

FPA Ebay

by Soetam Rizky Wicaksono

Submission date: 10-Mar-2020 06:57AM (UTC-0500)

Submission ID: 1272917355

File name: 2764-5429-1-SM.pdf (359.56K)

Word count: 2586

Character count: 15897

**ANALISIS KUALITAS PERANGKAT LUNAK MENGGUNAKAN
METODE FUNCTION POINT ANALYSIS
(STUDY KASUS: Transaksi Pembelian di eBay)**

Rambu Beatrix Hamu Eti Kapita^{1*}, Mayank Angelin², Soetam Rizky Wicaksono³

^{1,2,3} Fakultas Sains dan Teknologi, Universitas Ma Chung, Malang

^{*}Email: 321510008@student.machung.ac.id

Abstrak

Pada era teknologi yang ada saat ini banyak perangkat lunak bermunculan dan bersaing dalam bidang yang sama. Seperti halnya eBay yang merupakan situs ecommerce, akan tetapi dulunya eBay merupakan situs lelang yang menjual barang-barang yang masuk dalam kategori 'collectible' atau barang langka. Dunia teknologi terus berkembang pesat, begitu pula dengan eBay, eBay dituntut untuk mengembangkan websitenya untuk memenuhi permintaan yang ada juga untuk kesiapan dalam bersaing dengan ecommerce lainnya. Dalam mengukur estimasi suatu perangkat lunak salah satunya dapat dilakukan menggunakan metode Function Point Analysis (FPA). Function Point Analysis merupakan metode yang diadopsi berdasarkan standard ISO untuk pengukuran fungsionalitas perangkat lunak dari IFPUG. Dari pengukuran dan perhitungan yang dilakukan menggunakan metode FPA menghasilkan nilai UFP=219, TDI=62, TCA=1.27, FP=139.3 dan LOC≈8079.

Kata kunci: Function Point Analysis, pengukuran perangkat lunak

PENDAHULUAN

eBay merupakan situs yang mempertemukan antara penjual dan pembeli yang didirikan oleh Pierre Omidyar dengan tujuan sebagai situs lelang, bagian dari situs pribadi yang termasuk, tulisan sumbangan Omidyar terhadap virus Ebola. Awalnya situs ini dimiliki oleh *Echo Bay Technology Group*. eBay bermarkas di San Jose, California, Meg Whitman telah menjadi presiden eBay dan CEO-nya sejak Maret 1998.

Kesuksesan eBay telah menjadi sumber ide untuk perusahaan lelang lainnya di seluruh dunia (Jones et al. 2010). Ide ini merupakan salah satu contoh dari fenomena "*Long Tail*" yang menyatakan tentang adanya pasar ideal pada saat beragam barang yang dijual semakin banyak, sementara jumlahnya semakin sedikit.

eBay adalah sebuah situs lelang yang berdiri pada tahun 1995 di San Jose, CA. Pada saat awal memiliki nama *AuctionWeb*. Item pertama yang dijual di situs oleh pendirinya adalah laser *pointer* dalam keadaan rusak, untuk seorang kolektor. Selanjutnya nama tersebut berganti menjadi eBay pada bulan September, 1997 (Jones et al. 2010)

eBay adalah *ecommerce* yang mempertemukan penjual dan pembeli dari seluruh dunia untuk dapat melakukan transaksi jual beli. Model situs seperti ini banyak sekali beredar di situs internet, tetapi yang

membedakan eBay dengan situs *ecommerce* lainnya, adalah awalnya eBay merupakan situs lelang yang menjual barang-barang yang masuk dalam kategori 'collectible' atau barang langka, contohnya koin kuno, prangko, dll. pengembangan yang dilakukan eBay dari situs lelang barang hingga menambah kategori merupakan bukti nyata kesiapan eBay bersaing dengan situs *ecommerce* lainnya.

Dalam analisis kualitas perangkat lunak yang dilakukan mengangkat eBay sebagai dalam study kasus penelitian. eBay merupakan situs *ecommerce* yang besar sehingga sangat diperlukan untuk mengukur kualitasnya. Pengukuran kualitas pada penelitian ini memiliki batasan hingga bagian transaksi pembelian. Ada banyak metode yang dapat digunakan dalam melakukan pengukuran kualitas perangkat lunak, salah satunya yaitu menggunakan metode *Function Point Analysis* (FPA).

Salah satu metode yang umum digunakan dalam mengukur estimasi ukuran dari perangkat lunak adalah *Function Point Analysis* (FPA) (Balaji 2013; Latchoumi 2013). Metode yang diadopsi berdasarkan standard ISO untuk pengukuran fungsionalitas perangkat lunak dari IFPUG. Penelitian juga dilakukan untuk meningkatkan akurasi estimasi dengan pendekatan sistem cerdas logika *fuzzy* (Ferrucci, Gravino, and Sarro 2014) (Murti

2016)(Saptono and Utama 2015) sehingga hasil yang didapat menjadi lebih baik.

Metode *function point analysis* telah terbukti sebagai metode yang dapat diandalkan untuk mengukur ukuran perangkat lunak komputer. Selain mengukur output, Function point analysis sangat berguna dalam memperkirakan proyek, mengelola perubahan lingkup, mengukur produktivitas, dan mengkomunikasikan persyaratan fungsional (Rachmat and Kunci 2017).

Sehingga dalam penelitian ini dilakukan pengukuran terhadap situs eBay agar dapat membuat penyampaian *scope* dari aplikasi tersebut dengan membandingkan beberapa *functional area* yang terdapat pada eBay.

METODE PENELITIAN

FPA (*Function Point Analysis*) pertama kali dikenalkan oleh Allan Albrecht pada tahun 1979 dan sekarang terus diperbaharui oleh International Function Point User Group (IFPUG)(Ferrucci, Gravino, and Sarro 2014). Tujuan dari pengukuran menggunakan FP (*Function Point*) adalah untuk mendapatkan ukuran biaya, durasi, dan jumlah sumber daya yang diperlukan oleh sebuah proyek perangkat lunak dan dapat dilakukan pada setiap tahap pengembangan perangkat lunak. Pengukuran FP membutuhkan keahlian analisis perangkat lunak dan hasil perhitungannya bisa dinyatakan valid jika dilakukan oleh seseorang yang mempunyai sertifikasi FP (certified FP) dari IFPUG (Irawati and Mustofa 2013) (About Function Point Analysis - IFPUG n.d.)

Beberapa penelitian mengenai metode *Function Point* (FP) telah dilakukan, misalnya penelitian yang menjelaskan contoh perhitungan FP pada produk *software*(Kusrini and Yogyakarta 2015). Jumlah Proyek dapat secara luas diklasifikasikan menjadi tiga jenis seperti yang didefinisikan dalam *Function Point Manual* oleh David H. Longstreet.(Irawati and Mustofa 2013)

Kegiatan melakukan *Function Point Analysis* sering disebut sebagai 'Perhitungan Poin Fungsi' dan melibatkan identifikasi, klasifikasi dan pembobotan dari masing-masing komponen proses dan grup data ini. Pembobotan digabungkan untuk memberikan ukuran fungsional sebagai Jumlah Titik Fungsi Tidak Disesuaikan (UFP). Ini adalah ukuran fungsional sebagaimana didefinisikan oleh ISO

/ IEC standard 20926, IFPUG CPM 4.3 dan ISO / IEC 14143-1(Irawati and Mustofa 2013)

Untuk melakukan *function point analysis*, pertama-tama kita harus mendefinisikan batas penggunaan aplikasi. Batasan digunakan untuk menentukan apa yang internal untuk aplikasi dan komponen atau antarmuka apa yang berada di luar aplikasi. Pembedaan ini penting dan setelah batas ditetapkan, itu tetap konstan sepanjang umur aplikasi, dan digunakan untuk semua estimasi masa depan.

Dalam metode *Function Point Analysis* terdapat 5 fungsi sebagai parameter pengukuran sebuah perangkat lunak: (1) Jumlah input eksternal (**External Input – EIs**). Input berasal dari luar sistem, baik dari user maupun sistem lainnya yang selanjutnya digunakan untuk mengupdate Internal Logical Files, (2)Jumlah output eksternal (**External Output – EOs**). Output merupakan data yang ditampilkan pada aplikasi untuk menyediakan Informasi kepada user, baik dalam bentuk laporan, tampilan di layar, pesan error, dst, (3) Jumlah inquiries eksternal (**External Inquiries – EQs**). Inquiries eksternal didefinisikan sebagai input online yang memicu respon dari software untuk menghasilkan output online, (4) Jumlah file logikal internal (**Internal Logical Files – ILFs**). File logika internal merupakan data yang dikelompokkan secara logis, disimpan secara internal dan didapat dari input eksternal, dan (5) Jumlah file interface eksternal (**External Interface Files – EIFs**). Merupakan data yang dikelompokkan secara logis namun berada diluar aplikasi yang menyediakan Informasi yang dibutuhkan aplikasi. (Hapsari and Husen 2015)

Sedangkan ukuran kualitatif kompleksitas perangkat lunak ditentukan dari 14 faktor nilai penyesuaian (*Value Adjustment Factor*) sebagai berikut : (1) *ID Data communications*, (2) *Distributed function*, (3) *Performance objectives*, (4) *Heavily used configuration*, (5) *Transaction rate*, (6) *On-line data entry*, (7) *End-user efficiency*, (8) *On-line update*, (9) *Complex processing*, (10) *Reusability*, (11) *Installation ease*, (12) *Operational ease*, (13) *Multiple sites*, dan (14) *Facilitate change* (Rachmat and Kunci 2017)

Untuk setiap GSC memiliki peringkat 0-5 yang mewakili :

0 = Tidak ada pengaruh

1 = Pengaruh Rendah

- 2 = Lumayan Berpengaruh
- 3 = Agak Berpengaruh
- 4 = Pengaruh yang *Significant*
- 5 = Sangat Berpengaruh

Proses perhitungan melibatkan penghitungan DETS dan RET dalam fungsi data (tabel dan kelompok individual) dan penghitungan DET dan FTR dalam fungsi transaksi (Kusrini and Yogyakarta 2015)

Setelah menentukan keseluruhan faktor dan skor masing-masing, selanjutnya yaitu menghitung nilai Total *Degree of Influence* (TDI) dengan menjumlahkan ke 14 faktor tersebut. Lalu TDI diubah menjadi nilai akhir Technical Complexity Adjustment dengan formula.

Rumus yang digunakan untuk menghitung total nilai *Unadjusted Function Point* (UFP), *Technical Complexity Adjustment* dan Nilai *Function Point* dan Estimasi Ukuran adalah sebagai berikut :

$$UFP = \sum_{i=1}^5 \sum_{j=1}^3 x_{ij} \times w_{ij} \quad (1)$$

$$TCA = 0.65 + 0.01 \times TDI \quad (2)$$

$$FP = UFP \times TCA \quad (3)$$

Tabel.1 Productivity Factor Programming Language

Programming Language	Productivity Factor
SQL	37
PHP	53
HTML/Javascript	58

HASIL DAN PEMBAHASAN

Dalam penelitian ini, kasus yang diambil adalah sebuah *E-commerce* berbasis web. Penerapan FPA pada study kasus yang diambil, langkah yang harus diambil antara lain :

1. Melakukan proses identifikasi dan menghitung jumlah fungsionalitas dan kompleksitas dalam perangkat lunak dari sudut pandang pengguna.
2. Setelah melakukan proses identifikasi, setiap UF dalam perangkat lunak harus dihitung kompleksitasnya. Setiap UF diklasifikasikan berdasarkan tingkat kompleksitas antara lain *height*, *medium* (*average*) dan *low*. Nilai dari setiap UF pada masing-masing kategori kompleksitas dikalikan dengan *complexity weight* yang telah ditentukan pada Tabel 1.

3. Menghitung Total Nilai *Unadjusted Function Point* (UFP)

Setelah tingkat kompleksitas dari setiap jenis UF diidentifikasi maka selanjutnya adalah menghitung Nilai *Unadjusted Function Point* (UFP) dengan formula.

$$UFP = \sum_{i=1}^5 \sum_{j=1}^3 x_{ij} \times w_{ij} \quad (4)$$

UFP dihitung dengan menjumlahkan *weight* dari sejumlah UF dimana adalah sejumlah UF jenis *i* dengan kompleksitas level *j*, dan adalah nilai *weight* untuk jenis *i* dengan kompleksitas level *j*. Dengan kata lain UFP didapatkan dengan mengalikan nilai *weight* dengan jumlah fitur yang ada di setiap UF yang berbeda. Kemudian hasil yang didapat kelima fungsi dijumlahkan untuk mendapatkan nilai Total *Unadjusted Function Point* (TUFP).

4. Menghitung Technical Complexity Adjustment

Pada FPA diperlukan perhitungan tidak hanya jumlah kompleksitas dari beberapa fitur yang diberikan kepada pengguna, tetapi juga kepada operasional dari lingkungan sistem. Ada 14 faktor yang dibuat mempengaruhi tingkat kesulitan yang berhubungan dengan implementasi sistem. Setiap faktor diberi nilai dari 0 sampai 5. Nilai 0 jika faktor tersebut tidak menimbulkan efek apa pun dan 5 jika faktor tersebut sangat penting di perangkat lunak yang diukur. Dalam FPA, faktor tersebut disebut *Value Adjustment Factor* seperti terlihat pada Tabel 2.

5. Menghitung Nilai *Function Point* dan Estimasi Ukuran

Nilai *Function Point* (FP) dari perangkat lunak dihitung dengan mengalikan UFP dengan TCA sesuai formula.

$$FP = UFP \times TCA \quad (5)$$

Nilai FP dapat digunakan untuk mengukur LOC perangkat lunak, *cost*, *effort*, *resource* yang dibutuhkan dan lama pengerjaan perangkat lunak. Untuk memperoleh estimasi ukuran perangkat lunak dalam satuan *Lines of Code* (LOC), nilai FP dikalikan dengan *Productivity Factor* berdasarkan bahasa pemrograman yang akan digunakan. *Productivity Factor* adalah jumlah kode logis per *function point* dan nilainya

bervariasi untuk setiap bahasa pemrograman, seperti yang dibuat oleh Capers Jones pada Tabel 2.

Tabel 2. Klasifikasi

No.	Deskripsi	Kompleksitas
<i>External Input</i>		
1.	Halaman registrasi	Low
2.	Halaman login	Low
3.	Halaman search	Medium
4.	Halaman shopping chart	Medium
5.	Halaman sign in with google	Low
6.	Halaman sign in with facebook	Low
7.	Konfigurasi sistem	Low
8.	Quantity box	Low
9.	Halaman review	Low
10.	Address box	Height
11.	Additional information box	Height
12.	City box	Height
13.	State/province/region box	Height
14.	Postal code box	Height
15.	Mobile number box	Height
16.	Pay option	Height
17.	Input coupons	Medium
18.	Input card number	Medium
19.	Input expiration date	Medium
20.	Input security code	Medium
<i>External Output</i>		
1.	Cetak invoice	Medium
2.	Cetak resi pengiriman	Height
<i>Internal Logical File</i>		
1.	Gambar barang	Medium
2.	Database tabel seller	Height
3.	Database tabel buyer	Height
4.	Daftar harga	Medium
5.	List shopping chart	Medium
6.	History pembelian	Medium
<i>External Interface File</i>		
1.	Google	Height
2.	Facebook	Height
3.	Paypall	Height
<i>External Inquiry</i>		
1.	Lacak pengiriman	Medium
2.	Proses shipping	Height
3.	Proses pembatalan pembelian	Medium

Tingkat kompleksitas dari setiap jenis UF diidentifikasi akan dijumlahkan untuk mendapat hasil perhitungan UFP dengan menggunakan rumus (1) seperti pada Tabel 3.

Tabel 3. Perhitungan Nilai UFP

User function types	Complexity Weight x Counts			
	Low	Medium	Height	Total
External Input (EI)	3x7	4x6	6x7	87
External Output (EO)	0	5x1	7x1	12
Internal Logical File (ILF)	0	10x4	15x2	70
External Interface File (EIF)	0	0	10x3	30
External Inquiry (EQ)	0	4x2	6x2	20
Unadjusted Function Point (UFP)				219

Berdasarkan klasifikasi yang telah dibuat selanjutnya penghitungan nilai UFP dilakukan dengan mengkalikan UF dengan nilai kompleksitas masing lalu menjumlahkan keseluruhan nilai yang menghasilkan angka 219. Pada perhitungan ini nilai yang berada di depan merupakan nilai mutlak, nilai yang dibelakang merupakan hasil perhitungan dari UF. External input memiliki jumlah terbanyak, karena aktivitas yang dilakukan lebih banyak di luar sistem.

Tabel 4. Perhitungan Nilai TDI

ID	FACTOR	DI
C1	Data communications	5
C2	Distributed function	5
C3	Performance objectives	4
C4	Heavily used configuration	5
C5	Transaction rate	5
C6	On-line data entry	5
C7	End-user efficiency	4
C8	On-line update	3
C9	Complex processing	5
C10	Reusability	4
C11	Installation ease	3
C12	Operational ease	5

-C13	Multiple sites	4
C14	Facilitate change	5
Total Degree of Influence		62

Setelah mendapatkan nilai UFP, selanjutnya adalah menghitung nilai TDI yang hasil perhitungannya adalah 62 dengan menggunakan *Value adjustment Factor*. Hasil dari perhitungan TDI dimasukkan ke dalam formula $TCA = 0,65 + 0,01 \times TDI$ dan menghasilkan nilai TCA sebesar 1,27.

$$TCA = 0,65 \times 0,01 \times 62 = 1,27$$

Untuk mendapatkan nilai estimasi ukuran dari sistem dalam Lines of Codes (LOC), maka nilai FP dikalikan dengan 58 yang merupakan productivity factor dari bahasa pemrograman HTML/Javascript, sehingga LOC didapatkan seperti pada formula di bawah ini :

$$FP = 427 \times 1,08 = 139,3$$

Untuk mendapatkan nilai estimasi ukuran dari sistem pembelian dalam *Lines Of Codes* (LOC), maka nilai FP dikalikan dengan 58 yang merupakan *Productivity Factor* dari bahasa pemrograman HTML/Javascript. Sehingga LOC yang didapatkan adalah sebagai berikut ;

$$LOC = 139,3 \times 58 \cong 8079$$

Ukuran proyek telah diestimasi dengan nilai estimasi 8079 baris kode dengan menggunakan metode FPA.

KESIMPULAN DAN SARAN

Dalam penelitian yang telah dilakukan ini dapat memberikan penjelasan secara detail apa itu metode *function point analysis* yang digunakan untuk melakukan pengukuran estimasi perangkat lunak. FPA pada penelitian ini diimplementasikan pada study kasus EBay untuk mengukur bagian transaksi pembelian. Hasil yang diperoleh dalam pengukuran menunjukkan keakurasian yang tinggi. Melalui penelitian ini diharapkan mampu menjadi pembelajaran bagi penulis dan pembaca untuk mengembangkan aplikasi lain kedepannya.

Perhitungan dimulai dari *external input, external output, internal logical file, internal input, external interface file, external inquiry* yang kemudian dilakukan perhitungan untuk menemukan nilai UFP, nilai TDI, TCA, FP dan LOC. Dari pengukuran dan perhitungan yang dilakukan menghasilkan nilai UFP=219, TDI=62, TCA=1,27, FP=139,3 dan LOC≅8079.

UCAPAN TERIMAKASIH

Terima kasih kepada :

1. Lembaga Universitas Ma Chung yang telah memberikan sarana dalam melakukan penelitian.
2. Dosen pembimbing dalam pembuatan penelitian mengenai analisis kualitas perangkat lunak menggunakan metode *function point analysis*.
3. Rekan yang telah membantu dalam pelaksanaan penelitian dan penulisan.

DAFTAR PUSTAKA

- "About Function Point Analysis - IFPUG." <http://www.ifpug.org/about-function-point-analysis/> (December 5, 2018).
- Balaji, N. 2013. "Software Cost Estimation Using Function Point with Non Algorithmic Approach." *Global Journal of ...* 13(8). <http://computerresearch.org/stpr/index.php/gjcs/article/viewArticle/1484>.
- Ferrucci, Filomena, Carmine Gravino, and Federica Sarro. 2014. "Conversion from IFPUG FPA to COSMIC: Within-vs without-Company Equations." *Proceedings - 40th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2014* (December 2015): 293–300.
- Hapsari, Rincian Kembang, and M Jauhari Husen. 2015. "Estimasi Kualitas Perangkat Lunak Berdasarkan." *Seminar Nasional Sains dan Teknologi Terapan III*: 425–34.
- Irawati, Anie Rose, and Khabib Mustofa. 2013. "Pengukuran Fungsionalitas Perangkat Lunak Menggunakan Metode Function Point Berdasarkan Dokumentasi Desain." 7(2): 111–20.
- Jones, Alice P. et al. 2010. "EBay Analysis." *Journal of Child Psychology and Psychiatry* 51(11): 1188–97. <http://doi.wiley.com/10.1111/j.1469-7610.2010.02280.x>.
- Kusrini, Kusrini, and Universitas Amikom Yogyakarta. 2015. "Pengukuran Volume Software Berdasarkan Kompleksitasnya Dengan Metode Function Point Kusrini, Muhammad Dedi Iskandar." m(September).
- Latchoumi, T P. 2013. "International Journal of Advanced Research in Synthetic Identity of Crime Detection." 3(7): 551–60.

- Murti, Sambodo Wisnu. 2016. "Comparison Analysis of Weight Value Changing in Function Point Analysis Between Fuzzy Inference System Mamdani and Tsukamoto for Software Size Estimation." 5(2): 104-10.
- Rachmat, Nur, and Kata Kunci. 2017. "Estimasi Ukuran Perangkat Lunak Menggunakan Function Point Analysis - Study Kasus Aplikasi Pengujian Dan Pembelajaran Berbasis Web." 3(1).
- Saptono, Ristu, and Galih Dian Utama. 2015. "Peningkatan Akurasi Estimasi Ukuran Perangkat Lunak Dengan Menerapkan Logika Samar Metode Mamdani." *Scientific Journal of Informatics* 2(1): 41-52.

FPA Ebay

ORIGINALITY REPORT

20%

SIMILARITY INDEX

21%

INTERNET SOURCES

2%

PUBLICATIONS

2%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

6%

★ jurnal.uns.ac.id

Internet Source

Exclude quotes On

Exclude bibliography On

Exclude matches < 2%