

Estimation Effort Pengembangan Software Inventory PT. Infinity Global Mandiri Menggunakan Metode *Use Case Point*

Rifky Permana ^{a,1}, Mahmud Syarif ^{b,2}, Fuad Nur Hasan ^{c,3}, Ari Abdillah ^{d,4},

^a Program Studi Sistem Informasi Universitas Bina Sarana Informatika, Jln. Kramat Raya No.98, Jakarta dan 10450, Indonesia

^b Program Studi Manajemen Universitas Bina Sarana Informatika, Jln. Kramat Rata No. 98, Jakarta dan 10450, Indonesia

^c Program Studi Ilmu Komputer Universitas Bina Sarana Informatika, Jln. Kramat Rata No. 98, Jakarta dan 10450, Indonesia

^d Program Studi Teknologi Informasi Universitas Bina Sarana Informatika, Jln. Kramat Rata No. 98, Jakarta dan 10450, Indonesia

¹ rifky.rpp@bsi.ac.id*; ² mahmud.may@bsi.ac.id; ³ fuad.fnu @bsi.ac.id; ⁴ ari.aab@bsi.ac.id;

* Penulis Korespondensi

Diterima : 21 Juli 2023 | Direvisi : 26 Juli 2023 | Diterbitkan : 04 Agustus 2023

ABSTRAK

Estimasi biaya merupakan salah satu faktor penentu keberhasilan proyek pembuatan perangkat lunak. Perkiraan biaya yang terlalu kecil akan mengakibatkan perangkat lunak yang dihasilkan berkualitas rendah, sebaliknya, perkiraan biaya yang terlalu besar akan mengakibatkan pemborosan sumber daya yang digunakan. Perhitungan estimasi biaya didasarkan pada usaha yang dilakukan untuk pembuatan perangkat lunak dan standar biaya aktivitas tenaga IT. Proyek kerangka pengembangannya pada umumnya mencakup banyak modul yang dikembangkan pada unit-unit bisnis yang ada, namun studi kasus pada penelitian ini hanya akan dibuat kerangka pengembangan perangkat lunak *inventory* pada PT. Infinity Global Mandiri. Metode yang digunakan dalam penelitian ini yaitu metode *Use Case Point*. Untuk menghitung besarnya effort, proses perhitungan melibatkan analisa function point yang dalam perhitungannya melibatkan pendekatan fungsional perangkat lunak dengan mengidentifikasi dan mengkuantifikasi kegunaan pemrosesan informasi melalui hasil perhitungan faktor eksponen dan effort adjustment factor. Effort dalam satuan person-month yang didapatkan nantinya akan dikalikan nilai gaji dengan mengacu pada standar gaji/tarif biaya aktivitas pekerja TI, maka hasil perhitungan akhir yang diperoleh dapat memberikan estimasi biaya yang diperlukan dalam pembuatan kerangka pengembangan perangkat lunak *inventory* pada PT. Infinity Global Mandiri dengan hasil perhitungan yaitu nilai estimasi effort 256,93 jam dan aktual effort 310.5 jam, sedangkan estimasi cost Rp 14.077.727 dan aktual cost Rp 32.517.255 sehingga didapat nilai deviasi effort 17,2% dan deviasi cost 56,7%.



KATA KUNCI

Perangkat Lunak
Estimasi Biaya
Use Case Point

ABSTRACT

Cost estimation is one of the factors that determine the success of a software development project. A cost estimate that is too small will result in low-quality software, conversely, a cost estimate that is too large will result in a waste of resources used. Cost estimation calculations are based on the effort put into software development and standard IT labor activity costs. The development framework project generally includes many modules developed in existing business units, but the case study in this research will only be made an inventory software development framework at PT Infinity Global Mandiri. The method used in this research is the Use Case Point method. To calculate the amount of effort, the calculation process involves function point analysis which in its calculation involves functional approach to software by identifying and quantifying the usefulness of information processing through the calculation of the exponent factor and effort adjustment factor. Effort in person-month units obtained will later be multiplied by the salary value by referring to the standard salary/cost rate for IT worker activities, then the final calculation results obtained can provide an estimate of the costs required in making an inventory software development framework at PT Infinity Global Mandiri with the calculation results, namely the estimated effort value of 256.93 hours and the actual effort of 310.5 hours, while the estimated cost is Rp. 14,077,727 and the actual cost is Rp. 32,517,255 so that an effort deviation value of 17.2% and a cost deviation of 56.7% is obtained.



KEYWORD

Software
Cost Estimation
Use Case Point



This is an open-access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license

1. Pendahuluan

Pentingnya perencanaan proyek yang baik tentu sudah disadari oleh beberapa pelaku bisnis di segala bidang. Tidak terkecuali di ranah teknologi informasi dan komunikasi (TIK), khususnya pada proyek pengembangan perangkat lunak. Di Indonesia, estimasi usaha sejatinya merupakan perencanaan proyek guna mengalokasikan sumber daya yang mereka butuhkan, oleh karena itu, estimasi usaha juga bertujuan untuk efisiensi penggunaan sumber daya proyek.

Pada proyek *software* estimasi biaya dan usaha proyek, mempunyai kesulitan tersendiri karena karakteristik-karakteristik *software* yang berbeda dengan proyek fisik. Kesulitan-kesulitan yang sering dihadapi dalam estimasi proyek *software* sangat berkaitan dengan sifat alami *software*, khususnya kompleksitas dan keabstrakan. Selain itu pengembangan *software* merupakan kegiatan yang lebih banyak dilakukan secara intensif oleh manusia.

Upaya pengembangan perangkat lunak merupakan hal yang sangat penting untuk pengelolaan proyek. Akurasi dan upaya yang dapat diandalkan dan biaya yang akan ditentukan pada proyek perangkat lunak sangat penting dalam perencanaan dan penjadwalan, terutama pada tahap awal *software development life cycle*. Pengembangan proyek *software* yang baik memiliki dua kegiatan kritis yaitu menghitung estimasi usaha atau biaya *software* secara tepat dan melakukan manajemen proyek yang berdasarkan *Software Development Life Cycle (SDLC)* secara terukur [1]

Pengembangan proyek *software* yang baik memiliki dua kegiatan kritis yaitu menghitung estimasi usaha atau biaya *software* secara tepat dan melakukan manajemen proyek yang berdasarkan *Software Development Life Cycle (SDLC)* secara terukur. Terdapat banyak metode tradisional untuk memperkirakan upaya bersama dengan beberapa faktor yang ditetapkan, seperti COCOMO, *Use Case Point*, *Function Point* dan banyak lagi. *Use Case Point* dianggap sebagai model estimasi biaya yang paling sesuai dan andal. Namun, pada fase awal estimasi usaha proyek sulit dan lebih menantang bagi manajer proyek. Jika perkiraan dibuat pada tahap awal, maka akan membantu untuk melacak sisa proyek dalam hal waktu dan biaya. 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 [2]. Menurut pendapat lain, UCP adalah metode yang dapat menganalisa aktor, *use case*, dan berbagai faktor teknis dan faktor lingkungan hingga menjadi suatu persamaan. Kelebihan dari metode *use case point* yaitu UCP memiliki deviasi sebesar 6% [3], UCP memiliki deviasi sebesar 9% [4] dan UCP memiliki deviasi sebesar 19%, sementara estimasi para ahli memiliki rata-rata deviasi sebesar 20% [5]. Hal ini menunjukkan metode estimasi UCP memiliki tingkat akurasi yang baik digunakan dalam menentukan *effort* terhadap suatu proyek *software* yang akan dikerjakan.

Dalam usaha menerapkan model *Use Case Point* akan dilakukan juga proses kalibrasi terhadap beberapa konstanta model dengan menggunakan data lokal untuk menilai kerangka pengembangan perangkat lunak penjualan yang pernah dilakukan di PT. Infinity Global Mandiri. Proses kalibrasi tersebut diharapkan akan menghasilkan konstanta model yang sesuai dengan kondisi proyek kerangka pengembangan perangkat lunak di PT. Infinity Global Mandiri. Dengan menerapkan dan mengkalibrasi model *Use Case Point* untuk proyek kerangka pengembangan perangkat lunak inventory di PT. Infinity Global Mandiri, maka beberapa manfaat dapat diperoleh, yaitu :

1. Estimasi biaya kerangka pengembangan perangkat lunak inventory di PT. Infinity Global Mandiri diharapkan akan lebih akurat, lebih mendekati keadaan yang sebenarnya, dan lebih obyektif dalam arti tidak bias terhadap tingkat pengalaman manajer proyek kerangka pengembangan perangkat lunak atas proyek-proyek serupa di masa lalu.
2. Estimasi biaya kerangka pengembangan perangkat lunak yang satu dengan yang lainnya diharapkan dapat saling diperbandingkan secara lebih benar karena digunakannya model yang baku tersebut. Adanya model yang baku ini akan mendukung proses studi banding (*benchmarking*) antar perangkat lunak sehingga dapat mempercepat proses pengambilan keputusan menyangkut estimasi biaya kerangka pengembangannya.
3. Model *Use Case Point* yang sudah dikalibrasi sesuai dengan kondisi lingkungan proyek kerangka pengembangan perangkat lunak di PT. Infinity Global Mandiri diharapkan akan menjadi model acuan dalam melakukan estimasi biaya kerangka pengembangan perangkat lunak lainnya di masa mendatang. Model tersebut akan semakin akurat apabila terus dilakukan kalibrasi dengan menggunakan data proyek kerangka pengembangan perangkat lunak lainnya yang lebih baru.

Melakukan estimasi pengembangan perangkat lunak yang handal dan akurat merupakan tugas bagi manajer proyek. Terdapat beberapa metode awal yang digunakan dan dikembangkan untuk memperkirakan berapa tenaga dan waktu yang dibutuhkan untuk proyek pengembangan perangkat lunak. Oleh sebab itu dengan metode *Use Case Point* sebagai acuan untuk menentukan dan menghitung estimasi dalam pengembangan perangkat lunak inventori pada PT. Infinity Global Mandiri.

2. Tinjauan Pustaka

Pada penelitian ini, penulis mempunyai beberapa sumber referensi yang terkait dengan judul penelitian yang dibuat, diantaranya adalah :

Menurut [6] *Software Cost Estimation* adalah proses memprediksi *Effort* yang diperlukan untuk mengembangkan sebuah perangkat lunak. *Software Cost Estimation* meliputi 1) *Effort*; 2) *Project Duration*; 3) *Cost*.

Unified Modeling Language (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks – teks pendukung [7] diagram dalam *Unified Modeling Language* (UML) yang digunakan dalam pembuatan program ini, yaitu *Use Case Diagram*. “*Use Case Diagram* adalah rancangan awal untuk mendesain sistem guna menjadi bahan diskusi kepada pengguna untuk didapatkan evaluasi apa saja yang perlu ditambahkan dan kurangi” [8] Sedangkan pendapat lain tentang *Use Case Diagram* yaitu Menurut Satzinger, Jackson, & Burd dalam [9] “*Use Case Diagram* adalah diagram yang menampilkan berbagai peran pengguna dan bagaimana peran ini berfungsi dalam sebuah sistem”. Maka dapat disimpulkan pengertian *Use Case Diagram* adalah sebuah rancangan awal yang menampilkan berbagai peran atau interaksi sistem dengan aktor dan juga dapat mengetahui bagaimana peran ini berfungsi dalam sebuah sistem tersebut.

Metode UCP adalah perluasan metode *Function Point* dengan manfaat analisis dalam kebutuhan proses berorientasi objek. Tahap awal yaitu dengan mengukur fungsionalitas sistem berdasarkan model *use case* dalam hitungan yang disebut *Unadjusted Use Case Point* (UUCP). Faktor teknis yang sama digunakan sebagai *Function Point*. Faktor baru yang disebut Faktor Lingkungan diusulkan oleh penulis. Penulis mendefinisikan *Use Case Points* (UCP) sebagai produk dari ketiga faktor ini (yaitu UUCP, Faktor Teknis dan Faktor Lingkungan). UCP menunjukkan estimasi ukuran sistem yang dapat dipetakan lebih lanjut ke jam kerja untuk menghitung upaya yang diperlukan dalam mengembangkan sistem.

Penelitian sebelumnya pernah dilakukan oleh [10] : yaitu menjelaskan studi ini telah menunjukkan bahwa metode Cocomo II dan *Use Case Point* adalah pendekatan yang lebih kalkulatif yang meliputi banyak faktor yang dapat mempengaruhi biaya. Tetapi menurut para ahli merupakan pendekatan yang lebih prediktif, dimana pengalaman banyak diperlukan dalam melakukan perhitungan serta data lama juga diperlukan untuk menghasilkan estimasi yang lebih baik lagi. Setelah perhitungan menggunakan kedua metode, ditemukan perbedaan lebih dari 5% dari rata rata estimasi sehingga perlu dilakukan perhitungan ulang agar hasilnya lebih akurat. Untuk memperkiraan KLOC proyek dibagi menjadi kedalam modul dan sub modul sampai bisa diperkirakan KLOC nya. Menggunakan *use case point* dapat menunjukkan persyaratan fungsionalitas sistem.

Penelitian sebelumnya juga pernah dilakukan oleh [11] : yaitu menjelaskan pengembangan proyek *software* yang baik memiliki dua kegiatan kritis yaitu menghitung estimasi usaha atau biaya *software* secara tepat dan melakukan manajemen proyek yang berdasarkan SDLC (*Software Development Life Cycle*) secara terukur. 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. *Use Case Points* (UCP) adalah metode yang mempunyai kemampuan untuk memberikan estimasi *effort* berdasarkan jumlah dan kompleksitas *use case* yang dimiliki oleh *software*. 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%.

Penelitian sebelumnya juga pernah dilakukan oleh [12] : yaitu menjelaskan tingkat kegagalan pada proyek pengembangan perangkat lunak sangat tinggi. Pada tahun 2002 sampai 2010 hanya terdapat 37% proyek teknologi informasi yang sukses. Salah satu prosentase terbesar yang menyebabkan kegagalan proyek pengembangan perangkat lunak yaitu kurang baiknya perencanaan proyek, yakni mempunyai prosentase sebesar 39%. Penelitian penentuan nilai *effort rate* merupakan proyek penelitian yang dilakukan dengan menggunakan beragam proyek perangkat lunak. Penelitian dibagi meliputi perhitungan nilai ER pada proyek pengembangan perangkat lunak pemerintahan, pendidikan, bisnis dan proyek pengembangan *website* pemerintahan. Hasil akhir yang didapat dari penelitian ini adalah korelasi antara nilai *actual effort* dengan nilai UCP pada penelitian tugas akhir ini yaitu sebesar 0.850, maka dapat disimpulkan bahwa nilai *actual effort* dengan nilai UCP mempunyai korelasi yang sangat kuat. Dengan diketahuinya nilai *actual effort* dan nilai UCP masing-masing proyek pengembangan perangkat lunak *website* pemerintahan, maka didapatkan persamaan linier yaitu: $y = 1094,360 + 5,178x$, sehingga didapatkan nilai empiris *Effort Rate* (ER) yaitu sebesar 5,178 *man/hours*. Nilai ER tersebut lebih kecil dibandingkan oleh nilai ER yang dikemukakan oleh Karner sebesar 20 *man/hours*.

Penelitian sebelumnya juga pernah dilakukan oleh [13] menjelaskan metode *Use Case Point* adalah metode pengerjaan dan estimasi pada awal perangkat lunak berdasarkan pada jumlah *use case* yang disebut *use case point*. Untuk mendapatkan informasi yang berguna diawal proses pengembangan perangkat lunak proses, fokus utamanya terdapat pada dua diagram UML yaitu *Use Case Diagram* dan *Class Diagram*, sedangkan elisitasi kebutuhan diperoleh melalui *use case* diagram, sedangkan detail desain informasi tersedia dalam *class diagram*. Terdapat banyak pendekatan tersedia yang memanfaatkan pengetahuan dari penggunaan *Use case diagram* dan *class diagram* dalam proses memprediksi usaha untuk suatu sistem. *Use case diagram* bukan satu-satunya sumber estimasi awal. Analisis *class diagram* juga bisa menyediakan segenggam informasi yang berkaitan dengan upaya yang diperlukan untuk membangun perangkat lunak. Pendekatan lain dari poin UML menyediakan perhitungan sederhana, mudah diterapkan, dan disediakan perkiraan biaya yang masuk akal pada tahap pengembangan perangkat lunak karena memanfaatkan *use case* dan *class diagram*. Penelitian ini telah dianjurkan untuk menggunakan metode *use case point* sebagai metode efektif untuk estimasi usaha perangkat lunak dalam tahap awal proses pengembangan perangkat lunak. Penelitian makalah dari penulis yang berbeda setelah ditinjau ditekankan untuk memperbaiki metode estimasi *use case point* untuk proyek pengembangan perangkat lunak. Namun, agar efektif metode estimasi UCP perlu ditinjau dan diperbaiki dengan cara menghitung transaksi *use case*. Oleh karena itu, kebutuhan jam adalah untuk lebih lanjut memperbaiki parameter yang digunakan dalam lingkungan dan faktor teknis metode UCP dengan metode yang ditingkatkan dan mudah digunakan untuk menghitung jumlah transaksi *use case* yang dapat menghasilkan model estimasi UCP dengan upaya yang lebih efektif untuk proyek pengembangan perangkat lunak diagram.

3. Metodologi Penelitian

Dalam penelitian ini penulis menggunakan metode penelitian kualitatif yaitu dengan cara pengumpulan data berupa wawancara yang mendalam dan observasi serta penyebaran kuisioner kepada tim IT, dari hasil riset dan observasi yang telah dilakukan penulis terkait sistem inventori di PT. Infinity Global Mandiri yaitu pengolahan data inventori yang masih menggunakan sistem manual sehingga menghambat kinerja dan membuat penyajian laporan barang masuk serta barang keluar menjadi terhambat dan sering kali tidak sesuai jumlah stok tertulis dengan yang ada di gudang. Selain itu juga digunakan metode *Use Case Point* sehingga hasil jenis penelitian ini tidak dijadikan sebagai pengambilan keputusan, akan tetapi sebatas memberikan gambaran atau informasi yang signifikan atas permasalahan yang ada. Penulis juga mengumpulkan data-data yang dibutuhkan dengan mencari berbagai referensi atau sumber-sumber lainnya seperti buku, artikel ilmiah, maupun paper.

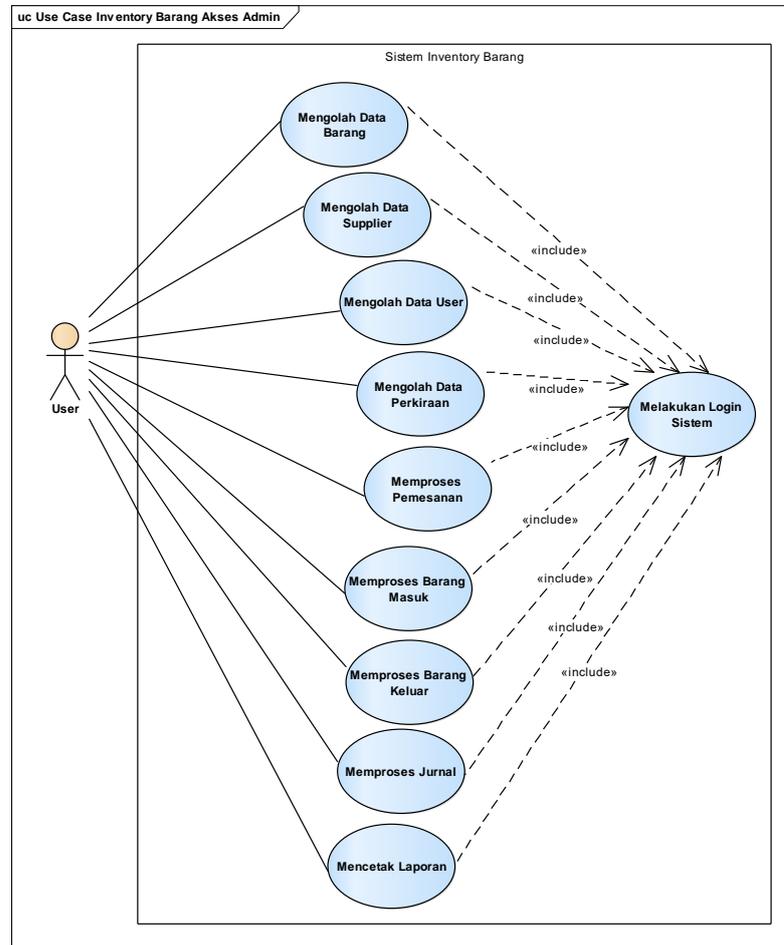
4. Hasil dan Pembahasan

4.1. Hasil

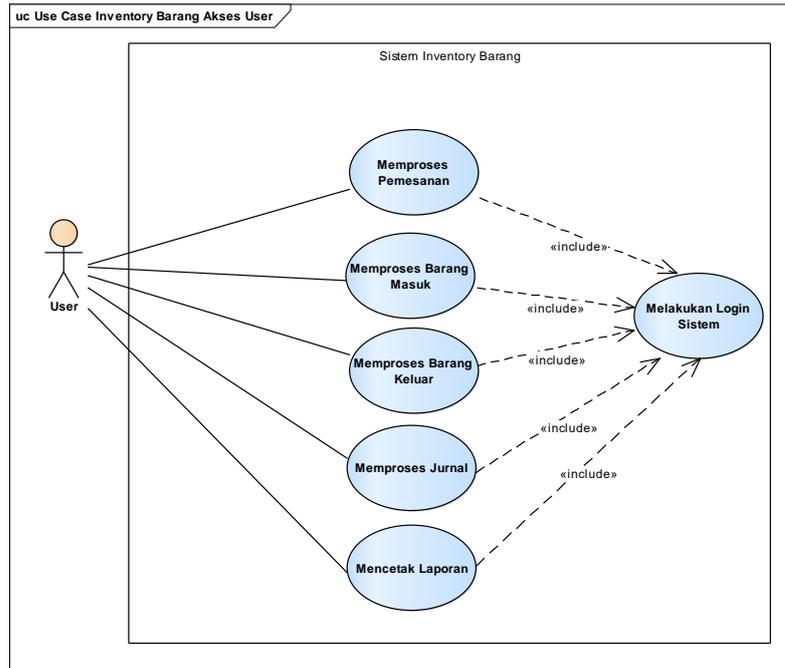
Hasil penelitian ini adalah estimasi biaya pembuatan *software inventory* berbasis *web* yang nantinya akan digunakan untuk pencatatan stok barang masuk dan barang keluar agar lebih memudahkan para petugas gudang dalam melakukan pencatatan dan pembuatan laporan barang masuk dan barang keluar serta pendataan stok barang.

4.2. Pembahasan

Use case diagram yang terdiri dari *use case* dan aktor menggambarkan apa yang dapat dikerjakan oleh sistem dan saat ini lebih banyak digunakan untuk melakukan estimasi proyek perangkat lunak. Sebagai contoh pengujian, diambil data proyek perangkat lunak inventori yang digunakan pada PT. Infinity Global Mandiri.



Gambar 1. *Use Case* Diagram Sistem Inventori Barang Akses Admin



Gambar 2. Use Case Diagram Sistem Inventori Barang Akses User

Berdasarkan diagram *use case* yang ada, dapat diketahui terdapat 2 (dua) aktor dan 10 (sepuluh) *use case*. Menghitung *effort* dengan metode *use case points* menggunakan tahapan-tahapan :

1. Klasifikasikan aktor yang terlibat

Terdapat dua aktor yaitu admin dan user, hasil perkalian jumlah aktor sesuai klasifikasi aktor disebut *Unadjusted Actor Weight (UAW)*

Tabel 1. *Unadjusted Actor Weighting*

<i>Actor Type</i>	<i>Description</i>	<i>Weighting Factor</i>	<i>Number</i>	<i>Result</i>
<i>Simple</i>	<i>External System with well-defined API</i>	1	--	--
<i>Average</i>	<i>External system using a protocol based interface</i>	2	--	--
<i>Complex</i>	<i>Human</i>	3	2	6
<i>Unadjusted Actor Weight Total (UAW)</i>				6

2. Klasifikasi *use case* diagram

Tabel 2. *Unadjusted Use Case Weighting Table*

<i>Use Case Type</i>	<i>Description</i>	<i>Weighting Factor</i>	<i>Number</i>	<i>Result</i>
<i>Simple</i>	<i>1 – 3 transactions</i>	5	2	10
<i>Average</i>	<i>4 – 7 transactions</i>	10	4	40
<i>Complex</i>	<i>> 7 transactions</i>	15	4	60
<i>Unadjusted Use Case Weight Total (UUCW)</i>				110

3. Nilai UAW dan UUCW dijumlahkan menjadi UUCP, sehingga UUCP tersebut adalah 116

4. Menghitung nilai faktor teknik seperti pada tabel 3.

Tabel 3. *Technical Complexity Factors Table*

<i>Metric</i>	<i>Description</i>	<i>Weight</i>	<i>Value</i>	<i>TCF</i>
TCF01	<i>Distributed System</i>	2,0	5,0	10,0
TCF02	<i>Response or throughput performance objectives</i>	1,0	4,0	4,0
TCF03	<i>End user efficiency (online)</i>	1,0	2,0	2,0
TCF04	<i>Complex internal processing</i>	1,0	4,0	4,0
TCF05	<i>Code must be re-usable</i>	1,0	2,0	2,0
TCF06	<i>Easy to install</i>	0,5	5,0	2,5
TCF07	<i>Easy to use</i>	0,5	3,0	1,5
TCF08	<i>Portable</i>	2,0	3,0	6,0
TCF09	<i>Easy to change</i>	1,0	3,0	3,0
TCF10	<i>Concurrent</i>	1,0	2,0	2,0
TCF11	<i>Includ special security features</i>	1,0	2,0	2,0
TCF12	<i>Provide direct access for third parties</i>	1,0	5,0	5,0
TCF13	<i>Special user training facilities are required</i>	1,0	3,0	3,0
<i>Technical Factor Value (TFactor)</i>			Total:	47,0

Rumus *Technical Complexity Factor*

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

$$TCF = 0.6 + (0.01 * 47)$$

$$TCF = 1.07$$

Tabel 4. *Technical Complexity Factors Table*

<i>Factor</i>	<i>Value</i>
<i>Unadjusted TCF value (UTV)</i>	47,0
<i>TCF Weighting (TWF)</i>	0,01
<i>TCF Constant (TC)</i>	0,60
<i>Technical Complexity Factor (TCF) = TC + (UTV * TWF)</i>	1,07

5. Menghitung nilai faktor lingkungan seperti pada tabel 5

Tabel 5. *Environmental Factors Table*

<i>Metric</i>	<i>Description</i>	<i>Weight</i>	<i>Value</i>	<i>TCF</i>
ECF01	<i>Familiar with Rational Unified Process</i>	1,5	4,0	6,0
ECF02	<i>Application experience</i>	0,5	3,0	1,5
ECF03	<i>Object-oriented experience</i>	1,0	4,0	4,0
ECF04	<i>Lead analyst capability</i>	0,5	4,0	2,0
ECF05	<i>Motivation</i>	1,0	3,0	3,0
ECF06	<i>Stable requirements</i>	2,0	4,0	8,0
ECF07	<i>Part-time workers</i>	-1,0	0,0	-0,0
ECF08	<i>Difficult programming language</i>	-1,0	3,0	-3,0
			Total:	21,5

Rumus *Environmental Factor*

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

$$EF = 1,4 + (-0,03 * 21,5)$$

$$EF = 0,75$$

Tabel 6. *Environmental Factors Table*

<i>Factor</i>	<i>Value</i>
<i>Unadjusted ECF value (UEV)</i>	21,5
<i>ECF Weighting (EWF)</i>	-0,03
<i>ECF Constant (EC)</i>	1,4
<i>Environmental Complexity Factor (ECF) = EC + (UEV * EWF)</i>	0,75

6. Nilai *UUCP*, *TCF* dan *EF* digunakan untuk menghitung nilai *Unadjusted Case Points* dengan rumus:

$$UCP = UUCP * TCF * EF$$

$$UCP = 116 * 1,07 * 0,755$$

$$UCP = 94$$

7. Berdasarkan penelitian [14], diketahui nilai *Effort Rate* (ER) adalah 8,2 *man/hours* untuk aplikasi bisnis di Indonesia. *Effort* aplikasi didapat dengan menggunakan rumus:

$$Effort = UCP * ER = 94 * 8,2$$

$$Effort = 770,8 \text{ man/hours}$$

Berdasarkan hasil penghitungan estimasi proyek *software* sistem penjualan dengan menggunakan metode *UCP* diperoleh hasil estimasi *effort* sebesar 770,8 *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 2021 pada perusahaan IT di Indonesia yang terdiri atas 50 item job description pada bidang IT. Pada penelitian ini mengambil batas bawah dari range gaji dikarenakan riset penelitian ini akan mengacu pada perusahaan IT skala kecil menengah di Indonesia. Pada 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 perangkat lunak inventori pada PT. Infinity Global Mandiri nilai estimasinya yaitu 770,8 dibagi 3 menjadi 256,93 jam atau 33 Hari kerja.

Berdasarkan referensi gaji untuk pekerja IT di Indonesia berdasarkan data tahun 2023 terlihat pada tabel 7 berikut ini :

Tabel 7. *Pay Rate* per Aktivitas

Activities	Pay rate/hours (rupiah)
<i>Needs analysis (Requirement)</i>	60.462
<i>Specification</i>	60.462
<i>Design</i>	48.902
<i>Implementation</i>	48.902
<i>Acceptance & installation</i>	48.902
<i>Project management</i>	106.705
<i>Configuration management</i>	106.705
<i>Documentation</i>	48.902
<i>Training & technical support</i>	48.902
<i>Integrated testing</i>	48.902
<i>Quality assurance</i>	66.243
<i>Evaluation & testing</i>	48.902

Berdasarkan penelitian sebelumnya, 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 phase sebesar 9.9%.

Berikut tabel selengkapnya distribusi *effort* fase-fase dalam aktivitas pengembangan perangkat lunak penjualan yang digunakan dalam estimasi *cost* pada perangkat lunak penjualan.

Tabel 8. Estimasi Distribusi *Effort* dan *Cost*

<i>Phases/Activities</i>	<i>Prosentase Effort Distribution</i>	<i>Effort Distribution</i>	<i>Cost Estimation (rupiah)</i>
Software Phase Development	72,6	186.5312	
Needs analysis (Requirement)	1,6	4.11088	Rp 248,552
Specification	7,5	19.26975	Rp1,165,088
Design	6	15.4158	Rp 753,863
Implementation	52	133.6036	Rp6,533,483
Acceptance & installation	5,5	14.13115	Rp 691,041
Ongoing life-cycle activity	17,5	44.96275	
Project management	3,8	9.76334	Rp1,041,797
Configuration management	4,3	11.04799	Rp1,178,876
Documentation	8,4	21.58212	Rp1,055,409
Training & technical support	1	2.5693	Rp 125,644
Quality and testing phases	9,9	25.43607	
Integrated testing	7	17.9851	Rp 879,507
Quality assurance	0,9	2.31237	Rp 153,178
Evaluation & testing	2	5.1386	Rp 251,288
Total of Effort	100	256.93	Rp14,077,727

Tabel 4.11 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 PT. Infinity Global Mandiri. Pada tabel tersebut terlihat *cost* yang dikeluarkan untuk menyelesaikan *software* aplikasi sebesar Rp 14.077.727. Berdasarkan data aktual yang diperoleh dalam pengembangan perangkat lunak penjualan pada PT. Infinity Global Mandiri maka didapat nilai aktual *cost* sebesar Rp 32.517.255 dengan waktu yang dibutuhkan 310,5 jam. Maka nilai deviasi *cost* adalah:

$$\begin{aligned} & (\text{actual cost} - \text{estimasi cost}) / \text{actual cost} * 100\% \\ & = (32.517.255 - 14.077.727) / 32.517.255 * 100\% \\ & = 56,7\% \end{aligned}$$

Dan nilai deviasi *effort* adalah:

$$\begin{aligned} & (\text{aktual effort} - \text{estimasi effort}) / \text{aktual effort} * 100\% \\ & = (310,5 - 256,93) / 310,5 * 100\% \\ & = 17,2\% \end{aligned}$$

5. Penutup

5.1. 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 persediaan yang dikembangkan oleh PT. Infinity Global Mandiri, maka sistem memiliki 2 aktor dan 10 use case. Nilai UCP yang diperoleh sebesar 94 dan nilai *effort* untuk *software* aplikasi domain bisnis sebesar 770,8 jam. Berdasarkan hasil interview yang dilakukan terhadap proyek penjualan, didapat nilai aktual *effort* dan *cost* mendekati nilai estimasi yang dilakukan. Nilai estimasi *effort* 256,93 jam dan aktual *effort* 310.5 jam, sedangkan estimasi *cost* Rp 14.077.727 dan aktual *cost* Rp 32.517.255 sehingga didapat nilai deviasi *effort* 17,2% dan deviasi *cost* 56,7%.

5.2. Saran

Untuk penelitian selanjutnya, dalam melakukan estimasi biaya perangkat lunak, diharapkan untuk menggunakan metode *use case point* dalam mengestimasi biaya kerangka pengembangan perangkat lunak penjualan di PT. Infinity Global Mandiri. Penelitian selanjutnya dapat mengembangkan penelitian ini dengan menambahkan metode estimasi biaya yang lainnya sebagai pembandingan dengan hasil penelitian ini. Serta dengan penambahan *point* mengenai tingkat kerumitan modul atau transaksi per *use case*, karena tingkat kerumitan yang ada saat ini bersifat keseluruhan

Daftar Pustaka

- [1] I. K. Walia and M. Kaur, "Combining COCOMO and use case for better effort estimation," *Int. J. Curr. Eng. Technol.*, vol. 4, no. 2, pp. 576–578, 2014.
- [2] G. Karner, *Resource Estimation for Objectory Projects*. Objective Systems SF AB, 1993.
- [3] S. Nageswaran, *Test Effort Estimasi Using Use Case Points*. 2001.
- [4] E. R. Carroll, "Estimating Software Based on Use Case Points," *Object-Oriented, Program. Syst. Lang. Object Oriented Program. Syst. Lang. Appl. Conf.*, pp. 257–265, 2005.
- [5] B. Anda, *Comparing effort estimates based on use cases with expert estimates*. Keele UK: *Empirical Assessment in Software Engineering (EASE)*, 2022.
- [6] M. R. Z. Alnobeta, M. C. Saputra, and A. D. Herlambang, "Estimasi Biaya Perangkat Lunak Menggunakan Metode Function Point (Studi Kasus : CV Aptikma Indonesia)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 1, pp. 40–46, 2018, [Online]. Available: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&cad=rja&uact=8&ved=0ahUKEwjx9Yil2IrQAhWMto8KHRDGCZkQFghFMAQ&url=https%3A%2F%2Fop.enlibrary.telkomuniversity.ac.id%2Fpustaka%2Ffiles%2F95865%2Fresume%2Fanalisis-dan-implementasi-part-of-speech->.
- [7] R. A. Sukamto and M. Shalahuddin, *Rekayasa Perangkat Lunak Struktur dan Berorientasi Objek*. Bandung: Informatika, 2016.
- [8] S. Luckyardi, H. Saputra, N. Safitri, A. Cahyaningrum, D. Septiani, and R. Hidayat, "Perancangan Sistem Informasi Penjualan Busana Muslim Berbasis Web Design of Web-Based Muslim Clothing Sales Information System," *IJIS Indones. J. Inf. Syst.*, vol. 6, no. September 2021, pp. 156–165, 2021.
- [9] I. W. Pratama and I. Nurlela, "Sistem Informasi Akuntansi Aset Tetap Kendaraan Pada Bumi Waras di Bandar Lampung," *J. Sist. Inf. Akunt. AMIK Dian Cipta Cendikia*, vol. 1, no. 1, pp. 56–66, 2018, [Online]. Available: <https://jurnal.dcc.ac.id/index.php/jusinta/article/view/133>.
- [10] C. Nagar and A. Dixit, "Efforts Estimation by combining the Use Case Point and COCOMO," *Int. J. Comput. Appl.*, vol. 52, no. 7, pp. 1–5, 2012, doi: 10.5120/8211-1624.

- [11] E. Prayitno, "Penggunaan Metode Estimasi Use Case Points (UCP) Dalam Proyek Software Domain Bisnis," J. Inform., vol. 4, no. 2, pp. 241–248, 2017.
- [12] W. Kurniawan, Sholiq, and T. Sutanto, "Jurnal Sistem Informasi PENENTUAN EFFORT RATE PADA ESTIMASI EFFORT," vol. 2, no. 2, pp. 62–71, 2013.
- [13] M. M. Kirmani and A. W. Dean, "Use Case Point Method of Software Effort Estimation: A Review," Int. J. Comput. Appl., vol. 116, no. 15, pp. 975–8887, 2015.
- [14] S. Subriadi and P. A. Ningrum, "Critical Review of The Effort Rate Value In Use Case Point Method for Estimating Software Development Effort," J. Theor. Appl. Inf. Technol., vol. 59, no. 3, pp. 735–744, 2014.