

Penggunaan Metode Estimasi *Use Case Points* (UCP) Dalam Proyek *Software Domain Bisnis*

Edhi Prayitno

AMIK BSI Jakarta
e-mail: edhi.epo@bsi.ac.id

Abstrak

Software memegang peranan penting pada sebuah bisnis untuk mempercepat dan mengatur proses bisnis secara lebih baik. Proyek pengembangan software seperti halnya proyek-proyek pada umumnya membutuhkan estimasi yang tepat agar dapat berjalan dengan baik dan lancar. Metode *use case points* digunakan untuk mengestimasi software berbasis obyek dengan tingkat akurasi cukup baik. Nilai *Effort Rate* (ER) yang digunakan dalam metode estimasi UCP memiliki nilai yang bervariasi mulai dari 4 sampai 35 man-hours, sehingga akan membuat sulit pengembang pemula untuk menentukan nilai ER yang tepat. Thomas Tan dalam disertasinya berjudul '*Domain-Based Effort Distribution Model For Software Cost Estimation*' menunjukkan bahwa setiap aplikasi memiliki distribusi effort yang berbeda-beda sesuai dengan karakteristiknya yang berjumlah 21 domain. Dari hasil penelitian yang dilakukan terhadap proyek sistem penjualan barang, didapat nilai aktual effort dan cost mendekati nilai estimasi yang dilakukan. Nilai estimasi effort 381,3 jam dan aktual effort 407.5 jam, sedangkan estimasi cost IDR 39.804.236 dan aktual cost IDR 47.675.417 sehingga didapat nilai deviasi effort 6,4% dan deviasi cost 16,5%.

Kata kunci: Estimasi usaha dan biaya software, *use case point* (UCP), domain aplikasi.

Abstract

Software plays an important role in a business to accelerate and manage business processes better. Software development projects as well as projects generally require the right estimates to work well and smoothly. Use case points method is used to estimate object-based software with good accuracy. The value of the Effort Rate (ER) used in the UCP estimation method has values range from 4 to 35 man-hours, so it will make it difficult for novice developers to determine the exact ER value. Thomas Tan in his dissertation entitled 'Domain-Based Effort Distribution Model For Software Cost Estimation' shows that each application has a different distribution effort according to its 21 domain characteristics. From the results of research conducted on the project sales system goods, obtained the actual value of effort and cost close to the estimated value made. The estimated value of effort is 381.3 hours and 407.5 hours actual effort, while the estimated cost is IDR 39,804,236 and the actual cost is IDR 47,675,417 to get the value of deviation effort 6,4% and deviation cost 16,5%.

Keywords: Estimate effort and cost software, *use case points* (UCP), application domain.

1. Pendahuluan

Proyek software merupakan kegiatan yang membutuhkan keahlian (skill) dan pengetahuan yang cukup memadai bagi developer. Minimnya kompetensi yang dimiliki developer akan berakibat fatal terhadap proyek yang sedang dikerjakan. Tidak sedikitnya jumlah proyek software yang gagal mengindikasikan kegagalan dalam proses estimasi effort dan manajemen yang dilakukan. Sebuah laporan dari Standish Group Study '*CHAOS Summary 2012*', yang menyatakan hanya 39% proyek IT yang sukses diselesaikan

tepat waktu dan biaya serta fungsi dan tampilan aplikasi yang sesuai kebutuhan. Hal senada juga disampaikan dalam laporan yang dirilis oleh Capers Jones, Chief Scientist Emeritus SPR yang mengidentifikasi estimasi biaya formal merupakan faktor penting untuk mencegah kegagalan proyek.

Trendowicz (Trendowicz, 2013) menyatakan bahwa estimasi biaya yang tepat merupakan salah satu faktor keberhasilan bagi manajer proyek dalam mengatur dan menyelesaikan proyek dengan baik. Menurut pendapat ahli yang

lain, estimasi usaha (effort) yang buruk merupakan salah satu dari dua masalah yang secara umum menyebabkan kegagalan proyek disamping kebutuhan yang tidak bisa dikendalikan (unstable requirements). Dan penggunaan alat dan metodologi pengembangan proyek secara formal akan meningkatkan produktivitas kerja dua kali lipat dibandingkan penggunaan secara konvensional.

Pengembangan proyek software yang baik memiliki dua kegiatan kritis yaitu menghitung estimasi usaha/biaya software secara tepat dan melakukan manajemen proyek yang berdasarkan SDLC (Software Development Life Cycle) secara terukur. Untuk point yang pertama; ada beberapa metode estimasi yang dapat digunakan untuk menghitung estimasi effort software, beberapa diantaranya yaitu Function Point (FP), Function Point Analysis (FPA), Delphi, Use Case Points (UCP), Cocomo, Analogy dan sebagainya. Metode Use Case Points (UCP) dikembangkan oleh Gustav Karner tahun 1993. Use Case Points (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 software tersebut (Karner, 1993). Menurut pendapat lain, UCP adalah metode yang dapat menganalisa aktor, use case, dan berbagai faktor teknis dan faktor lingkungan hingga menjadi suatu persamaan (Clemmons, 2006). Kelebihan dari metode use case point yaitu UCP memiliki deviasi sebesar 6% (Nageswaran, 2001), UCP memiliki deviasi sebesar 9% (Carrol, 2005) dan UCP memiliki deviasi sebesar 19%, sementara estimasi para ahli memiliki rata-rata deviasi sebesar 20% (Anda, 2002). Ini menunjukkan metode estimasi UCP memiliki tingkat akurasi yang baik digunakan dalam menentukan effort terhadap suatu proyek software yang akan dikerjakan.

Berikut langkah-langkah menentukan nilai estimasi effort dengan menggunakan Use Case Point (UCP):

1. Menghitung nilai Unadjusted Use Case Point (UUCP), dengan terlebih dahulu menentukan nilai Unadjusted Actor Weights (UAW) dan Unadjusted Use Case Weights (UUCW).

Nilai Unadjusted Actor Weights (UAW) ditentukan dari jumlah aktor yang

diklasifikasikan sebagai simple, average, atau complex sesuai tabel 1 berikut ini.

Tabel 1. Tipe, Bobot, dan Deskripsi Actor

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

Kemudian menghitung jumlah actor dari masing-masing tingkat kompleksitas dikali dengan nilai faktor berat dari masing-masing tingkatan.

Nilai Unadjusted Use Case Weights (UUCW) didapat sama dengan seperti menentukan UAW, yaitu masing-masing use case dibagi menjadi 3 kelompok yaitu simple, average, dan complex, tergantung dari jumlah transaksi yang dilakukan. Berikut klasifikasi use case berdasarkan jumlah transaksinya dapat dilihat pada tabel 2 berikut ini.

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

Nilai Unadjusted Use Case Weights (UUCW) didapat dari menghitung jumlah use case dari masing-masing tingkat kompleksitas dikali dengan nilai faktor berat setiap use case. Setelah nilai UAW dan UUCW didapat maka dijumlahkan untuk mendapatkan Unadjusted Use Case Point (UUCP).

2. Menghitung nilai Technical Complexity Factor (TCF) dan Environmental Complexity Factor (ECF).

Pada perhitungan nilai TCF dan ECF terdapat nilai complexity factor. Complexity factor merupakan faktor-faktor yang berpengaruh secara langsung dalam proses pengembangan proyek perangkat lunak.

a. Technical Complexity Factor (TCF)

Nilai-nilai pada technical factor tersebut dikalikan dengan bobot nilai masing-masing. Bobot nilai yang diberikan pada

setiap faktor tergantung dari seberapa besar pengaruh dari faktor tersebut. Dimana nilai 0 berarti tidak mempengaruhi, nilai 3 berarti rata-rata, dan nilai 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) berdasarkan rumus:

$$TCF=0.6+(0.01 \times TF)$$

Tabel 3. Technical Factor dan Bobot

No.	Technical Factor	Bobot
1.	<i>Distributed System Required</i>	2
2.	<i>Response Time is Important</i>	1
3.	<i>End User Efficiency</i>	1
4.	<i>Complex Internal Processing Required</i>	1
5.	<i>Reusable Code Must Be A Focus</i>	1
6.	<i>Installation easy</i>	0.5
7.	<i>Usability</i>	0.5
8.	<i>Cross-platform support</i>	2
9.	<i>Easy to change</i>	1
10.	<i>Highly concurrent</i>	1
11.	<i>Custom security</i>	1
12.	<i>Dependence on third-part code</i>	1
13.	<i>User training</i>	1

b. Environmental Complexity Factor (ECF)

Sama seperti pada *technical factor*, nilai-nilai pada *environmental factor* tersebut dikalikan dengan bobot nilai masing-

masing. Bobot nilai yang diberikan pada setiap faktor tergantung dari seberapa besar pengaruh dari faktor tersebut. Nilai 0 berarti tidak mempengaruhi, nilai 3 berarti rata-rata, dan nilai 5 berarti memberikan pengaruh yang besar. Hasil perkalian nilai dan bobot tersebut kemudian dijumlahkan untuk mendapatkan *Environmental Factor* (EF), yang kemudian digunakan untuk mendapatkan *Environmental Complexity Factor* (ECF) berdasarkan rumus:

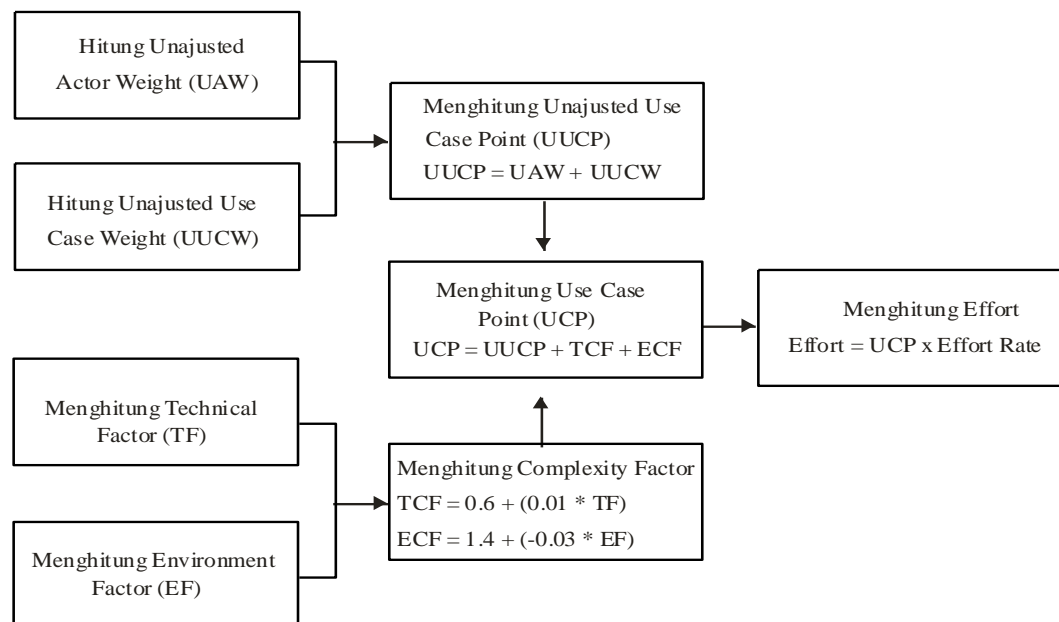
$$ECF=1.4+(-0.03 \times EF)$$

Tabel 4. Environmental Factor dan Bobot

No.	Technical Factor	Bobot
1.	<i>Familiarity with the Project</i>	1.5
2.	<i>Application Experience</i>	0.5
3.	<i>OO Programming Experience</i>	1
4.	<i>Lead Analyst Capability</i>	0.5
5.	<i>Motivation</i>	1
6.	<i>Stable Requirements</i>	2
7.	<i>Part Time Staff</i>	-1
8.	<i>Difficult Programming Language</i>	-1

3. Dan terakhir nilai dari *Use case Point* (UCP) didapatkan melalui perkalian UUCP dengan TCF dan ECF, yaitu $UCP=UUCP+TCF+ECF$.

Berikut gambaran secara mudah penggunaan metode estimasi use case point dalam melakukan estimasi perangkat lunak (Karner, 1993):



Gambar 1. Proses Estimasi Effort dengan Use Case Points

Penggunaan metode estimasi UCP tidak lepas kaitannya dengan *Effort Rate* (ER). *Effort Rate* setiap aplikasi berbeda-beda antara satu pengembang dengan pengembang yang lain. Hal ini disebabkan oleh kemampuan dan kapasitas yang dimiliki oleh pengembang yang jelas berbeda. *Effort Rate* pertama kali digunakan oleh Karner tahun 1993 dan memiliki nilai 20 *man-hours* berdasarkan tiga data proyek pengembangan perangkat lunak (Karner, 1993). Schneider & Winters tahun 1998 mengusulkan nilai ER sebesar 20, 28, dan 30 *man-hours* (Schneider, 1998). Clemmons tahun 2006 memberikan nilai ER sebesar 18 *man-hours* yang berdasarkan kualitas tim proyek dan data historis (Clemmons, 2006). Dan Ochodek tahun 2011 memberikan nilai ER berkisar antara 4 sampai 35 *man-hours* berdasarkan perhitungan 14 proyek perangkat lunak yang telah selesai (Ochodek, 2011).

Setiap perangkat lunak aplikasi yang dikembangkan memiliki jenis dan karakteristiknya berdasarkan dari kelompok domain masing-masing. Menurut Thomas Tan dalam disertasinya dengan judul

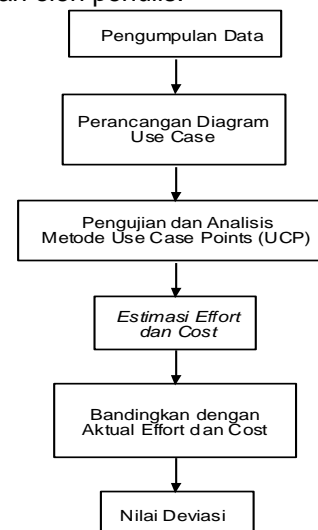
'*Domain-Based Effort Distribution Model For Software Cost Estimation*', mengelompokkan aplikasi ke dalam 21 wilayah domain yang meliputi sistem bisnis, internet, alat dan sistem alat, sains sistem, simulasi dan pemodelan, test dan evaluasi, training, perintah dan kontrol, manajemen misi, alat pertahanan dan pengontrolan,

komunikasi, tampilan dan pengontrolan, infrastruktur atau perangkat perantara, informasi asuransi, maintenance dan diagnostik, perencanaan misi, kontrol proses, sensor pengontrol dan pemrosesan, pesawat ruang angkasa, dan muatan pesawat ruang angkasa. Berdasarkan penelitian aplikasi domain bisnis yang dilakukan di Indonesia (Subriadi dkk, 2014), diketahui nilai *Effort Rate* (ER) untuk software aplikasi bisnis di Indonesia adalah 8.2 *man-hours*.

2. Metode Penelitian

Desain Penelitian

Berikut gambaran desain penelitian yang dilakukan oleh penulis:



Gambar 2. Desain Penelitian

Data Penelitian

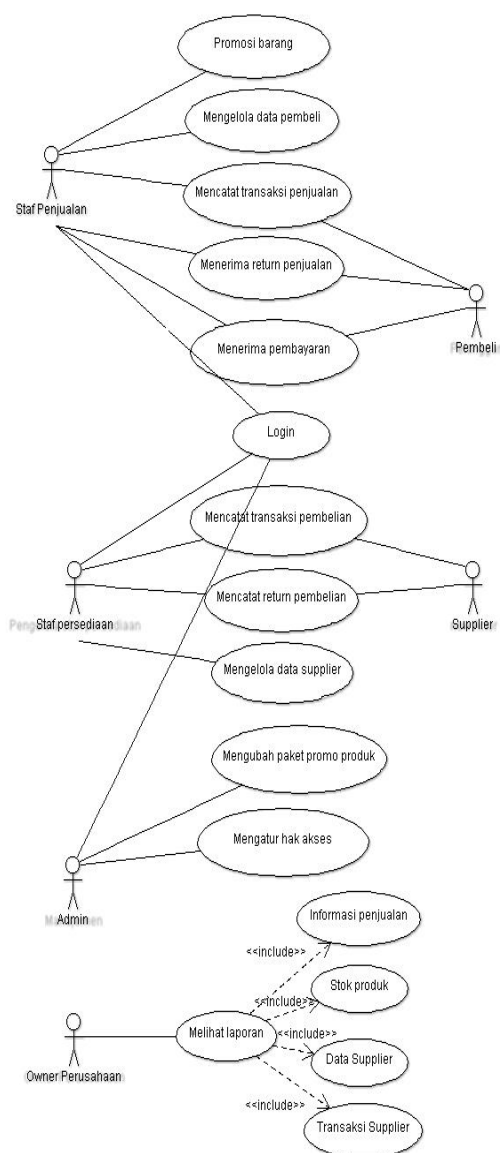
Data penelitian dalam riset ini diambil dari teknik interview dengan Ketua Koperasi Yayasan Istiqlal Indonesia (YII) untuk menentukan ukuran perangkat lunak berdasarkan kebutuhan fungsional sistem yang tergambar pada diagram use case.

Data yang didapat berupa use case metric berisi jumlah aktor dan fungsi yang terlibat dalam sistem.

Berikut diagram use case sistem penjualan barang pada Koperasi Yayasan Istiqlal Indonesia (YII):

3. Hasil dan Pembahasan

3.1. Usecase Diagram



Gambar. 3 Diagram Use Case Penjualan Barang

Diagram use case terdiri dari use case dan aktor yang menggambarkan sistem yang berjalan. Pada aplikasi sistem penjualan barang diketahui terdapat 6 aktor dan 16 use case yang digunakan. Berikut tahapan dalam menentukan nilai estimasi *effort* software aplikasi:

1. Klasifikasikan aktor yang terlibat untuk menghasilkan nilai *Unadjusted Actor Weight* (UAW).

Tabel 5. Mencari nilai *Unadjusted Actor Weight* (UAW)

Actor Type	Description	Weighting Factor	Number	Result
Simple	External System with well-defined API	1	--	--
Average	External system using a protocol based interface as TCP/IP	2	--	--
Complex	Human interacts through Graphic User Interface (GUI) or Web Page	3	6	18
Unadjusted Actor Weight Total (UAW)				18

2. Klasifikasikan *use case* yang ada di *use case* diagram untuk menghasilkan nilai *Unadjusted Use Case Weighting* (UUCW)

Tabel 6. Mencari nilai *Unadjusted Use Case Weighting* (UUCW)

Use Case Type	Description	Weighting Factor	Number	Result
Simple	1 – 3 transactions	5	6	30
Average	4 – 7 transactions	10	5	50
Complex	> 7 transactions	15	5	75
Unadjusted Use Case Weight Total (UUCW)				155

Mencari Nilai *Unadjusted Use Case Points* (UUCP) dengan menjumlahkan nilai UAW dan nilai UUCW, sehingga nilai UUCP dari aplikasi sistem penjualan barang adalah:

$$UUCP = UAW + UUCW = 173$$

mmm Menghitung nilai faktor teknik (TF) untuk memperoleh nilai Technical Complexity Factor (TCF) seperti terlihat pada tabel 7.

Tabel 7. Nilai Technical Factor (TF)

Factor Number	Description	Weight	Assigned Value (0-5)	UCP Weighted Value
T1	Distributed Systems	2,0	5	10,0
T2	Response time or throughput performance objectives	1,0	4	4,0
T3	End-user online efficiency	1,0	4	4,0
T4	Complex internal processing	1,0	4	4,0
T5	Reusability of code	1,0	2	2,0
T6	Easy to install	0,5	5	2,5
T7	Ease of use	0,5	4	2,0
T8	Portability	2,0	4	8,0
T9	Ease of change	1,0	3	3,0
T10	Concurrency	1,0	3	3,0
T11	Special security objectives included	1,0	2	2,0
T12	Direct access for third parties	1,0	5	5,0
T13	Special user training required	1,0	3	3,0
Nilai Tehnical Factor (TF)				52,5

$$TCF=0,6+(0,01*TF)$$

$$TCF=1,12$$

5. Menghitung nilai faktor lingkungan (EF) untuk memperoleh nilai *Environmental Complexity Factor* (ECF) seperti terlihat pada tabel 8.

Tabel 8. Nilai *Environmental Factor* (EF)

Factor Number	Description	Weight	Assigned Value (0-5)	UCP Weighted Value
E1	Familiarity with system development process being used	1,5	4	6
E2	Application experience	0,5	4	2
E3	Object-oriented experience	1,0	5	5
E4	Lead analyst capability	0,5	4	2
E5	Motivation	1,0	3	3
E6	Requirements stability	2,0	4	8
E7	Part time staff	-1,0	0	0
E8	Difficulty of programming language	-1,0	3	-3
Nilai <i>Environmental Factor</i> (EF)				23

$$ECF=1,4+(-0,03*EF)$$

$$ECF=0,72$$

6. Nilai UUCP, TCF dan ECF yang didapat digunakan untuk menghitung nilai Use Case Points dengan rumus:

$$UCP=UUCP*TCF*ECF$$

$$UCP=173*1,12*0,72$$

$$UCP=139,5$$

7. Berdasarkan penelitian (Subriadi dkk, 2014), diketahui nilai Effort Rate (ER) adalah 8,2 man-hours untuk aplikasi bisnis di Indonesia. Effort aplikasi didapat dengan menggunakan rumus:

$$Effort=UCP*ER =139,5 * 8,2$$

$$Effort= 1.144 \text{ man-hours}$$

Dari hasil penghitungan estimasi proyek software Sistem Penjualan Barang dengan menggunakan metode UCP diperoleh hasil estimasi effort sebesar 1.144 man-hour (orang per jam)

Sedangkan untuk perhitungan cost estimation tentang jam kerja wajib pekerja per minggu mengacu pada UU No. 13 Th. 2003 tentang ketenagakerjaan (UU Ketenagakerjaan) yaitu 40 jam kerja wajib per minggu atau sekitar 173 jam per bulan. Sedangkan gaji pekerja IT mengacu pada riset Kelly Service tahun 2017 pada perusahaan IT di Indonesia yang terdiri atas 50 item job description pada bidang IT. Dalam penulisan jurnal ini mengambil batas bawah dari range gaji dikarenakan riset penelitian ini akan mengacu pada perusahaan IT skala kecil menengah di Indonesia. Dalam pengembangan proyek sistem aplikasi ini yang dikerjakan oleh tim yang terdiri dari 3 orang, memiliki job desk masing-masing yaitu project leader, analis sistem, programmer dan technical support. Pengembangan Sistem Informasi Penjualan pada koperasi YII nilai estimasinya yaitu 1.144 dibagi 3 menjadi 381.3 jam atau 2.2 bulan.

Rata-rata pendapatan per jam (pay rate per hours) untuk pekerja IT di Indonesia berdasarkan data yang dikeluarkan oleh Kelly Service tahun 2017 seperti terlihat pada tabel di bawah ini.

Tabel 9. Pay Rate per Aktivitas

Activities	Pay rate per hours (rupiah)
<i>Needs analysis (Requirement)</i>	40.462
<i>Specification</i>	40.462
<i>Design</i>	28.902
<i>Implementation</i>	28.902
<i>Acceptance & installation</i>	28.902
<i>Project management</i>	86.705
<i>Configuration management</i>	86.705
<i>Documentation</i>	28.902
<i>Training & technical support</i>	28.902
<i>Integrated testing</i>	28.902
<i>Quality assurance</i>	46.243
<i>Evaluation & testing</i>	28.902

Berdasarkan penelitian yang dilakukan oleh Sholih dan Putu Linda tahun 2015 terhadap perusahaan skala kecil menengah di Indonesia, nilai distribusi effort untuk masing-masing fase sebagai berikut:

- *Software phase development* sebesar 72.6 %.
- *Ongoing life-cycle activity* sebesar 17.5 %.
- *Quality and testing phases* sebesar 9.9%.

Berikut tabel selengkapnya distribusi effort fase-fase dalam aktivitas pengembangan software aplikasi yang digunakan dalam estimasi cost pada software aplikasi.

Tabel 10. Estimasi Distribusi Effort dan Cost

Phases/ Activities	Prosentase Effort Distribution	Effort Distribution	Cost Estimation (rupiah)
Software Phase Development	72.6	830.5	
<i>Needs analysis (Requirement)</i>	1.6	18.3	740.455
<i>Specification</i>	7.5	85.8	3.471.640
<i>Design</i>	6.0	68.6	1.982.677
<i>Implementation</i>	52.0	594.9	17.193.800
<i>Acceptance & installation</i>	5.5	62.9	1.817.936
Ongoing life-cycle activity	17.5	200.2	
<i>Project management</i>	3.8	43.5	3.771.668
<i>Configuration management</i>	4.3	49.2	4.265.886
<i>Documentation</i>	8.4	96.1	2.777.482
<i>Training & technical support</i>	1.0	11.4	329.483
Quality and testing phases	9.9	113.3	
<i>Integrated testing</i>	7.0	80.1	2.315.050
<i>Quality assurance</i>	0.9	10.3	476.303
<i>Evaluation & testing</i>	2.0	22.9	661.856
Total of Effort	100	1144	39.804.236

Tabel 10 menunjukkan fase-fase dalam aktivitas pengembangan software yang memiliki persentase effort masing-masing yang berbeda. Total jumlah persentase pada masing-masing fase menunjukkan jumlah effort dan cost yang dikeluarkan untuk menyelesaikan sistem aplikasi yang akan digunakan pada Koperasi YII. Pada tabel terlihat cost yang dikeluarkan untuk menyelesaikan software aplikasi sebesar Rp 39.804.236.

Berdasarkan data aktual yang diperoleh dalam pengembangan Sistem Informasi Penjualan Barang pada Koperasi Istiqlal maka didapat nilai aktual cost sebesar Rp 47.675.417 dengan waktu yang dibutuhkan 407.5 jam.

Maka nilai deviasi cost adalah:

$$\begin{aligned} & (\text{actual cost} - \text{estimasi cost}) / \text{actual cost} * 100\% \\ & = (47,675,417 - 39,804,236) / 47,675,417 * 100\% \\ & = \mathbf{16,5\%} \end{aligned}$$

Dan nilai deviasi effort adalah:

$$\begin{aligned} & (\text{aktual effort} - \text{estimasi effort}) / \text{aktual effort} * 100\% \\ & = (407,5 - 381,3) / 407,5 * 100\% \\ & = \mathbf{6,4\%} \end{aligned}$$

4. Kesimpulan

Use Case Points (UCP) adalah metode yang mempunyai kemampuan untuk memberikan estimasi effort berdasarkan jumlah dan kompleksitas use case yang

dimiliki oleh software. Berdasarkan proyek sistem informasi penjualan yang dikembangkan oleh koperasi YII, maka sistem memiliki 6 aktor dan 16 use case. Nilai UCP yang diperoleh sebesar 139.5 dan nilai effort untuk software aplikasi domain bisnis sebesar 1144 jam.

Dari hasil interview yang dilakukan terhadap proyek sistem penjualan barang, didapat nilai aktual effort dan cost mendekati nilai estimasi yang dilakukan. Nilai estimasi effort 381,3 jam dan aktual effort 407.5 jam, sedangkan estimasi cost Rp 39.804.236 dan aktual cost Rp 47.675.417 sehingga didapat nilai deviasi effort 6,4% dan deviasi cost 16,5%.

Referensi

- Bente Anda. (2002). "Comparing effort estimates based on use cases with expert estimates." *Empirical Assessment in Software Engineering (EASE)*, Keele UK, p. 13.
- Carroll, Edward R. (2005). "Estimating Software Based on Use Case Points." *Object-Oriented, Programming, Systems, Languages, and Object Oriented Programming Systems Languages and Applications (OOPSLA) Conference*, San Diego, CA, pp.257–265.
- Clemmons. (2006). "Project Estimation with Use Case Points." *CrossTalk–The Journal of Defense Software Engineering*, vol. 19 , no.2, pp. 8-22.
- Dennis, A., Wixom, B. H., & Tegarden, D., (2005). "System Analysis and Design with UML Version 2.0 An Object-Oriented Approach (2nd ed)". USA: John Wiley & Sons, Inc.
- Edward R Carrol. (2005). "Estimating Software Based on Use Case Points". *Object Oriented Programming Systems Languages and Applications (OOPSLA) Conference*, San Diego, pp. 257–265.
- Grandys Frieska Prassida, Achmad Holil Noor Ali, dan Sholiq. (2012). Estimasi Biaya Pembuatan Modul Enterprise Resource Planning (ERP) Untuk Unit Bisnis Pabrik Gula Di PT. Perkebunan XYZ Dengan Metode Use Case Point. *Jurnal Teknik POMITS*, vol. 1, no. 1, pp. 1-6.
- Karner Gustav. (1993). "Resource Estimation for Objectory Projects". *Objective Systems SF AB*.
- Kassem Shaleh. (2011). "Effort and Cost Allocation in Medium to Large Software Development Projects". *International Journal of Computer*, vol. 5, no. 1, pp. 74-79.
- Nageswaran, Suresh. (2001). "Test Effort Estimasi Using Use Case Points". www.cognizant.com/cogcommunity/presentations/Test_Effort_Estimasi.pdf.
- Ochodek, M; Nawrocki, J; Kwarciak, K. (2011). "Simplifying Effort Estimasi Based on Use Case Points", *Sciencedirect*.
- Sholiq, Teguh Sutanto, Arifin Puji Widodo, and Wahyu Kurniawan, "Effort Rate on Use Case Point Method for Effort Estimation of Website Development," *Journal of Theoretical and Applied Information Technology*, vol. 63, no. 1, pp. 209-218.
- Standish Group. (2012), *Chaos Reports* <http://standishgroup.com/news>.
- Subriadi, Sholiq, dan Puji Agustin Ningrum. (2014). "Critical Review of The Effort Rate Value In Use Case Point Method for Estimating Software Development Effort," *Journal of Theoretical and Applied Information Technology*, vol. 59, no. 3, pp. 735-744.
- Tan, Thomas. (2012). "Domain-Based Effort Distribution Model For Software Cost Estimation". *A Dissertation Presented To The Faculty Of The USC Graduate School University Of Southern California*.
- Trendowicz, Adam. (2013). "Software Cost Estimation, Benchmarking, and Risk Assessment". Springer-Verlag Berlin Heidelberg. ISBN 978-3-642-30764-5 (eBook).