IMPLEMENTASI METODE INCREMENTAL DALAM MEMBANGUN APLIKASI USE CASE POINT PADA PERUSAHAAN DTS

Mukhamad Faiz Fanani [1]. Sholiq[2] dan Feby Artwodini Muqtadiroh[3]
Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Jl. Arief Rahman Hakim, Surabaya 60111
E-mail: fananifaiz@yahoo.com [1], sholiq@is.its.ac.id [2], feby.herbowo@gmail.com [3]

Abstrak

Dynamic Team Solution (DTS) merupakan salah satu perusahaan yang bergerak dalam pembuatan aplikasi berbasis project. Dalam menentukan harga proyek pembuatan aplikasi, tim pengembang melakukan perkiraan nilai harga dari pembangunan aplikasi. Selama proses estimasi, tim pengembang sering menemui beberapa permasalahan.Oleh karena itu, aplikasi Use Case Point sangat dibutuhkan untuk melakukan estimasi harga perangkat lunak berdasarkan metode Use Case Point(UCP). Metode pengembangan perangkat lunak untuk membuat aplikasi Use Case Point adalah metode incremental model. Hasil dari penelitihan ini berupa Aplikasi Use Case Point melalui 3 kali increment. Increment pertama ditambahkan fitur estimasi usaha. Increment kedua ditambahkan fitur biaya, increment ketiga ditambahkan fitur-fitur untuk melakukan kalibrasi perhitungan estimasi. Pengujian terhadap aplikasi dilakukan disetiap increment yang ad dengan menggunakan metode pengujian blackbox testing/correctness testing, useablity testing, dan portability testing, dan security testing. Keluaran dari penelitian ini yaitu berupa Aplikasi Use Case Point, dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL),Deskripsi Perancangan Perangkat Lunak (DPPL), dokumen pengujian, dan dokumen panduan.

Kata kunci: *Estimasi harga perangkat, Effort, UCP, Incremental Model, SKPL, DPPL*

Abstract

Team Dynamic Solution (DTS) is a company engaged in the manufacture of application-based project. In determining the price of the project of making the application, the development team do an estimated value of the price of applications development. During the estimation process, the developer team often encountered several problems. Therefore, the application Use Case Point is needed to estimate the price of a software based method Use Case Point (UCP). Software development methods to create applications Use Case Point is a method of incremental models. Results from this research is form of Application Use Case Point through 3 times of increment. The first increment added features Effort estimation. The second increment costs added feature, the third increment added features to perform calibration estimation calculations. Testing of applications carried in each increment that add testing using blackbox testing / correctness testing, useablity testing and portability testing, and security testing. The output of this research is in the form of Application Use Case Point, Software Requirements Specification document (SRS), Software Design Description (SDD), document testing, and guidance documents.

Keywords: *Software Cost Estimation, Effort, UCP, Incremental Model, SRS, SDD*

1. PENDAHULUAN

Perusahaan DTS merupakan perusahaan yang bergerak dalam bidang pengembangan perangkat lunak berbasis proyek. Sebelum proyek pengembangan perangkat lunak dikerjakan, tim pengembang di DTS melakukan estimasi nilai harga dari proyek perangkat lunak terlebih dahulu untuk kemudian dikirimkan dokumen penawaran ke calon *client*-nya. Selama proses menentukan harga estimasi tersebut, Perusahaan DTS sering menemui beberapa masalah. Oleh karena itu, peneliti membuat aplikasi Use Case Point untuk melakukan estimasi harga perangkat lunak berdasarkan metode Use Case Point dan alur bisnis dari Perusahaan DTS . UCP adalah metode untuk mengukur besarnya *Effort* yang harus dilakukan dalam pembuatan perangkat lunak berdasarkan use case yang ada [1] Penggunaan metode digunakan karena tingkat akurasi metode UCP dalam menentukan estimasi usaha (*Effort*) ini lebih tinggi daripada metode estimasi harga lainnya [2]. Tahapan pengembangan perangkat lunak menggunakan model incremental. Model ini bisa digunakan pada saat sumber daya tim pengembang terbatas. Hal ini dikarenakan model incremental dibagi menjadi beberapa bagian yang lebih kecil (*software release)*. Selain itu, terdapat *feedback* pada setiap software release membuat kebutuhan

pengguna semakin jelas. Model Incremental juga dapat meminimalisir resiko cacat/*bug* selama proses pengembangan perangkat lunak karena setiap *realease* dilakukan pengujian secara bertahap.

2. PUSTAKA

Pada bagian ini akan dijelaskan tentang *Use Case Point* (UCP), Hasil Penelithan terkait UCP, Prosedur perhitungan UCP, *Unadjusted Use Case Points* (UUCP), *Technical Complexity Factor* (TCF), *Environment Complexity Factor* (ECF).

2.1 Use Case Point (UCP)

Use Case point adalah teknik atau metode untuk melakukan estimasi ukuran suatu proyek pengembangan perangkat lunak. Salah satu keunggulan dari metode UCP ini yaitu hasil estimasi yang dihasilkan mendekati benar [1].

2.2 Prosedur Perhitungan UCP

Metode UCP menganalisa use case, actor dari use case, secenario, faktor lingkungan dan faktor technical untuk kemudian dibuatkan dalam bentuk sebuah persamaan. UCP disusun atas 3 variabel:

- *Unadjusted Use Case Points* (UUCP).
- *Technical Complexity Factor* (TCF).
- *Environment Complexity Factor* (ECF).

Hasil dari masing-masing komponen tersebut dilakukan pengalian untuk mendapatkan nilai UCP. Adapun rumus perhitungan nilai UCP sendiri yaitu sebagai berikut.

$$UCP=UUCP*TCF*ECF \quad (1)$$

2.3 Unadjusted Use Case Points (UUCP)

UUCP didapatkan melalui dua perhitungan yaitu sebagai berikut.

- *Unadjusted Use Case Weight* (UUCW)

Terdapat 3 kategori dalam menentukan nilai UUCW. Tabel 1 berikut ini menunjukan pembobotan pada setiap use case yang ada pada aplikasi.

Tabel 1 Klasifikasi Use Case

Tipe	Jumlah Transaksi	Bobot
Simple	1 sampai 3 transaksi	5
Average	4 sampai 7 transaksi	10
Complex	8 transaksi atau lebih	15

Nilai UUCW didapatkan dengan rumus berikut ini.

$$UUCW = (Simple\ Use\ Cases \times 5) + (Average\ Use\ Case \times 10) + (Complex\ Use\ Cases \times 15) \quad (2)$$

- Unadjusted Actor Weight (UAW)

UAW didapatkan berdasarkan kompleksitas dari semua actor yang ada di semua use case. Adapun klasifikasi bobot aktor seperti pada tabel berikut.

Tabel 2 Klasifikasi aktor

Klasifikasi Aktor	Tipe Aktor	Bobot
Simple	Berinteraksi melalui baris perintah atau *Command Prompt*	1
Average	Berinteraksi dengan protokiol komunikasi seperti (e.g. TCP/IP, FTP, HTTP, database)	2
Complex	Berinteraksi dengan GUI atau web page	3

Hasil perhitungan UAW didapatkan dengan menggunakan rumus sebagai berikut.

$$UAW = (Simple\ actors \times 1) + (Average\ actors \times 2) + (Complex\ actors \times 3) \quad (3)$$

Dari dua perhitungan tadi, didapatkan nilai UUCP didapatkan dari rumus berikut.

$$UUCP=UAW+UUCW \quad (4)$$

2.4 Technical Complexity Factor(TCF)

Penilaian terhadap tngkat kompleksitas dri setiap faktor, diberikan bobot dari 0 sampai 5. Untuk perhitungan nilai TCF-nya menggunkan rumus berikut ini.

TCF=0, 6 + (0, 01*TF) (5)

Adapun indikator-indikator yang ada dalam perhitungan TCF ini yaitu sebagai berikut.

Tabel 3 Indikator penilaian TCF

Indikator	Deskripsi	Bobot
T1	Kebutuhan System terdistribusi	2.0
T2	Waktu respon	1.0
T3	Efisiensi pengguna	1.0
T4	Kompleksitas proses internal	1.0
T5	Penggunaan kode dari hasil daur ulang	1.0
T6	Kemudahan untuk instal	0.5
T7	Kemudahan untuk digunakan	0.5
T8	Mudah dipakai di berbagai platform	2.0
T9	*Maintenance System*	1.0
T10	Proses paralel	1.0
T11	Fitur keamanan	1.0
T12	Akses pihak ke-3	1.0
T13	Pelatihan pengguna	1.0

2.5 Enviromental Complexity Factor (ECF)

Enviromental Complexity Factor (ECF) juga digunakan untuk menentukan besar nilai usaha. Jangkauan (range) pembobotann yang ada yaitu atara 0(nol) sampai 5. Untuk menghitung ECF, menggunakan rumus berikut.

ECF = 1.4 + (-0.03 x EF) (6)

Adapun indikator yang ada pada perhitungan ECF ini yaitu sebagai berikut.

Tabel 4 Indikator penilaian ECF

Faktor	Deskripsi	Bobot
E1	Familiar dengan proses yang digunakan	1.5
E2	Pengalaman aplikasi	0.5
E3	Pengalaman team terhadap Object Oriented (OOP)	1.0
E4	Kemampuan memimpin analis	0.5
E5	Motivasi tim	1.0
E6	Stabilitas kebutuhan (requirement)	2.0
E7	Pekerja yang paru waktu	-1.0
E8	Tingkat kesulitan Bahasa pemrograman	-1.0

2.6 Nilai usaha *(Effort)*

Nilai usaha *(Effort)* didapatkan dari hasil perkalian antara nilai UCP dengan *Effort* Rate (*ER*). ER merupakan nilai usaha (staff-hour) yang dibutuhkan tiap satu UCP. Rumus perhitungan usaha sebagai berikut [3].

Hours of *Effort* = UCP * Staff-hour per use case point (7)

2.7 Distribusi usaha kedalam aktivitas.

Setelah nilai usaha total didapatkan, langkah selanjutnya yang dilakukan yaitu mendistribusikan usaha kedalam masing-masing aktivitas yang ada dalam pengembangan perangkat lunak. Berdasarkan penelitihan yang dilakukan oleh Kassem Shaleh (2011), aktivitas yang ada pada pengembangan perangkat lunak dibedakan menjadi 3 bagian, yaitu: *Software Development, On Going Activity, dan Quality and Testing*.

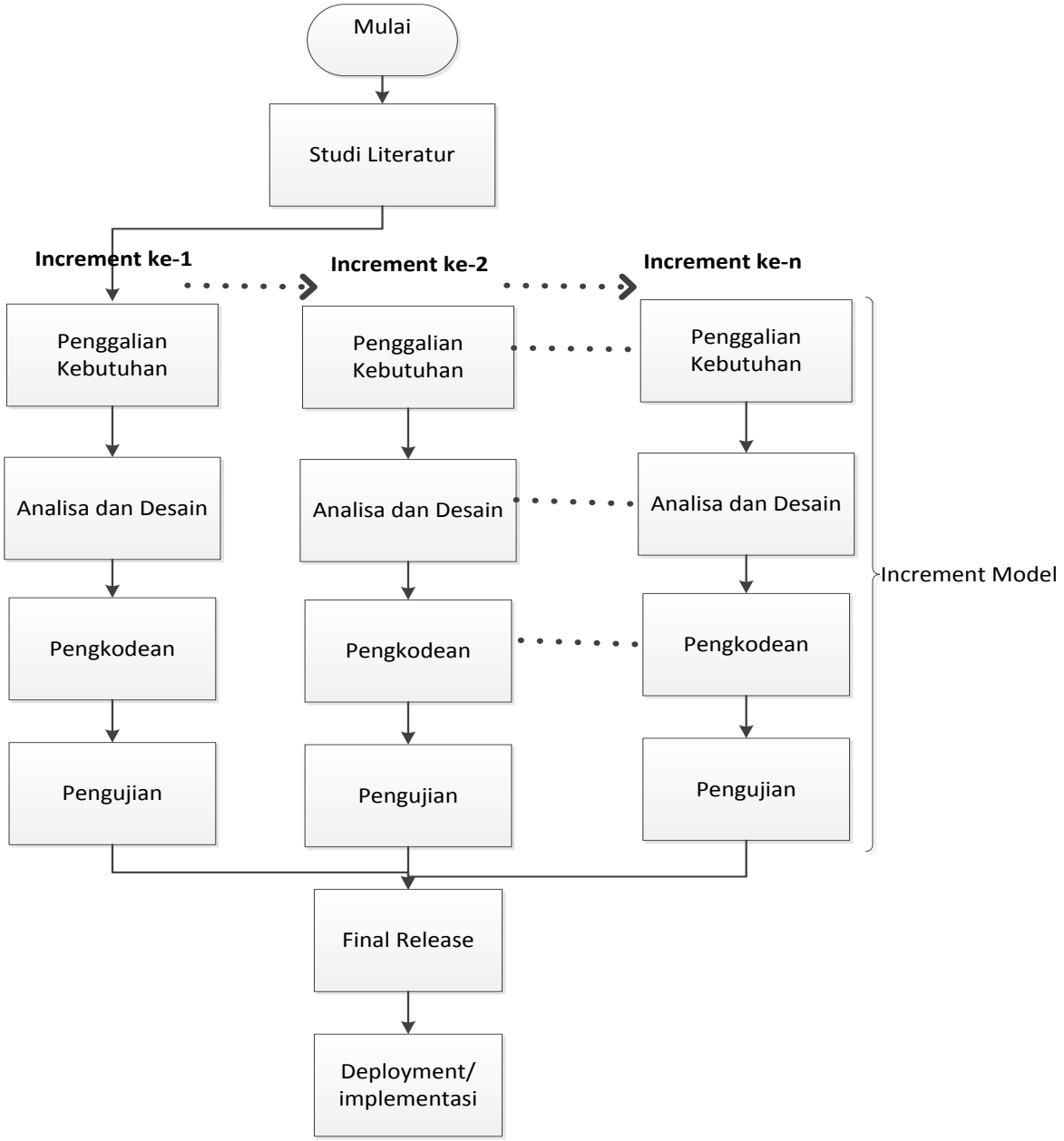
2.8 Mendapatkan nilai biaya

House of *Effort* dari setiap aktivitas dikalikan dengan standard gaji pada masing-masing personel yang ada pada

masing-masing kelompok aktivitas. Berdasarkan penelitihan yang sudah dilakukan menunjkan bahwa gaji dari Kelly Service masih terlalu tinggi, Oleh karena itu, perlu dilakukan penurunan hingga 55% untuk mendekati keakuratan yang lebih baik [4]

3. METODE PENELITIAN

Metode yang digunakan dalam pembangunan perangkat lunak ini terdiri dari mengikuti model pengembangan incremen (*Incremental development model*). Incremental model dipilih karena metode ini dapat meminalisir ketidak sesuaian dalam pengembangan perangkat lunak [5]. Pada metode incement,setiap tahapan yang ada dalam metodologi terdapat masukan(*input*) dan keluaran (*output*). Output dari *increment* akan dujadikan masukan (*input*) untuk increment selanjutnya. Adapun metodologi dalam pengerjaan tugas akhir ini ada pada gambar berikut.



Gambar 1 Metodologi pengerjaan tugas akhir

Berikut adalah detail dari setiap tahapan yang ada pada metodologi pengerjaan tugas akhir ini:

Tabel 5 Detail metodologi pengerjaan tugas akhir

No.	Aktivitas	*Input*	*Output*
1.	Studi Literatur	• Latar Belakang • Permasalahan • Tujuan penelitihan	• Informasi terkait UCP • Informasi metode pengembangan perangkat lunak • List pertanyaan wawancara • Informasi UML • Framework Codeigniter (CI).
2.	Penggalian kebutuhan	List pertanyaan yang digunakan untuk melakukan wawancara	• Dokumen hasil wawancara (interview result), • Informasi proses proses bisnis
3.	Analisa dan Desain	• Informasi Estimasi Biaya dengan UCP • Informasi proses bisnis dari Perusahaan DTS	Model proses bisnis
4.		Model proses bisnis	*Use case diagram, sequence diagram, class diagram,* desain database, desain antarmuka
		• Model proses bisnis • *Use case diagram* • *Sequence diagram* • *Class diagram* • Desain database • Desain antarmuka	• Dokumen SKPL • Dokumen DPPL • Dokumen Testing
5.	Pembangunan Aplikasi	• Dokumen SKPL • Dokumen DPPL	Aplikasi Use Case Point (UCP) yang belum di uji.
6.	Pengujian Aplikasi	Aplikasi UCP yang belum teruji	• Release aplikasi/fitur aplikasi yang sudah teruji • Laporan Hasil Pengujian
7.	Deployment	Aplikasi yang sudah diuji	Aplkasi yang sudah terpasang di server
8	Final Release	Release aplikasi yang sudah teruji dengan semua fitur yang sesuai dengan kebutuhan client	Aplikasi UCP dengan fitur yang sudah lengkap

4. HASIL DAN PEMBAHASAN

Hasil dari penelitihan ini yaitu menghasilkan aplikasi Use Case Point(UCP) untuk estimasi harga perangkat lunak yang lebih akurat dan cepat. Hasil estimasi dengan menggunakan aplikasi UCP ini memiliki tingkat kesalahan (*margin of error*) sebesar 10% terhadap harga proyek perangkat lunak sesungguhnya (*real cost*). Aplikasi sudah teruji melalui 3 tahapan *increment*. Setiap increment menghasilkan aplikasi UCP dengan penambahan fitur yang berbeda. Proses *Reverse Engineering* dilakukan pada tahap *increment* ke-2 karena adanya ketidaksesuaian desain yang sudah dibuat dengan implementasi. Pengujian aplikasi UCP dilakukan terhadap kebutuhan fungsional dan kebutuhan non fungsional. Pengujian fungsional dilakukan dengan menggunakan metode *blackbox testing*. Sekenario untuk pengujian kebutuhan fungsional didapakan dari skenario sukses dan skenario alternatif pada setiap use case fungsional. Pengujian untuk kebutuhan non-fungsional dilakukan dengan melakukan pengujian usabilitas (*Usability Testing*), *Portability Testing*,dan *Security Testing* .

5. KESIMPULAN DAN SARAN

Berikut ini adalah kesimpulan dan saran dari penelitihan yang sudah dilakukan.

5.1 Kesimpulan

Kesimpulan yang didapatkan dari pengerjaan tugas akhir ini yaitu sebagai berikut.

1. Metode pengembangan yang digunakan untuk mengembangkan perangkat lunak UCP yaitu dengan metode incremental dengan tiga kali *increment*.
    a. Tahap increment pertama menghasilkan aplikasi UCP yang bisa melakukan estimasi usaha dengan menggunakan metode UCP
    b. Pada tahap increment kedua, aplikasi hasil dari increment pertama ditambahkan fitur baru sesuai dengan proses bisnis penentuan harga yang ada di perushaan DTS diantaranya yaitu: fitur menghasilkan nilai biaya, dokumen penawaran, notifikasi e-mail, dan pengeloaan apliaksi UCP seperti pengeloaan pengguna, profesi, dan lain-lain.
    c. Pada tahap increment ketiga, aplikasi dari hasil increment kedua ditambahkan beberapa fitur, baik itu fitur fungsional dan fitur non-fungsional. Fitur fungsional diantaranya ditambahkan fitur biaya operasional, fitur informasi *client* apliaksi, fitur nilai ER yang dinamis, dan lain-lain.
2. Tahapan Implementasi aplikasi UCP menghasilkan aplikasi UCP berbasis web yang sudah diuji dengan menggunakan metode pengujian *blackbox testing/correctness testing* untuk pengujian *use case*, *useablity testing* untuk pengujian antarmuka, dan *portability* testing untuk pengujian felibilitas aplikasi dalam device atau browser , dan *security testing* untuk menguji keamanan web.
3. Terdapat beberapa perbaikan dari hasil pengujian, baik itu hasil pengujian untuk kebutuhan fungsional maupun untuk kebutuhan non fungsional.

5.2 Saran

Saran yang bisa diberikan untuk penelitihan selanjutnya yaitu sebagai berikut.

1. Aplikasi UCP juga bisa digunakan untuk melihat avalaible tim. Setiap tim yang masih terlibat dalam pegembangan aplikasi akan tercatat dalam aplikasi UCP sehingga bisa menghindari pemberian beban yang berlebihan pada tim pengembang.
2. Hasil pengujian keamanan mengunakan tools accunetix masih tergolong dalam kategori medium (*level*-2). Kategori ini sudah menunjukan level cukup baik. Namun, untuk kedepan, *level* keamanan perlu dilakukan peningkatan lagi menjadi level *low risk* (level-1).
3. Penggunaan metode increment model dapat membuat waktu pengerjaan aplikasi menjadi lebih lama dan rawan terjadi perluasan ruang lingkup. Oleh karena itu, dalam pengembangan perangkat lunak dengan menggunakan model increment ini, pendefinisan ruang lingkup dan waktu harus jelas dan didefinisikan diawal.

6. DAFTAR PUSTAKA

[1] Edward and Carroll, "Estimating Software Based on Use Case Points.”," *2005 Object-Oriented, Programming,Systems, Languages, and Applications (OOPSLA) Conference, San Diego,CA,* 2005.

[2] C. E R, "Estimating Software Based on Use Case Points," *Object-Oriented, Programming, Systems, Languages, and Object Oriented Programming Systems Languages and Applications (OOPSLA) Conference,* pp. 257-265, 2005.

[3] S. KASSEM, "*Effort* and Cost Allocation in Medium to Large Software Development Projects," *International Journal Of Computers,* vol. 5, no. 1, pp. 75-78, 2011.

[4] P. Linda, Estimasi Biaya proyek Pengembangan Perangkat Lunak Kepemerintahan Berskala Small-Medium Dengan Metode Use Case Point (UCP), Surabaya: Not Yet Published, 2015.

[5] R. S. Pressman, "The Incremental Model," in *Software Engineering, A Practitioner's Approach*, New York, McGraw-Hill Series in Computer Science, p. 36.