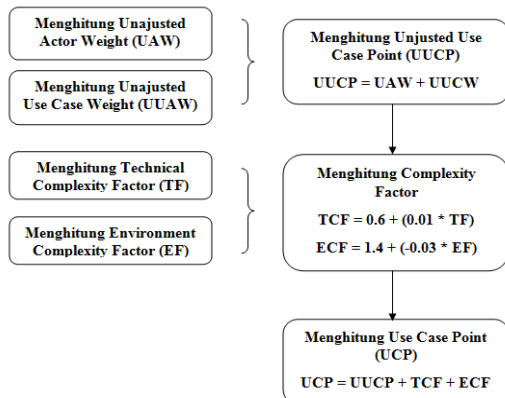
Dari pengertian estimasi dan *effort* di atas, maka dapat disimpulkan bahwa estimasi *effort* adalah suatu kegiatan melakukan prediksi atau ramalan mengenai berapa banyak pekerja dan berapa lama waktu yang diperlukan untuk menyelesaikan proyek tersebut. Estimasi *effort* pada penelitian ini akan didapatkan setelah melakukan perhitungan menggunakan metode use case point (UCP).

### Use Case Point (UCP)

Metode *use case point* (UCP) adalah metode yang mempunyai kemampuan untuk

memberikan estimasi effort yang diperlukan untuk membuat suatu proyek berdasarkan jumlah dan kompleksitas *use case* yang dimiliki oleh proyek perangkat lunak tersebut (Karner, 1993).

Langkah-langkah yang dilakukan dalam proses estimasi *effort* dengan *use case point* digambarkan dalam gambar 1. berikut ini (Karner, 1993) :



Gambar 1. Langkah-langkah Metode *Use Case Point* (UCP)

**Menghitung Unadjusted Use Case Point (UUCP)**

**a. Unadjusted Actor Weights (UAW)**

Langkah pertama adalah menentukan terlebih dahulu aktor sebagai *simple*, *average*, atau *complex* sesuai tabel 1. seperti berikut:

Tabel 1. Tipe, Bobot, dan Deskripsi Actor

Actor	Weight	Description
Simple	1	Didefinisikan dengan API
Medium	2	Berinteraksi dengan protokol TCP/IP
Complex	3	Berinteraksi dengan GUI atau Web Page

Total *Unadjusted Actor Weights* (UAW) didapat dari menghitung jumlah *actor* dari masing-masing jenis (tingkat kompleksitas), dikali dengan total faktor berat masing-masing sesuai dengan tabel.

**b. Unadjusted Use Case Weights (UUCW)**

Cara menghitung UUCW sama dengan cara menghitung UAW, yaitu masing-masing *use case* dibagi menjadi 3 kelompok yaitu *simple*, *average*, dan *complex*, tergantung dari jumlah transaksi yang dilakukan. Untuk penjelasan lebih detail tentang deskripsi *use case* dapat dilihat pada tabel 2. seperti berikut :

Tabel 2. Tipe, Bobot, dan Deskripsi Use Case

Use Case	Weight	Description
Simple	5	Menggunakan <=3 transaksi
Medium	10	Menggunakan 4 sampai 7 transaksi
Complex	15	Menggunakan >7 transaksi

Total *Unadjusted Use Case Weights* (UUCW) didapat dari menghitung jumlah *use case* dari masing-masing tingkat kompleksitas dikali dengan total faktor setiap *use case*. Kemudian jumlahkan UAW dan UUCW untuk mendapatkan *Unadjusted Use Case Point* (UUCP), seperti rumus berikut :

$$UUCP = UAW + UUCW$$

**Menghitung Technical Complexity Factor (TCF)**

Tabel 3. *Technical Factor* dan Bobot

	Technical Factor	Bobot
1	Distributed System Required	2
2	Response time is Important	1
3	End User Efficiency	1
4	Complex Internal Processing Required	1
5	Reusable Code Must Be a Focus	1
6	Installation Easy	0.5
7	Usability	0.5
8	Cross-Platform Support	2
9	Easy to Change	1
10	Highly Concurrent	1
11	Custom Security	1
12	Dependence on Third-part Code	1
13	User Training	1

Nilai-nilai pada *technical factor* tersebut dikalikan dengan bobot nilai masing-masing. Bobot nilai yang diberikan pada setiap *factor* tergantung dari seberapa besar pengaruh dari *factor* tersebut. 0 berarti tidak mempengaruhi, 3 berarti rata-rata, dan 5 berarti memberikan pengaruh yang besar.

Hasil perkalian nilai dan bobot tersebut kemudian dijumlahkan untuk mendapatkan total *Technical Factor* (TF), yang kemudian digunakan untuk mendapatkan *Technical Complexity Factor* (TCF).

$$TCF = 0.6 + (0.01 * TF)$$

**Environmental Complexity Factor (ECF)**

Tabel 4. *Environmental Factor* dan Bobot

	Environmental Factor	Bobot
1	Familiarity with the Project	1.5
2	Application Experience	0.5
3	OO Programming Experience	1
4	Lead Analyst Capability	0.5
5	Motivation	1
6	Stable Requirements	2
7	Part Time Staff	-1
8	Difficult Programming Language	-1

Nilai-nilai pada *environmental factor* tersebut dikalikan dengan bobot nilai masing-masing. Bobot nilai yang diberikan pada setiap factor tergantung dari seberapa besar pengaruh dari faktor tersebut. 0 berarti tidak mempengaruhi, 3 berarti rata-rata, dan 5 berarti memberikan pengaruh yang besar.

Hasil perkalian nilai dan bobot tersebut kemudian dijumlahkan untuk mendapatkan total *Environmental Factor* (EF), yang kemudian digunakan untuk mendapatkan *Environmental Complexity Factor* (ECF).

$$ECF = 1.4 + (-0.03x EF)$$

Sehingga akhirnya kita bisa mendapatkan nilai dari *Use Case Point* (UCP) yang didapatkan melalui perkalian UUCP, TCF, dan ECF.

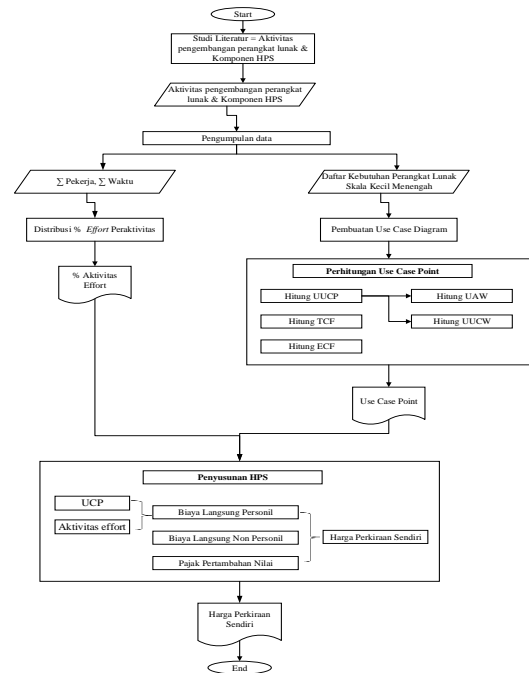
$$UCP = UUCP * TCF * ECF$$

**Agile Development**

Kata *Agile* berarti bersifat cepat, ringan, bebas bergerak, waspada. Kata ini digunakan sebagai kata yang menggambarkan konsep model proses yang berbeda dari konsep model-model proses yang sudah ada (Beck, 2000). Konsep *Agile Software Development* dicetuskan oleh Kent Beck dan 16 rekannya.

Dalam *Agile Software Development* interaksi dan personel lebih penting dari pada proses dan alat, software yang berfungsi lebih penting daripada dokumentasi yang lengkap, kolaborasi dengan klien lebih penting dari pada negosiasi kontrak, dan sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana.

**METODE PENELITIAN**



Gambar 2. Tahapan Penelitian Tugas Akhir

**HASIL DAN PEMBAHASAN**

**Nilai Actual Effort dan Actual Cost**

Pada penelitian tugas akhir ini, studi kasus yang diteliti yaitu proyek pengembangan perangkat lunak. Untuk mempermudah proses penyusunan tabel maka dilakukan pemberian inisialisasi kepada masing-masing proyek pengembangan perangkat lunak. Daftar proyek pengembangan perangkat lunak antara lain dapat dilihat pada Tabel 4.1 seperti berikut:

Tabel 5. Inisialisasi Nama Proyek

Kode	Nama Proyek
A	Website Resmi Pemerintahan Bekasi
B	Website Dinas Kesehatan Cimahi
C	Kepegawaian Bank Mandiri
D	E-Procurement PDAM Kota Tangerang
E	Bursa Kerja Online Dinas Tenaga Kerja Pemerintahan Bogor

Setelah dilakukan inisialisasi kepada masing-masing nama proyek pengembangan perangkat lunak, maka selanjutnya dapat dilakukan pembuatan tabel nilai mandays dan *actual cost*. Nilai mandays dan *actual cost* didapatkan dari hasil survey kuesioner dan wawancara kepada team pengembang proyek. Nilai *actual effort* dan *actual cost* dapat dilihat pada tabel 6. Seperti berikut :

Tabel 6. Nilai Actual Effort dan Actual Cost

No	Kode Proyek	Actual Effort	Actual Cost	Dokumen Terk:
1	A	446	677,xxx,xxx	Lampiran A.1
2	B	458	564,xxx,xxx	Lampiran A.2
3	C	483	385,xxx,xxx	Lampiran A.3
4	D	425	415,xxx,xxx	Lampiran A.4
5	E	447	347,xxx,xxx	Lampiran A.5

**Distribusi Effort**

Pendistribusian nilai persentase effort ke dalam masing-masing aktivitas pengembangan perangkat lunak skala kecil menengah dapat dilakukan setelah diketahui nilai *actual effort*. Pendistribusian nilai persentase *effort* rata-rata dari 5 proyek dapat dilihat pada tabel 7. Seperti berikut :

Tabel 7. Distribusi Effort

		Jabatan Pekerja (% Effort)				
		Project Manager	System analyst	Web/System Design	Programmer	System Testing
<b>1) Pengujian Kebutuhan (Evolving)</b>						
<b>Fase Pengembangan</b>						
	Survey ke SKPD terkait	1.9%	1.0%	1.1%		
	Analisis proses bisnis aplikasi berdasarkan kebutuhan	0.8%	7.6%	2.0%		
	Rapat hasil analisis dengan tim pengembang	3.2%	1.9%	3.0%	1.8%	
	<b>Total</b>	<b>6.0%</b>	<b>10.5%</b>	<b>6.1%</b>	<b>1.8%</b>	
<b>2) Iteration Development</b>						
<b>Iterasi 1</b>						
<b>a) Iterasi Pemecahan</b>						
	Rapat estimasi jadwal dan pekerjaan	2.7%	1.6%	2.4%	1.8%	5.6%
	Pembagian SDM dan jobdesk	1.3%	0.8%	1.2%	0.9%	2.8%
	Penyusunan dokumen SKPL	4.5%	1.8%			
	Rapat hasil analisis dengan stakeholder	1.3%	0.8%	0.6%		
	<b>Total</b>	<b>9.9%</b>	<b>5.1%</b>	<b>4.3%</b>	<b>2.8%</b>	<b>8.3%</b>
<b>b) Model Storing</b>						
	Pengidentifikasi masalah yang akan dipecahkan setiap tim	3.5%	1.8%	2.9%	2.4%	5.6%
	Benchmark template aplikasi	3.0%	1.6%	2.9%	2.2%	2.8%
	Perancangan desain sesuai dengan pengidentifikasi kebutuhan user		6.6%	6.0%		
	Pembuatan prototipe		1.9%	8.5%	7.9%	
	Rapat kesepakatan desain prototip dengan stakeholder	2.7%	1.6%	2.4%	1.5%	
	<b>Total</b>	<b>9.1%</b>	<b>13.6%</b>	<b>22.7%</b>	<b>14.0%</b>	<b>8.3%</b>
<b>c) Implementasi via Test Driven Development</b>						
	Sosialisasi Prototip	1.3%	0.8%	1.2%	0.9%	1.4%
	Ekskusi kode program				10.5%	
	Uji Coba :					
	- Checklist integrasi sistem		4.0%	1.7%	1.3%	11.1%
	- User acceptance test plan		3.4%	2.1%	1.3%	11.1%
	- Pengujian dengan blackbox		3.8%	2.3%	3.1%	8.3%
	Rapat internal	2.7%	1.6%	2.4%	1.8%	5.6%
	Revisi Program					5.5%
	Rapat dengan stakeholder	1.3%	0.8%	1.2%	0.7%	
	<b>Total</b>	<b>5.4%</b>	<b>14.5%</b>	<b>10.9%</b>	<b>25.2%</b>	<b>37.5%</b>
	<b>Total</b>	<b>24.4%</b>	<b>33.2%</b>	<b>37.9%</b>	<b>42.0%</b>	<b>54.2%</b>
<b>Iterasi 2</b>						
<b>a) Iterasi Pemecahan</b>						
	Rapat estimasi jadwal dan pekerjaan	1.6%	0.8%	1.4%	1.1%	2.8%
	Pembagian SDM dan jobdesk	1.3%	0.8%	1.2%	0.9%	2.8%
	Penyusunan dokumen SKPL yang telah direvisi	2.7%	1.0%			
	Rapat hasil analisis dengan stakeholder	1.3%	0.8%	0.4%		
	<b>Total</b>	<b>7.0%</b>	<b>3.5%</b>	<b>3.1%</b>	<b>2.0%</b>	<b>5.6%</b>
<b>b) Model Storing</b>						
	Pengidentifikasi masalah yang akan dipecahkan setiap tim	1.6%	0.8%	1.4%	1.1%	2.8%
	Benchmark template aplikasi	1.3%	0.8%	1.4%	1.1%	1.4%
	Perancangan desain sesuai dengan pengidentifikasi kebutuhan user		3.9%	2.9%		
	Pembuatan prototipe yang telah direvisi		1.4%	3.4%	4.1%	
	Rapat kesepakatan desain prototip dengan stakeholder	1.3%	0.8%	1.2%	0.7%	
	<b>Total</b>	<b>4.3%</b>	<b>7.8%</b>	<b>10.4%</b>	<b>7.0%</b>	<b>4.2%</b>
<b>c) Implementasi via Test Driven Development</b>						
	Sosialisasi Prototip	1.3%	0.8%	1.2%	0.9%	
	Ekskusi kode program				5.7%	
	Uji Coba :					
	- Checklist integrasi sistem		2.5%	1.3%	0.7%	5.6%
	- User acceptance test plan		2.5%	1.3%	0.7%	5.6%
	- Pengujian dengan blackbox		3.7%	1.7%	1.7%	5.6%
	Rapat internal	1.3%	0.8%	1.2%	0.9%	2.8%
	Revisi Program					3.5%
	Rapat dengan stakeholder	1.3%	0.8%	1.2%	0.7%	
	Peluncuran versi beta	1.3%	0.8%	2.3%	2.2%	
	Penyusunan user guide versi beta		3.7%	2.4%		
	<b>Total</b>	<b>5.4%</b>	<b>15.7%</b>	<b>12.5%</b>	<b>17.2%</b>	<b>19.4%</b>
	<b>Total</b>	<b>16.7%</b>	<b>26.9%</b>	<b>26.0%</b>	<b>26.2%</b>	<b>29.2%</b>
	<b>Total</b>	<b>41.1%</b>	<b>60.1%</b>	<b>63.9%</b>	<b>68.1%</b>	<b>83.3%</b>
<b>3) Penerimaan &amp; Penyebaran</b>						
	Rapat penitn levisan aplikasi oleh stakeholder	3.2%	1.9%	2.9%	2.2%	
	Instalasi ke server SKPD				3.0%	
	User training ke SKPD		6.8%	5.6%	4.6%	
	Penyusunan dokumen laporan akhir	3.2%				
	Serah terima aplikasi, database, dan laporan akhir	3.0%	1.8%	1.5%		
	<b>Total</b>	<b>9.4%</b>	<b>10.5%</b>	<b>10.0%</b>	<b>9.8%</b>	
<b>Fase Selang Berlangsung</b>						
<b>1) Manajemen Proyek</b>						
	Penentuan target proyek	4.6%	2.4%	4.1%	2.8%	8.3%
	Rekrutmen programmer	6.4%				
	Penghitungan gaji upah pegawai	5.4%	0.2%			
	Penyusunan dokumen penawaran	6.2%				
	Penyusunan dokumen perencanaan proyek	6.2%				
	Evaluasi tengah proyek	4.3%	0.8%	1.2%	1.7%	
	Rapat penutupan proyek	1.3%	0.8%	1.2%	0.9%	2.8%
	<b>Total</b>	<b>34.4%</b>	<b>4.3%</b>	<b>6.5%</b>	<b>5.4%</b>	<b>11.1%</b>
<b>2) Manajemen Konfigurasi</b>						
	Pengaturan konfigurasi server internal			1.3%	3.9%	
	Pengaturan konfigurasi server SKPD	0.8%			2.9%	
	Uji coba akses aplikasi ke server	2.7%	1.6%	2.6%	2.0%	5.6%
	Penyusunan dokumen konfigurasi	1.3%	1.6%	1.2%		
	<b>Total</b>	<b>4.8%</b>	<b>3.3%</b>	<b>5.1%</b>	<b>8.8%</b>	<b>5.6%</b>
<b>3) Penjaminan Mutu</b>						
	Perencanaan aplikasi oleh tim Diskrenatif		2.1%	1.0%	2.0%	
	Perundatanan software quality	1.3%			0.9%	
	Maintenance aplikasi secara berkala			1.3%	3.1%	
	<b>Total</b>	<b>1.3%</b>	<b>2.1%</b>	<b>2.3%</b>	<b>6.1%</b>	<b>0.0%</b>
<b>4) Pendokumentasian</b>						
	User Guide Aplikasi		4.2%	2.7%		
	Video Tutorial		3.0%	3.5%		
	Laporan Akhir Proyek		3.0%			
	<b>Total</b>	<b>3.0%</b>	<b>9.1%</b>	<b>6.1%</b>	<b>0.0%</b>	<b>0.0%</b>
	<b>TOTAL</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>	<b>100%</b>

**Perhitungan Use Case Point**

**1. Unadjusted Use Case Point (UUCP)**

Untuk mendapatkan nilai *unadjusted use case point*, maka perlu dilakukan pembobotan dan skoring terkait kompleksitas *actor* dan *use case* ditinjau dari *use case diagram* yang telah didapatkan dari tim pengembang perangkat lunak. Skoring dihitung berdasarkan parameter-parameter yang telah ditentukan. Nilai dari UUCP dapat dilihat pada tabel 8. Seperti berikut :

Tabel 8. Nilai *Unadjusted Use Case Point* (UUCP)

Nama Proyek	UAW	UUCW	UUCP
Website Resmi Pemerintahan Bekasi (A)	12	325	33
Website Dinas Kesehatan Cimahi (B)	12	290	30
Bursa Kerja Online Dinas Tenaga Kerja Pemerintah Bogor (C)	12	265	27
Kepegawaian Bank Mandiri (D)	9	185	19
E-Procurement PDAM Kota Tangerang (E)	9	150	15

**2. Technical Complexity Factor (TCF)**

Untuk mengetahui *technical complexity factor*, telah ada beberapa parameter pengukuran dengan disertai bobotnya. Namun untuk pemberian nilai terhadap masing-masing parameter tersebut membutuhkan penilaian obyektif yang didapatkan melalui tim pengembang perangkat lunak. Nilai dari TCF dapat dilihat pada tabel 9. Seperti berikut :

Tabel 9. *Technical Complexity Factor* (TCF)

No	Technical Factor	Bobot (B)	Kode Proyek Perangkat Lunak									
			A		B		C		D		E	
			Nilai (N)	B x N	Nilai (N)	B x N	Nilai (N)	B x N	Nilai (N)	B x N	Nilai (N)	B x N
1	Distributed System Required	2	5	10	5	10	5	10	4	8	4	8
2	Response Time is Important	1	3	3	3	3	3	4	4	4	4	4
3	End User Efficiency	1	3	3	3	3	3	3	3	3	3	3
4	Complex Internal Processing Required	1	3	3	3	3	2	2	2	2	2	2
5	Reusable Code Must Be A Focus	1	4	4	4	4	2	2	2	2	2	2
6	Installation easy	0.5	3	1.5	3	1.5	2	1	2	1	2	1
7	Usability	0.5	4	2	4	2	5	2.5	5	2.5	5	2.5
8	Cross-platform support	2	2	4	2	4	4	8	2	4	2	4
9	Easy to change	1	3	3	3	3	1	1	2	2	2	2
10	Highly concurrent	1	2	2	2	2	3	3	3	3	3	3
11	Custom security	1	4	4	4	4	2	2	4	4	4	4
12	Dependence on third-part code	1	3	3	3	3	3	3	3	3	3	3
13	User training	1	2	2	4	4	3	3	2	2	2	2
TF			44.5	46.5	44.5	40.5	40.5	40.5	40.5	40.5	40.5	40.5
Nilai TCF = 0.6 + (0.01*TF)			1.045	1.065	1.045	1.005	1.005	1.005	1.005	1.005	1.005	1.005

**3. Environmental Complexity Factor (ECF)**

*Environmental Complexity Factor* didapatkan dari hasil survey kuesioner yang telah dibagikan kepada pihak tim pengembang perangkat lunak. Dimana hasil kuesioner dikalikan dengan masing-masing bobot pada faktor *environmental complexity factor* sesuai dengan ketentuan yang ada. Nilai ECF dapat dilihat pada tabel 10. Berikut :

Tabel 10. *Environmental Complexity Factor* (ECF)

No	Environmental Factor	Bobot (B)	Kode Proyek Perangkat Lunak									
			A		B		C		D		E	
			Nilai (N)	B x N	Nilai (N)	B x N	Nilai (N)	B x N	Nilai (N)	B x N	Nilai (N)	B x N
1	Familiarity with the Project	1.5	5	7.5	5	7.5	5	7.5	3	4.5	3	4.5
2	Application Experience	0.5	4	2	4	2	4	2	3	1.5	3	1.5
3	OO Programming Experience	1	2	2	2	2	4	4	2	2	2	2
4	Lead Analyst Capability	0.5	4	2	4	2	5	2.5	3	1.5	3	1.5
5	Motivation	1	4	4	4	4	5	5	3	3	3	3
6	Stable Requirements	2	3	6	4	8	3	6	2	4	2	4
7	Part Time Staff	-1	4	-4	4	-4	1	-1	1	-1	1	-1
8	Difficult Programming Language	-1	2	-2	2	-2	1	-1	2	-2	2	-2
EF			17.5	19.5	25	13.5	13.5	13.5	13.5	13.5	13.5	13.5
Nilai ECF = 1.4 + (0.03*EF)			0.875	0.815	0.650	0.995	0.995	0.995	0.995	0.995	0.995	0.995

**4. Use Case Point (UCP)**

Setelah diketahui Nilai dari UUCP, TCF, dan ECF nilai UCP dapat dihitung dengan rumus seperti berikut :

$$UCP = UUCP * TCF * ECF$$

Nilai dari UCP pada tabel 11. Seperti berikut :

Tabel 11. *Use Case Point* (UCP)

Kode Proyek	UUCP	TCF	ECF	UCP
A	337	1.045	0.875	308
B	302	1.065	0.815	262
C	277	1.045	0.65	188
D	194	1.005	0.995	194
E	159	1.025	0.995	162

**Estimasi Effort**

Nilai estimasi effort diketahui dengan melakukan perhitungan dengan rumus seperti berikut:

$$Estimasi\ Effort = UCP \times Effort\ Rate$$

Nilai Estimasi effort dari keseluruhan proyek dapat dilihat pada tabel 12. Seperti berikut :

Tabel 12. *Estimasi Effort* (Hour)

UCP	Effort Rate	Estimasi Effort (Hour)
308	20	6160
262	20	5240
188	20	3760
194	20	3880
162	20	3240

**Biaya Langsung Personil**

Penyusunan biaya langsung personil perkativitas dapat dilakukan dengan rumus sebagai berikut :

Biaya langsung personil perkativitas = Estimasi effort perkativitas x biaya/tarif personil  
 Untuk biaya langsung keseluruhan proyek dapat dilihat pada tabel 13. Seperti berikut :



Tabel 13. Biaya Langsung Personil

	Estimasi Biaya (effort)				
	A	B	C	D	E
<b>Face Pengembang (Software Development)</b>					
<b>1 Pengujian Kebutuhan (Ersighting)</b>					
Survey ke SKPD terkait	5.552,287	8.665,450	7.650,783	3.244,593	2.576,054
Analisis proses bisnis aplikasi berdasarkan kebutuhan	12.982,960	13.358,568	7.018,667	6.208,000	3.943,087
Rapat hasil analisis dengan tim pengembang	27.769,723	12.267,092	10.434,584	14.630,795	7.744,107
<b>Total Ersighting</b>	<b>46.304,969</b>	<b>34.291,109</b>	<b>23.330,232</b>	<b>19.186,372</b>	<b>11.671,248</b>
<b>2 Iteration Development</b>					
<b>Iterasi 1</b>					
<b>a Iterasi Pemodelan</b>					
Rapat estimasi jadwal dan pekerjaan	20.601,471	17.065,467	10.434,584	9.753,864	7.744,107
Pembagian SDM dan jobdesk	10.300,735	8.532,734	5.217,292	4.876,932	3.872,054
Penyusunan dokumen SKPL	14.778,475	12.241,921	7.317,598	9.713,694	9.640,268
Rapat hasil analisis dengan stakeholder	5.552,287	4.599,301	3.825,391	3.244,593	2.576,054
<b>Total</b>	<b>51.232,969</b>	<b>42.439,424</b>	<b>26.794,865</b>	<b>27.589,082</b>	<b>23.832,483</b>
<b>b Model Skoring</b>					
Pengidentifikasi masalah yang akan diselesaikan setiap tin	20.601,471	17.065,467	15.651,876	9.753,864	14.840,215
Benchmark template aplikasi	18.513,148	17.065,467	10.434,584	14.630,795	9.688,107
Perancangan desain sesuai dengan pengidentifikasi kebutuhan user	11.808,969	9.782,096	8.773,333	6.989,478	6.535,087
Pembuatan prototipe	34.573,345	27.659,878	16.732,389	20.438,927	18.144,000
Rapat kesepakatan desain prototip dengan stakeholder	13.574,099	15.335,581	10.434,584	9.753,864	7.744,107
<b>Total</b>	<b>99.071,031</b>	<b>86.908,489</b>	<b>62.026,766</b>	<b>61.566,927</b>	<b>56.951,517</b>
<b>c Implementasi via Test Driven Development</b>					
Sosialisasi Prototip	9.256,574	8.532,734	5.217,292	4.876,932	3.872,054
Ekskusi kode program	24.695,247	24.547,913	19.486,609	14.691,049	15.552,000
Uji Coba					
- Checklist integrasi sistem	12.629,381	10.461,694	5.626,766	11.070,325	4.901,315
- User acceptance test plan	12.629,381	10.461,694	7.018,667	11.070,325	3.915,544
- Pengujian dengan backbox	5.602,009	4.640,489	12.586,269	13.128,094	11.691,544
Rapat internal	20.601,471	17.065,467	10.434,584	9.753,864	7.744,107
Revisi Program	12.347,623	10.228,297	9.743,304	8.161,694	9.072,000
Rapat dengan stakeholder	6.787,049	7.667,790	5.217,292	4.876,932	3.872,054
<b>Total</b>	<b>104.548,735</b>	<b>93.606,079</b>	<b>75.330,783</b>	<b>77.629,214</b>	<b>60.620,617</b>
<b>Total Iterasi 1</b>	<b>254.852,735</b>	<b>222.953,991</b>	<b>164.152,414</b>	<b>166.785,224</b>	<b>141.404,617</b>
<b>Iterasi 2</b>					
<b>a Iterasi Pemodelan</b>					
Rapat estimasi jadwal dan pekerjaan	10.300,735	8.532,734	5.217,292	9.753,864	3.872,054
Pembagian SDM dan jobdesk	10.300,735	8.532,734	5.217,292	4.876,932	3.872,054
Penyusunan dokumen SKPL yang telah direvisi	9.226,188	7.642,620	4.188,157	4.856,847	5.784,161
Rapat hasil analisis dengan stakeholder	5.552,287	4.599,301	3.129,441	3.244,593	2.576,054
<b>Total</b>	<b>35.379,946</b>	<b>29.307,389</b>	<b>17.752,182</b>	<b>22.732,235</b>	<b>16.104,322</b>
<b>b Model Skoring</b>					
Pengidentifikasi masalah yang akan diselesaikan setiap tin	10.300,735	8.532,734	5.217,292	4.876,932	7.744,107
Benchmark template aplikasi	9.256,574	8.532,734	5.217,292	4.876,932	5.816,054
Perancangan desain sesuai dengan pengidentifikasi kebutuhan user	6.226,296	5.157,624	5.264,000	4.115,539	3.267,544
Pembuatan prototipe yang telah direvisi	16.051,910	12.807,109	7.322,269	13.093,402	8.424,000
Rapat kesepakatan desain prototip dengan stakeholder	6.787,049	7.667,790	5.217,292	4.876,932	3.872,054
<b>Total</b>	<b>48.622,565</b>	<b>42.697,991</b>	<b>28.238,145</b>	<b>31.839,736</b>	<b>29.125,758</b>
<b>c Implementasi via Test Driven Development</b>					
Sosialisasi Prototip	9.256,574	7.667,790	5.217,292	4.876,932	3.872,054
Ekskusi kode program	17.286,673	12.273,956	9.743,304	8.161,694	7.776,000
Uji Coba					
- Checklist integrasi sistem	4.557,848	3.775,546	4.901,234	5.747,878	3.267,544
- User acceptance test plan	4.557,848	3.775,546	4.901,234	5.747,878	3.267,544
- Pengujian dengan backbox	4.557,848	3.775,546	6.293,135	7.380,216	9.127,087
Rapat internal	10.300,735	8.532,734	5.217,292	4.876,932	3.872,054
Revisi Program	9.878,099	8.182,638	5.567,602	4.897,016	5.184,000
Rapat dengan stakeholder	6.787,049	7.667,790	5.217,292	4.876,932	3.872,054
Pelancaran versi beta	11.726,099	9.713,450	7.305,143	9.773,948	7.760,054
Penyusunan user guide versi beta	6.226,296	5.157,624	3.509,333	4.115,539	3.267,544
<b>Total</b>	<b>85.135,067</b>	<b>70.522,620</b>	<b>57.872,861</b>	<b>60.454,965</b>	<b>51.265,933</b>
<b>Total Iterasi 2</b>	<b>169.137,578</b>	<b>142.528,000</b>	<b>103.863,188</b>	<b>115.026,936</b>	<b>96.494,013</b>
<b>Total Iteration Development</b>	<b>423.990,314</b>	<b>365.481,991</b>	<b>268.015,602</b>	<b>281.812,160</b>	<b>237.898,631</b>
<b>Total Software Development</b>	<b>470.295,283</b>	<b>399.773,100</b>	<b>290.335,834</b>	<b>308.995,532</b>	<b>249.569,879</b>
<b>Face Sedang Berlangsung (On Going Activity)</b>					
<b>1 Manajemen Proyek</b>					
Pencarian target proyek	30.902,206	25.496,201	14.529,975	19.507,727	14.192,215
Rekrutmen programmer	14.695,605	12.173,275	14.495,072	7.285,371	11.568,322
Penghasilan gaji upah pegawai	14.695,605	12.173,275	7.270,890	9.713,694	9.640,268
Penyusunan dokumen penawaran	14.695,605	12.173,275	10.353,623	12.142,118	9.640,268
Penyusunan dokumen penawaran proyek	18.369,507	15.216,594	10.353,623	9.713,694	7.712,215
Evaluasi tengah proyek	19.073,001	15.800,087	12.821,366	13.794,313	5.800,107
Rapat penutupan proyek	10.300,735	8.532,734	5.217,292	4.876,704	3.872,054
<b>Total Manajemen Proyek</b>	<b>122.733,166</b>	<b>101.667,441</b>	<b>74.771,843</b>	<b>77.033,520</b>	<b>62.425,450</b>
<b>2 Manajemen Konfigurasi</b>					
Pengaturan konfigurasi server internal	12.347,623	10.228,297	5.567,602	7.345,525	7.128,000
Pengaturan konfigurasi server SKPD	11.082,475	9.180,297	6.246,427	4.897,016	5.184,000
Uji coba akses aplikasi ke server	20.601,471	17.065,467	10.434,584	9.753,864	9.688,107
Penyusunan dokumen konfigurasi	6.226,296	5.157,624	2.117,433	4.856,847	5.784,161
<b>Total Manajemen Konfigurasi</b>	<b>50.257,865</b>	<b>41.631,686</b>	<b>24.366,046</b>	<b>26.853,252</b>	<b>27.784,268</b>
<b>3 Penjaminan Mutu</b>					
Pemeriksaan aplikasi oleh tim Diskonform	8.695,821	7.203,284	4.175,702	5.747,878	5.832,000
Pensadartanganan software quality	6.143,426	5.088,978	3.462,625	4.060,762	3.224,054
Maintenance aplikasi secara berkala	7.408,574	6.136,978	9.047,354	7.345,525	3.888,000
<b>Total Penjaminan Mutu</b>	<b>22.247,821</b>	<b>18.429,240</b>	<b>13.205,930</b>	<b>17.154,165</b>	<b>11.000,054</b>
<b>4 Pendokumentasian</b>					
User Guide Aplikasi	6.226,296	5.157,624	3.509,333	6.173,308	3.267,544
Video Tutorial	7.461,058	6.180,454	5.264,000	6.173,308	4.253,315
Laporan Akhir Proyek	7.347,803	6.086,638	4.141,449	7.285,271	3.856,107
<b>Total Pendokumentasian</b>	<b>21.035,157</b>	<b>17.424,716</b>	<b>12.914,783</b>	<b>19.631,887</b>	<b>11.376,966</b>
<b>Total On going Activity</b>	<b>216.274,099</b>	<b>179.153,083</b>	<b>125.258,606</b>	<b>140.672,824</b>	<b>112.586,738</b>
<b>Quality Assurance &amp; Control</b>					
<b>1 Quality Assurance &amp; Control</b>					
Serah terima aplikasi, database, dan laporan akhir	11.104,574	9.200,754	11.476,174	6.459,186	5.152,107
Instalasi ke server SKPD	7.408,574	6.136,978	4.175,702	3.264,678	3.888,000
User training ke SKPD	27.913,363	21.570,677	18.516,638	14.760,433	11.691,544
<b>Total Quality Assurance &amp; Control</b>	<b>46.426,511</b>	<b>36.908,409</b>	<b>34,168,513</b>	<b>24,514,296</b>	<b>20,731,651</b>
<b>2 Evaluation &amp; Testing</b>					
Rapat penilaian kesiapan aplikasi oleh stakeholder	18.513,148	19.938,108	10.434,584	9.753,864	11.616,161
Penyusunan dokumen laporan akhir	7.347,803	6.086,638	4.141,449	7.285,271	5.784,161
<b>Total Evaluation &amp; Testing</b>	<b>25.860,951</b>	<b>26,024,746</b>	<b>14,576,033</b>	<b>17,039,134</b>	<b>17,400,322</b>
<b>Total Quality &amp; Testing</b>	<b>72,287,462</b>	<b>62,933,155</b>	<b>48,744,547</b>	<b>41,553,431</b>	<b>38,131,973</b>
<b>1 TOTAL</b>	<b>758,856,752</b>	<b>641,859,339</b>	<b>470,602,534</b>	<b>488,121,902</b>	<b>404,824,591</b>

### Biaya Langsung Non Personil

Untuk mendapatkan total biaya langsung non personil, terlebih dahulu harus mengetahui volume atau waktu dari masing-masing kegiatan, kemudian dapat dihitung dengan rumus sebagai berikut :

$$\text{Biaya Langsung Non Personil} = \text{Volume/waktu} \times \text{biaya satuan}$$

Setelah direkap biaya satuan dari masing-masing komponen biaya non personil, maka akan mendapatkan total biaya langsung non personil.

Dalam tugas akhir ini biaya langsung non personil diketahui telah *include* dengan biaya langsung personil.

### Harga Perkiraan Sendiri (HPS)

Setelah total biaya langsung personil dan biaya langsung non personil diketahui maka kedua biaya tersebut di jumlahkan dan akan mendapatkan nilai dari HPS sebelum pajak. Untuk mendapatkan nilai HPS setelah pajak maka harus dilakukan perhitungan dengan rumus sebagai berikut :

HPS setelah pajak = HPS sebelum pajak x PPN  
Nilai HPS dari keseluruhan Proyek dapat dilihat pada tabel 14. Seperti berikut :

Tabel 14. Harga Perkiraan Sendiri (HPS)

Kode Proyek	HPS Sebelum Pajak	PPN	HPS Setelah Pajak
A	758,856,753	75,885,675	834,742,428
B	641,859,339	64,185,934	706,045,272
C	470,602,534	47,060,253	517,662,788
D	488,121,802	48,812,180	536,933,983
E	404,824,591	40,482,459	445,307,050

### Uji Coba

Uji Coba dilakukan dengan mencari nilai deviasi antara harga perkiraan sendiri dengan actual cost jika nilai deviasi < 20% maka dapat disimpulkan bahwa dalam penelitian tersebut mempunyai tingkat akurasi yang cukup baik. Berikut hasil uji coba dapat dilihat pada tabel 15. Sebagai berikut :

Tabel 15. Uji Coba HPS

Kode Proyek	Actual Cost	HPS	Deviasi	%
A	727,000,000	834,742,428	107,742,428	15%
B	664,000,000	706,045,272	42,045,272	6%
C	485,000,000	517,662,788	32,662,788	7%
D	487,000,000	536,933,983	49,933,983	10%
E	396,000,000	445,307,050	49,307,050	12%

Dari hasil uji coba yang telah dilakukan dapat diketahui tingkat deviasi yang didapatkan sebesar 10%, hal ini menunjukkan bahwa model perhitungan pada penelitian ini memiliki tingkat akurasi yang baik.

### SIMPULAN

Kesimpulan yang dapat diambil dari pengerjaan Tugas Akhir ini yakni sebagai berikut:

1. Nilai aktual *effort* dan aktual didapatkan dari hasil wawancara dengan tim pengembang proyek perangkat lunak.
2. Penilaian faktor *technical* dan *environmental* pada metode UCP didapatkan dari hasil kuisisioner yang diberikan kepada pihak pengembang masing-masing proyek perangkat lunak.
3. Model estimasi biaya untuk pengembangan perangkat lunak skala kecil menengah menggunakan metode UCP dan model Agile
4. Pengujian model dilakukan dengan 5 data proyek pengembangan perangkat lunak mendapat penyimpangan tingkat antara estimasi HPS dengan aktual *cost* sebesar 10%, dimana menunjukkan bahwa model estimasi pada penelitian ini mempunyai tingkat akurasi yang baik.

### SARAN

Beberapa hal yang diharapkan dapat dikembangkan untuk penelitian dimasa mendatang, yaitu:

1. Penelitian ini menggunakan model agile dalam penentuan aktivitas di fase pengembangannya, untuk penelitian selanjutnya disarankan melakukan uji coba dengan menggunakan model yang lain.
2. Penelitian ini dapat dilanjutkan ketahap yang selanjutnya menjadi penelitian pembuatan aplikasi dalam penentuan Harga Perkiraan Sendiri.

### RUJUKAN

Ambler, Scott W. *An Introduction to Agile Modelling*. Agile Modeling Home Page. 2001-2008. Online : <http://www.agilemodelling.com/essays/introductionToAM.html>. Diakses pada tanggal 15 Juli 2015.

Anda, B. 2002. *Comparing Effort Estimates Based on Use Case with Expert Estimates*. Empirical Assessment in Software Engineering (EASE), (p. 13). Keele UK.

Barbosa da Silva, C. Monteiro., L. Denis Silva., da Cunha, Adilson Marques. 2008. *Applying The*



- Use Case Points Effort Estimation Technique To Avionics Systems*. IEEE
- Beck, Kent., Jeffries, Ron., and Ward Cunningham. 2000. *Extreme Programming: Embrace Change*. Addison-Wesley.
- BMI Research. 2015. *Indonesia Information Technology Report publish 11 July 2015*. Online : <http://store.bmiresearch.com/indonesia-information-technology-report.html>. Diakses pada tanggal 13 Juli 2015.
- Bull Survey. 1998. *Failure Causes*. Online : [http://www.itcortex.com/Stat\\_Failure\\_Cause.htm#surveys](http://www.itcortex.com/Stat_Failure_Cause.htm#surveys). Diakses pada tanggal 13 Juli 2015
- Carroll, Edward R. 2015. *Estimating Software Based on Use Case Points*. ObjectOriented, Programming, Systems, Languages, and Object Oriented Programming Systems Languages and Applications (OOPSLA) Conference, San Diego, CA, pp.257-265.
- Clemmons, Roy K. 2006. *Project Estimation With Use Case Point*. Diversified Technical Services, Inc.
- Damodaran, Mel. 2002. *Estimation Using Use Case Points*. Computer Science Program, University of Houston-Victoria.
- Frohnhoff, S. 2008. *Revised Use Case Point Method-Effort Estimation in Development Projects for Business Applications*. Offenbach.
- Galorath, D. 2006. *Software Sizing, Estimation and Risk Management*. Auerbach.
- Haapio, Topi. 2011. *Improving Effort Management in Software Development Project*. University of Eastern Finland.
- Ikatan Nasional Konsultasi Indonesia (INKINDO). 2014. *Pedoman standar Minimal 2014 Biaya Langsung Personil dan Biaya Langsung Non Personil untuk Kegiatan Jasa Konsultasi*. Online : <http://www.inkindo.org>. Diakses pada tanggal 01 Agustus 2015
- Karner, Gustav. 1993. *Resource Estimation for Objectory Projects*. Objective Systems SF AB.
- Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintahan. 2012. *Rancangan Pedoman Umum Perencanaan Pengadaan Barang/Jasa Pemerintahan di Lingkungan Kementrian/Lembaga/Satuan Kerja Perangkat Daerah/Institusi Lainnya*. Online : <http://www.lpse.manggaraitimurkab.go.id/edproc/index.filedownload:download/3132383436313b31>. Diakses pada tanggal 15 Juli 2015.
- Monteiro, C., Silva, D., dan Marques, A. 2008. *Applying The Use Case Points Effort Estimation Technique To Avionics Systems*. IEEE.
- Mulder, H., Kontakos, I., & Standish, G. 2015. *Rethinking The Public Spending on ICT projects*. The Standish Group. Boston.
- Nageswaran, Surech. 2001. *Test Effort Estimation Using Use Case Points*. Online: [http://www.cognizant.com/cogcommunity/presentations/Test\\_Effort\\_Estimation.pdf](http://www.cognizant.com/cogcommunity/presentations/Test_Effort_Estimation.pdf). Diakses pada tanggal 13 Juli 2015.
- Ochodek, M., Nawrocki, J., Kwarciak, K. 2011. *Simplifying Effort Estimation Based on Use Case Points*. Sciencedirect.
- Parthasarathy, M.A. 2007. *Partical Software Esimaion : Function Point Methods for Insourced and Outsorced Projects*. Infosys.
- Peraturan Presiden Republik Indonesia No 70 Tahun 2012. Online : <http://www.lkpp.go.id/v2/files/content/file/09082012110709BT%20Perpres%200702012.pdf>. Diakses pada tanggal 15 Juli 2015.
- Rizky. Soetam. 2008. *Konsep Dasar Rekayasa Perangkat Lunak Software Reengineering*. Prestasi Pustaka.
- Saleh, K. 2011. *Effort and Cost Allocation in Medium to Large Software Development Projects*. International Journal of Computers (1), 74-79.
- Silverburg, A. 2002. *Agile Analytics in Higher Education*. USA. Phytorion.
- Schneider, G. And Winters, J. 1998. *Applying Use Cases-A Practical Guide*. Addison-Wesley.

Yavari, Y., Afsharchi, M., & Karami, M. 2011. *Software Complexity Level Determination Using Software Effort Estimation Use Case Points Metrics*. 2011 5th Malaysian Conference in Software Engineering (pp. 257-262). IEEE