 e-ISSN 2338-5197

Pengembangan Kakas Estimasi Perangkat Lunak Dengan *Function Point Dan Use Case Point* Untuk Praktikum Rekayasa Perangkat Lunak

Siti Sariyanti [a,1], Ardiansyah[a,2]

[a] Program Studi Teknik Informatika, Universitas Ahmad Dahlan,
Prof. Dr. Soepomo, S.H., Janturan, Umbulharjo, Yogyakarta 55164
[1]siti12018183@webmail.uad.ac.id; [2]ardiansyah@tif.uad.ac.id

Abstrak

Estimasi biaya perangkat lunak adalah proses yang sangat penting dalam pengembangan perangkat lunak. Estimasi biaya perangkat lunak sangat penting untuk mengontrol dan mengatur efisiensi pada seluruh proses yang dilakukan dalam pengembangan perangkat lunak. Aspek kualitas juga masuk dalam bagian yang sangat penting dalam suatu pengembangan perangkat lunak. Kualitas perangkat lunak tidak hanya dilihat dari hasil produknya tetapi juga kualitas terhadap tahap pengembangan perangkat lunak itu sendiri. Dimana dalam menjamin kualitas perangkat lunak maka perlu dilakukan pengukuran terhadap perangkat lunak. Tujuan penelitian ini adalah untuk pengembangan software kakas estimasi biaya dan usaha untuk proyek pengembangan software yang sesuai dengan kondisi dunia pendidikan. Metodologi yang digunakan dalam penelitian ini adalah metode eksperimen dan metodologi pengembangan dengan waterfall. Dalam tahap eksperimen dilakukan percobaan dan pengolahan informasi yang didapatkan dari wawancara dengan dosen pengampu mata kuliah Rekayasa Perangkat Lunak Universitas Ahmad Dahlan dan studi pustaka. Hasil dari penelitian ini adalah paket aplikasi web yang dapat digunakan untuk mengestimasi biaya dan usaha pada proyek pengembangan software. Kesimpulan yang dapat ditarik dari penelitian ini adalah aplikasi berbasis Web ini dapat diterapkan untuk mengestimasi biaya dan usaha proyek pengembangan software dengan menggunakan alat ukur function dan use case points.

Kata Kunci: Estimasi Kakas Perangkat Lunak, *Use Case Point* (UCP), *Function Point* (FP).

1. Pendahuluan

Kajian Pustaka Penelitian tentang kemampuan dan pentingnya metode-metode estimasi biaya serta pengaruhnya terhadap kesuksesan proyek sangat dibutuhkan dalam proyek perangkat lunak agar manajer mengetahui dan yakin dengan kemampuan dari metode-metode tersebut, sehingga para manajer dapat memilih metode mana yang akan digunakan untuk estimasi biaya bagi proyek perangkat lunak yang akan dikerjakan [1].

Kesulitan-kesulitan yang sering dihadapi dalam estimasi proyek *software* sangat berkaitan dengan sifat alami *software* khususnya kompleksitas dan invisibilitas (keabstrakan). Selain itu pengembangan *software* merupakan kegiatan yang lebih banyak dilakukan secara intensif oleh manusia sehingga tidak dapat diperlakukan secara mekanistik murni [2].

Berdasarkan wawancara dengan Bapak Ali Tarmuji, S.T., M. Cs. Selaku dosen pengampu mata kuliah rekayasa perangkat lunak, ada beberapa aplikasi perangkat lunak yang dapat digunakan dalam dunia pendidikan, baik yang berbayar maupun yang dapat digunakan secara gratis. Pada saat penyusunan modul praktikum Rekayasa Perangkat Lunak di Program Studi Teknik Informatika Universitas Ahmad Dahlan, aplikasi yang lebih tepat untuk menunjang pembelajaran perhitungan estimasi biaya perangkat lunak, dalam praktik perhitungan saat itu hanya aplikasi online yaitu *Tiny Tools* yang dapat menyajikan hasil dari *function point.*

 e-ISSN 2338-5197

Dari kekurangan tersebut maka timbul keinginan untuk mendapatkan suatu kakas estimasi perangkat lunak yang mudah digunakan, serta dapat menjelaskan secara keseluruhan materi yang sesuai dengan perkuliahan. Sehingga muncul gagasan mengembangkan kakas estimasi dengan metode *function Point* dan *use case point.*

2. Kajian Pustaka

Penelitian ini mengacu pada penelitian terdahulu [3]. Pada penelitian tersebut dibahas mengenai tentang Estimasi Kualitas Perangkat Lunak Berdasarkan Pengukuran Kompleksitas Menggunakan *Metrik Function Oriented*. Dalam penelitian tersebut cukup mempunyai keunggulan namun belum dapat menampilkan rumus secara mendetail agar dapat digunakan untuk pembelajaran.

Penelitian ini juga mengacu pada penelitian [4]. Pada penelitian tersebut membahas mengenai Estimasi Proyek Pengembangan Perangkat Lunak dengan Fuzzy Use Case Points. Pada penelitian ini sudah dapat menyimpan data use case matrics dari proyek perangkat lunak namun aplikasi ini harus terus mengembangkan website kedepan agar lebih baik dan mudah digunakan untuk pembelajaran.

Pada penelitian ini akan dikembangkan sebuah kakas yang menjelaskan tahapan perhitungan sebagai pengetahuan, dan dapat menampilkan hasil tes perhitungan dengan *function point* dan *use case point*.

1. Function Point

Function Point adalah sebuah teknik terstruktur dalam memecahkan masalah dengan cara memecah sistem menjadi komponen yang lebih kecil dan menetapkan beberapa karakteristik dari sebuah software sehingga dapat lebih mudah dipahami dan dianalisis [5].

2. Use Case Point

Use Case adalah cara formal yang menggambarkan bagaimana sebuah sistem bisnis berinteraksi dengan lingkungannya [6].

3. CodeIgniter

CodeIgniter adalah sebuah framework PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi *web* berbasis PHP dibandingkan jika menulis semua kode program dari awal[7].

4. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language)* adalah suatu alat bantu yang sering digunakan dalam pengembangan sistem yang bererientasi objek. UML menyediakan bahasa permodelan visual yang memungkinkan pengembang sistem untuk membuat cetak biru atas visi user dalam bentuk yang baku, mudah dimengerti, serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan user dengan yang lain[8].

3. Metode Penelitian

1. Obyek Penelitian

Dalam penelitian ini membahas tentang pengembangan kakas estimasi mengambil objek pengembang perangkat lunak skala kecil yaitu mahasiswa praktikum mata kuliah Rekayasa Perangkat Lunak Universitas Ahmad Dahlan.

2. Function Point

Pada bagian ini menjelaskan tentang tahapan-tahapan perhitungan dengan rumus *function point*.

a. Crude Function Point

Langkah pertama dalam menghitung CFP (Crude Function Points) adalah dengan mencari jumlah dari komponen fungsional sistem pertama kali diidentifikasi dan dilanjutkan dengan mengevaluasi kuantitasi bobot kerumitan dari tiap komponen tersebut. Pembobotan tersebut kemudian dijumlahkan dan menjadi angka CFP.

b. Relative complexity adjuxment factor (RCAF)

Langka kedua untuk menghitung function point adalah dengan menghitung RCAF (Relative Complexity Adjustment Factor), yang dihitung berdasarkan pada keseluruhan kompleksitas sistem. Cara menghitung RCAF (Relative Complexity Adjustment Factor) adalah dengan menggunakan 14 (empat belas) GSC (General System Characteristic), dimana masing-masing GSC berskala 0 (nol) sampai 5 (lima) . Skala 0 (nol) menunjukkan tidak adanya pengaruh dan skala 5 (lima) menunjukkan adanya pengaruh yang luas terhadap keseluruhan proyek.

c. Function point

Setelah setiap karakteristik diberi bobot masing-masing dan dijumlahkan, maka langkah selanjutnya adalah proses melakukan perhitungan untuk mendapat nilai Function point (FP) dari software yang dibangun. Untuk menghitung Function Point menggunakan rumus pada persamaan sebagai berikut :

FP = CFP x (0.65 + 0.01 x RCAF)

Misalkan telah diketahui nilai CFP = 338 dan RCAF = 7. Maka nilai FP dapat dicari sebagai berikut:

FP = 338 x (0.65 + 0.01 x 7)

= 338 x 0.72
= 243.36

d. Line of code (LOC)

Setelah mendapat nilai FP, maka dapat digunakan sebagai acuan untuk mengetahui jumlah LOC yaitu dikalikan dengan bobot bahasa pemrograman yang digunakan.

LOC = FP x Function code point
= 234 x 55
= 13365 baris code

e. In person – months

Effort = 1.4 x LOC

= 1.4 x 13365
= 18.71

f. In months

Time = 3.0 x $Effort^{1/3}$
= 3.0 x $18.71^{1/3}$
= 7.96 *monts*

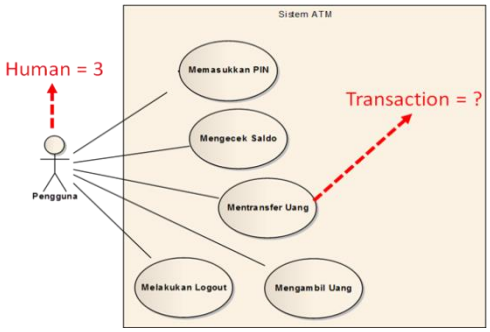
3. Use Case Point

Pada bagian ini menjelaskan tentang tahapan-tahapan perhitungan dengan rumus use case point.

a. Unadjusted Actor Weighting

Setiap palaku dalam sistem diklasifikasi sebagai *simple*, *average*,*complex* dan diberi bobot dengan cara yang sama pada setiap use case. Aktor yang terlibat dalam sistem ATM bersifat berbeda yaitu complex dan average. Berikut salah satu

e-ISSN 2338-5197



Gambar 1. Ilustrasi Sistem ATM

perhitungan nilai UAW sistem ATM :

UAW = ∑(jumlah aktor x bobot aktor)
= (0 x 1) + (0 x 1) + (1 x 3)
= 3

b. Unadjusted Use Case Weight (UUCW)

Unadjusted Use case Weight (UUCW) dihasilkan dari proses perhitungan jumlah use case berdasarkan masing-masing kompleksitas, dikali dengan bobot kompleksitas.

UUCW = ∑(jumlah use case x bobot use case)
= (3 x 5) + (1x 10) + (1 x 15)
= 40

c. Unadjusted Use Case Point

Nilai UAW dan UUCW digunakan untuk menghitung UUCP. Berikut salah satu perhitungan sistem ATM :

UUCP = UAW + UUCW
= 3 + 40
= 43

d. Technical Complexity Factor

Tabel 1. Technical Complexity Factor

No	*Description*	*Weight*	*Value*
T_1	*Distributed Systems*	2.0	5
T_2	*Response time or throughput performance objectives*	1.0	5
T_3	*End-user online efficiency*	1.0	4
T_4	*Complex internal processing*	1.0	5
T_5	*Reusability of code*	1.0	3
T_6	*Easy to install*	0.5	1
T_7	*Ease of use*	0.5	5
T_8	*Portability*	2.0	1
T_9	*Ease of cange*	1.0	4
T_{10}	*Concurrency*	1.0	3
T_{11}	*Special security objectives included*	1.0	5
T_{12}	*Direct access for third parties*	1.0	1
T_{13}	*Special user training required*	1.0	2
Total TF			47

TCF = 0.6 + (0.01 x TF)
= 0.6 + (0.01 x 47)
= 1.07

e. Environmental Complexity Factor

Tabel 2. Environmental Complexity Factor

No	Description	Weight	Value

E_1	Familiarity with system development process being used	1.5	4
E_2	Application experience	0.5	3
E_3	Object-oriented experience	1.0	4
E_4	Lead analyst capability	0.5	4
E_5	Motivation	1.0	3
E_6	Requirements stability	2.0	4
E_7	Part time staff	-1.0	0
E_8	Difficulty of programming language	-1.0	3
Total EF			21.5

ECF = 1.4 + (-0.03 x EF)
= 1.4 + (-0.03 x 21.5)
= 0.77

F_1 = Jumlah E_1 sampai E_6 yang < 3
F_2 = Jumlah E_7 sampai E_8 yang > 3
Jika $F_1 + F_2$ <= 2 maka PHM = 20
Jika $F_1 + F_2$ = 3 atau 4 maka PHM = 28
Jika $F_1 + F_2$ > 4 maka Pikirkan kembali proyek; memiliki risiko kegagalan terlalu tinggi
PHM => E1 – E6 < 3 = 3
E7 – E8 > 3 = 1

4

f. Use Case Point

UCP = UUCP + TCF + ECF
= 43 + 1.07 + 0.77
= 35.43

g. Person month

PM = Effort : 8 : 22
= 708.60 : 8 : 22
= 4.02 Pm

h. Time

Time = 3.0 x $PM^{1/3}$

= 3.0 x $4.02^{1/3}$
= 4.77 *Months*

4. Pengujian

Pengujian Black Box Test disebut juga uji fungsional, yaitu menguji yang mengabaikan mekanisme internal sistem atau komponen dan hanya berfokus pada output dihasilkan dalam menanggapi input yang dipilih dan kondisi eksekusi[9].

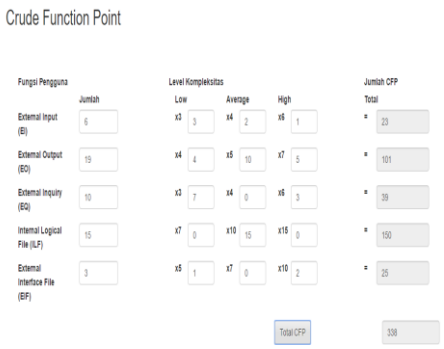
4. Hasil Dan Pembahasan

4.1. Tahapan Perhitungan

a. *Function Point*

Pada perhitungan Crude Function Point (CFP) terdapat beberapa field fungsi pengguna diantaranya adalah field jumlah External Input (EI), External output (EO), External Inquiry (EQ), Internal Logical File (ILF), dan field jumlah External Interface File (EIF).

e-ISSN 2338-5197



Gambar 2. Form Perhitungan *Crude Function Point*

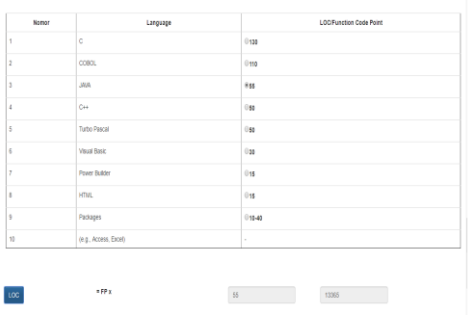
Gambar 3. Hasil RCAF dan FP

b. *Relative Complexity Adjustment Factor*

Halaman ini menampilkan tabel *Relative Complexity Adjustment Factor* yang berisi 14 pertanyaan dengan skore 1-5.

c. Tabel *LOC function point*

Halaman ini menampilkan point dari bahasa pemrograman yang digunakan untuk membangun sebuah aplikasi. Halaman ini penting karena jika tidak dipilih akan mempengaruhi hasil akhir.



Gambar 4. Tabel LOC function point

Gambar 5. Perhitungan *LOC*, *Effort*, dan *Time*

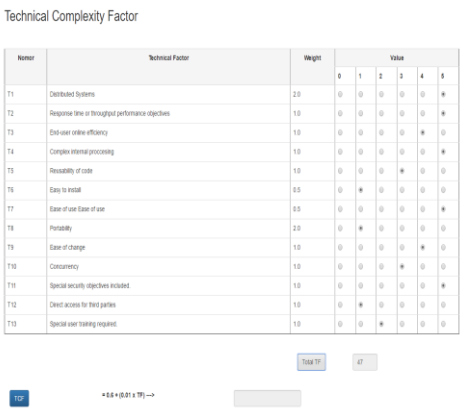
d. Perhitungan *LOC, Effort, dan Time*

Ini adalah inti dari perhitungan function point, pada halaman ini menampilkan semua yang dibutuhkan dalam estimasi biaya perangkat lunak.

e. *Use Case Point*

Menampilkan table dengan 14 pernyataan yang memiliki bobot dan range skore.

 e-ISSN 2338-5197



Gambar 6. Tabel Technical Complexity Factor

Gambar 7. Hasil Akhir use case point

f. Hasil akhir *use case point* keseluruhan

Ini adalah inti dari perhitungan use case point, pada halaman ini menampilkan semua yang dibutuhkan dalam estimasi biaya perangkat lunak.

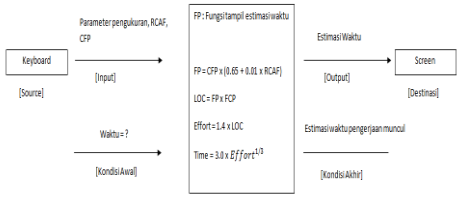
4.2. Pengujian Fungsional

Tampilan pada program, uji akurasi menggunakan metode *Black Box Test*. Spesifikasikan kebutuhan fungsional dalam bentuk form standar Spesifikasi Kebutuhan Sistem (SKS). SKS yang pertama adalah fungsi untuk menampilkan detail barang seperti yang dijelaskan pada tabel 3.

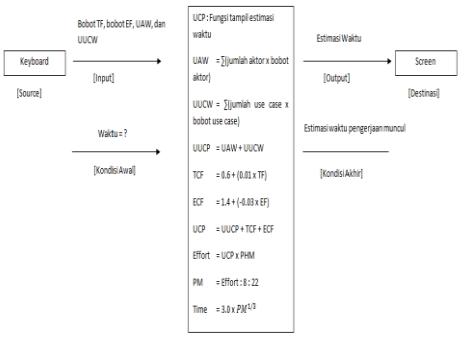
Tabel 3. SKS Fungsi Tampil Estimasi waktu dengan *Function Point*

FP : Tampil Estimasi Biaya dengan *Function Point*	
Fungsi	Menampilkan FP
Deskripsi	Menampilkan hasil estimasi waktu pengerjaan RPL dengan *function point* berdasarkan parameter pengukuran yang diinputkan.
Input	Jumlah CFP, hasil RCAF, hasil FP, hasil LOC, dan hasil Effort
Sumber	Tidak ada
Output	Estimasi waktu pengerjaan proyek
Destinasi	Screen, kotak samping tombol time
Aksi	Setelah menginputkan parameter, maka akan menghasilkan CFP. Selanjutnya hasil CFP akan digunakan untuk menghitung FP, dengan mengalikan CFP dengan FP. Kemudian hasil FP yang dikalikan dengan bobot bahasa pemrograman yang dipakai akan menghasilkan LOC. Selanjutnya LOC akan menghasilkan Effort yang akan menentukan estimasi waktu pengerjaan proyek.
Kebutuhan	Tidak ada
Kondisi Awal	Waktu tidak diketahui
Kondisi Akhir	Estimasi waktu pengerjaan akan muncul
Pengaruh	Tidak ada

Bila divisualisasikan dalam bentuk blok diagram, maka akan tampak seperti gambar 8.



Gambar 8. Blok Diagram Fungsi FP



Gambar 9. Blok Diagram Fungsi UCP

Setelah memanggil fungsi FP, maka dilanjutkan untuk menjalankan fungsi UCP yaitu menghitung estimasi waktu pengerjaan dengan use case point. SKS fungsi ini ditunjukkan pada tabel 4.

Tabel 4. SKS Fungsi Tampil Estimasi waktu dengan *Use Case Point*

UCP : Tampil Estimasi Biaya dengan *Use Case Point*	
Fungsi	Menampilkan UCP
Deskripsi	Menampilkan hasil estimasi waktu pengerjaan RPL dengan *Use Case Point* berdasarkan *use case*.
Input	Bobot TF, bobot EF, UAW, dan UUCW
Sumber	Tidak ada
Output	Estimasi waktu pengerjaan proyek
Destinasi	Screen, kotak samping tombol time
Aksi	Setelah menginputkan masing-masing bobot TF dan EF, maka akan diketahui jumlah TF dan EF. Selanjutnya TF digunakan untuk menentukan TCF, dan EF untuk menentukan ECF. Kemudian menghitung masing-masing bobot UAW berdasarkan jumlah dan bobot aktor, menghitung UUCW berdasarkan jumlah use case dan bobot use case. Dengan menjumlahkan UAW dan UUCW akan menghasilkan UUCP. Selanjutnya untuk menghasilkan UCP perlu menjumlahkan TCF, ECF, dan UUCP. Kemudian Ucp akan menghasilkan Effort yang akan menentukan estimasi waktu pengerjaan proyek.
Kebutuhan	Tidak ada
Kondisi Awal	Waktu tidak diketahui
Kondisi Akhir	Estimasi waktu pengerjaan akan muncul
Pengaruh	Tidak ada

Bila divisualisasikan dalam bentuk blok diagram, maka akan tampak seperti gambar 9.

5. Penutup

5.1. Kesimpulan

A. Telah dikembangkan Kakas Estimasi Perangkat Lunak dengan *Function Point* Dan *Use Case Point* yang mampu menampilkan *effort* dan estimasi waktu pengerjaan dengan dua metode yang berbeda.

B. Kakas Estimasi Perangkat Lunak berjalan di aplikasi web yang mampu menampilkan rumus dan tahapan perhitungan yang mendetail. Sehingga pengguna dapat dengan mudah mengingat dan memahami materi perkuliahan.

e-ISSN 2338-5197

C. Kakas Estimasi Perangkat Lunak telah diuji dengan hasil semua fungsi yang ada dapat berjalan sesuai dan siap digunakan.

5.2. Saran

Pengembangan kakas estimasi perangkat lunak dengan *function point* dan *use case point* masih mengalami keterbatasan, maka diharapkan adanya pengembangan. Kekurangan pada kakas estimasi ini adalah tampilan antar muka atau *user interface* yang kurang menarik dan *user friendly*, diharapkan untuk memperbaiki tampilan pada kakas estimasi perangkat lunak sehingga tampilan lebih menarik dan *friendly*.

Penambahan *database* pada kakas estimasi perangkat lunak akan menambah fungsi guna kakas estimasi perangkat lunak. Sehingga pengguna dapat menyimpan hasil perhitungan kedalam komputer yang sedang digunakan.

Daftar Pustaka

[1] Bintiri, M. G., Sn, A., & Dillak, R. Y. (2012). "*Perbandingan Model Aloritmik Dan Non Algoritmik Untuk Estimasi Biaya Perangkat Lunak"*, Seminar Nasional Aplikasi Teknologi Informasi. ISSN: 1907-5022. Yogyakarta.

[2] Prasetyo, S. and B. (2006). "Penggunaan Model Function Point dalam Estimasi Biaya dan Usaha Proyek Pengembangan Software Sistem Informasi Bisnis". Risalah Lokakarya Komputasi Dalam Sains Dan Teknologi Nuklir XVII. Batan.

[3] Hapsari, R. K., & Husen, M. J. (2015). "*Estimasi Kualitas Perangkat Lunak Berdasarkan Pengukuran Kompleksitas Menggunakan Matriks Function Oriented"*. Seminar Nasional Sains Dan Teknologi Terapan III. ISBN: 978-602-98569-1-0.

[4] Hariyanto, M., & Mandiri, N. (2015). "*Estimasi Proyek Pengembangan Perangkat Lunak Dengan Fuzzy Use Case Points"*. Jakarta: Program Studi Sistem Informasi, STMIK Nusa Mandiri.

[5] Hapsari, R. K., & Husen, M. J. (2015). "*Estimasi Kualitas Perangkat Lunak Berdasarkan Pengukuran Kompleksitas Menggunakan Matriks Function Oriented"*. Seminar Nasional Sains Dan Teknologi Terapan III. ISBN: 978-602-98569-1-0.

[6] Dennis, A., Wixom, B.H. & Tegarden, D., 2009. *SYSTEMS ANALYSIS AND DESIGN WITH UML VERSION 2.0* 3rd ed., United States of America: John Wiley & Sons, Inc.

[7] Koespradono, Suraya & Yuliana R.K. (2013). Sistem Informasi Pengolahan Data Pertumbuhan Ekonomi Dan Ketimpangan Di Kabupaten Klaten (Tahun 2003-2012) Menggunakan Framework Codeigniter. ISSN:2338-6304. Yogyakarta.

[8] Sholiq . (2006). "Pemodelan Sistem Informasi Berorientasi Objek dengan UML". Graha Ilmu. Yogyakarta.

[9] Simarmata, Janner. 2010. "*Rekayasa Perangkat Lunak"*. Yogyakarta: Andi.