IJCCS, Vol.7, No.2, July 2013, pp. 111~120
ISSN: 1978-1520

Pengukuran Fungsionalitas Perangkat Lunak Menggunakan Metode Function Point Berdasarkan Dokumentasi Desain

Anie Rose Irawati*[1], Khabib Mustofa[2]
[1]Mahasiswa Prodi.S2/S3 Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM
[2]Staf pengajar Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM
e-mail: [1*]**anie_rose@unila.ac.id**, [2]khabib@ugm. ac.id

Abstrak

Perkiraan nilai perangkat lunak yang disepakati oleh end user dan tim pengembang harus dinyatakan dalam besaran tertentu, salah satunya adalah dalam u ku ran fungsionalitas (functional size measurement/FSM).

Metode Function Point (FP) merupakan salah satu metode yang digunakan untuk mendapatkan ukuran fungsionalitas, pertama kali dikenalkan oleh Allan Albrecht dan kemudian terus dikembangkan sampai saat ini oleh International Function Point User Goup (IFPUG). Tujuan dari pengukuran menggunakan FP 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 memerlukan keahlian dalam hal analisis perangkat lunak dan hasil perhitungannya dinyatakan valid jika dilakukan oleh seseorang yang mempunyai sertifikasi FP (certified FP) dari IFPUG.

Penelitian ini bertujuan untuk membangun sebuah sistem yang dapat memberikan kemudahan bagi pelaku pengukuran perangkat lunak dalam menganalisis perangkat lunak dengan metode Function Point berdasarkan pada IFPUG CPM 4.3.1.

Penelitian ini menghasilkan sistem yang membantu pengguna untuk melakukan analisis FP dengan cepat dan dengan validitas yang tinggi. Sistem yang dibuat mampu menghitung FP sesuai dengan dokumentasi yang dimiliki software dan menggunakan input berupa dokumen XMI yang di-eksport dari Use Case Diagram, Class Diagram dan perpaduan antara Use Case dan Class. Selain itu, penelitian ini menunjukkan pola bahwa semakin lengkap dokumen UML (Unified Modeling Language) yang dimiliki oleh sebuah perangkat lunak maka semakin akurat hasil perhitungan FP yang didapatkan.

Kata kunci— *Pengukuran perangkat lunak, UML, Function Point, dokumentasi desain perangkat lunak.*

Abstract

Estimated value of software as agreed by the end user and the developer team should be expressed in a certain magnitude, one of which is the measure of functionality (functional size measurement / FSM).

Function Point (FP) Method is one of the methods used to obtain the size of the functionality, first introduced by Allan Albrecht and then further developed by the International Function Point User Goup (IFPUG). FP is a method to get a measure of the cost, duration, and amount of resources required by a software project and can be done at any stage of software development. However, Function Point measurement requires expertise in software analysis and the results of the calculation are valid if done by someone with IFPUG certification.

Received June 1st,2013; Revised July 1st, 2013; Accepted July 15th, 2013

This study realizes a system that is convenient for actors in analyzing software using Function Point method based on IFPUG CPM 4.3.1.

The system helps users to perform FP analysis in a faster way and confers accurate results. The system is able to count FP based on software design documentation and uses XMI document exported from use case diagrams, class diagrams, and diagrams illustrating the use cases and classes relationship as an input. The study also indicates that the more complete software UML (Unified Modeling Language) documents, the more accurate the FP calculation results obtained.

Keywords— *Software measurement, UML, Function Point, software design documentation.*

1. PENDAHULUAN

Setiap proyek TI yang baik seharusnya dimulai dengan perkiraan usaha, biaya, jadwal dan waktu, sebagai dasar perencanaan proyek dan sebagai ukuran kesuksesan diakhir proyek tersebut. Kegiatan memperkirakan tersebut dilakukan dengan mengetahui ukuran dari obyek yang akan dikembangkan terlebih dahulu, sehingga pengukuran menjadi elemen kunci dalam penilaian. Ukuran juga model dan produk yang dibuat [1].

Metode Function Point yang pertama kali diusulkan oleh Allan Albrecht dan kemudian terus dikembangkan sampai saat ini oleh IFPUG (*International Function Point User Goup),* adalah salah satu metode yang digunakan untuk mendapatkan ukuran fungsionalitas (*functional size)* perangkat lunak, yaitu ukuran yang menunjukkan seberapa besar fungsi yang disediakan oleh perangkat lunak kepada *user*-nya. Tujuan utama dari pengukuran perangkat lunak dengan Function Point adalah untuk mendapatkan ukuran biaya proyek, durasi proyek, dan jumlah sumber daya yang diperlukan oleh sebuah proyek [2].

IFPUG dalam panduannya mengungkapkan bahwa komponen-komponen yang dinilai dalam FPA tidak dapat diukur dengan mudah karena memerlukan pemahaman dan analisis mendalam terhadap subyek yang diukur. Karena kesulitan ini maka hasil pengukurannya juga sangat tergantung keahlian pelaku pengukurannya sendiri sehingga menambah subyektifitas hasil pengukuran FPA [3,2]. Komponen yang perlu dihitung dalam FPA adalah komponen fungsi transaksional dan fungsi data.

Beberapa penelitian mengenai penerapan metode Function Point (FP) telah dilakukan, misalnya penelitian yang menjelaskan contoh perhitungan FP pada produk software [4] dan pengukuran FP dengan menggunakan pendekatan CBR [5]. Penelitian untuk menerapkan FP pada proyek software yang berorientasi object juga telah banyak dilakukan, dengan konsep pendekatan pemetaan komponen FP yang berbeda [6, 7, 8]. FP juga terbukti dapat diterapkan pada tahap desain yang merupakan tahap awal pengembangan proyek, yaitu dengan menggunakan UML sebagai salah satu model desain [9,10,11,12,13]. Analisis FP juga terbukti lebih mudah diadopsi untuk penerapan FUR (*Functional User Requirements)* menggunakan desain UML dibandingkan dengan metode Cosmic-Full FP (CFFP) [14].

Kelebihan pengukuran FP pada desain UML adalah bahwa desain UML dapat mengakomodasi berbagai model pengembangan perangkat lunak terutama perangkat lunak berbasis objek dan beberapa diagram UML juga menyediakan antar muka yang mudah dipahami, sehingga pengukuran FP menggunakan desain UML dapat memberi pengertian yang lebih baik pada pihak-pihak yang berkepentingan dalam pengerjaan proyek software.

Use case diagram terbukti dapat mewakili komponen fungsi transaksi dalam [7,11,14] sedangkan Class diagram mewakili komponen fungsi data yang merujuk pada file-file logikal yang digunakan dalam realisasi use case [9,10,11,13].

Mempertimbangkan kepentingan end user dan tim pengembang perangkat lunak dalam hal perolehan ukuran FP maka diperlukan sistem sistem yang dapat melakukan perhitungan FP berdasarkan kelengkapan dokumentasi diagram UML yang dimilikinya.

2. METODE PENELITIAN

Penelitian ini bertujuan untuk membuat sistem yang dapat melakukan perhitungan FP berdasarkan dokumentasi yang dimiliki proyek perangkat lunak dengan mengotomatisasi beberapa analisa manual dalam perhitungan tersebut. Perhitungan FP pada penenlitian ini mengacu pada metode perhitungan FP dari IFPUG CPM 4.3.1 dimana besaran fungsionalitas (dalam satuan FP) adalah jumlah komponen fungsi data dan fungsi transaksi (hanya uFP/*unadjusted FP* tanpa perhitungan VAF/ *Value Adjusment Factor).* Proses otomasi analisis FP yang dilakukan berdasarkan dokumentasi diagram UML yang dimiliki dapat dilihat pada Tabel 1:

Tabel 1 Proses otomasi oleh sistem pada analisis FP

<table>
<tr><td colspan="2" rowspan="2">Proses perhitungan FP</td><td colspan="3">Basis Perhitungan</td></tr>
<tr><td>Use case diagram</td><td>Use case dan Class diagram</td><td>Asosiasi antara use case dan class</td></tr>
<tr><td colspan="2">Identifikasi user</td><td>dari actor atau package</td><td>dari actor atau package</td><td>dari actor atau package</td></tr>
<tr><td colspan="2">Identifikasi batasan aplikasi</td><td>dari actor atau package</td><td>dari actor atau package</td><td>dari actor atau package</td></tr>
<tr><td colspan="2">Identifikasi Fungsi Transaksional (TF)</td><td>dari use case</td><td>Dari use case</td><td>dari use case</td></tr>
<tr><td colspan="2">Penentuan Tipe TF</td><td>Oleh user</td><td>Oleh user</td><td>dari cara mengacu file logical</td></tr>
<tr><td rowspan="2">Kompleksitas TF</td><td>DET</td><td rowspan="2">TF diberi kompleksitas “average”</td><td rowspan="2">TF diberi kompleksitas “average”</td><td>dari atribut class dan DET tambahan dari user</td></tr>
<tr><td>FTR</td><td>dari hubungan antara use case dan class</td></tr>
<tr><td colspan="2">Identifikasi Fungsi Data (DF)</td><td rowspan="4">Tidak diketahui</td><td>Dari Class, dipilih oleh user</td><td>Dari Class</td></tr>
<tr><td colspan="2">Penentuan tipe DF</td><td>Oleh user</td><td>Dari cara DF diacu oleh TF</td></tr>
<tr><td rowspan="2">Kompleksitas (DF)</td><td>DET</td><td>Dari atribut class</td><td>Dari atribut class</td></tr>
<tr><td>RET</td><td>Dari multiplicity dan dependency antar Class</td><td>Dari multiplicity dan dependency antar Class</td></tr>
<tr><td colspan="2">Perhitungan uFP</td><td>uFP = FPTF</td><td>uFP = FPTF + FPDF</td><td>uFP = FPTF + FPDF</td></tr>
</table>

2.1 Perhitungan Sumber Daya Menggunakan FP

Function Point dapat digunakan untuk penentuan besar sumber daya pada sebuah proyek *software*. Sumber daya yang dimaksud antara lain menyangkut waktu dan usaha yang diperlukan untuk penyelesaian sebuah proyek *software.* Berikut ini merupakan rumus perhitungan sumber daya yang diambil dari ISBSG [9] dengan hanya menggunakan ukuran fungsionalitas (FP) untuk proyek pengembangan software tanpa melihat platform dan bahasa pemrograman yang digunakan untuk mengembangkan software tersebut.

a. PWE (Project Work Effort).

PWE adalah besaran yang menunjukkan banyaknya waktu kerja yang diperlukan untuk pembangunan proyek software. Rumus penentuan PWE adalah sebagai berikut:

$$PWE = C \times Size^{E1} \quad (1)$$

Dimana:

PWE = Project Effort yang ternormalisasi untuk team pengembang (dalam satuan Hours).
Size = software size (FP).
C = konstanta (23.25).
E1 = konstanta (0,8 14).
Nilai Median MRE untuk rumus (1) adalah 0,55.

b. Project Duration.

Project duration menujukkan banyaknya waktu yang diperlukan untuk penyelesaian proyek software. Projectt duration dihitung menggunakan rumus 2.

$$Duration = C \times Size^{E1} \quad (2)$$

Dimana:

Duration = waktu aktif pengerj aan proyek software (months).
Size = software size (FP).
C = Konstanta (0,543).
E1 = konstanta (0,408).
Nilai Median MRE untuk rumus (2) adalah 0,41.

c. Speed of Delivery.

Speed of delivery menunjukkan kecepatan pengerjaan proyek software oleh seluruh tim developer.

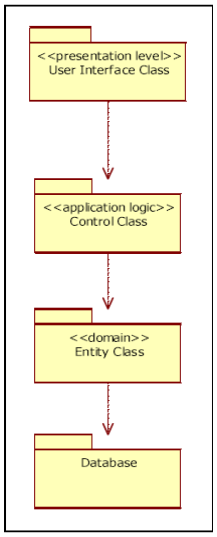
$$Speed\ for\ Project = C \times Size^{E1} \quad (3)$$

Speed for Project = kecepatan pengerjaan project untuk keseluruhan team (FP/month yang dipakai).
Size = software size (FP).
C = konstanta (1,842).
E1 = konstanta (0,592).
Nilai Median MRE untuk rumus (3) adalah 0,40.

2.2 Arsitektur Sistem

Penggambaran sistem analisis FP berdasarkan lapisan abstraksinya mengacu pada [16], dapat dilihat pada Gambar 1.

Level abstraksi pada Gambar 1. menjelaskan bahwa sistem dibagi menj adi 4 level utama yaitu level *presentation*, level *application logic*, level *domain* dan level *database.* Level presentasi berisi kumpulan *user interface* yang memungkinkan *user* mengakses sistem analisis FP. Level *application logic* berisi kumpulan *control class* yang bertugas untuk mengatur aksi-aksi yang harus dijalankan oleh sistem sesuai dengan perintah yang diminta *user.* Level *domain* merupakan kumpulan *entity class* yang digunakan untuk menghubungkan sistem dengan databasenya. Level Database merupakan bagian dimana file-file database disimpan dan dimanipulasi oleh sistem melalui *operasi* dan *method* yang terdapat dalam *entity class.*



Gambar 1 Arsitektur sistem analisis FP

2.3 Implementasi

Sistem analisis FP dibuat dengan mengimplementasikan modul-modul utama seperti:

2.3.1 Pembuatan project perhitungan FP baru.

Sistem akan membuat proyek FP baru dengan menyimpan nama proyek yang mewakili sebuah proyek perangkat lunak.

2.3.2 Perhitungan FP

Perhitungan FP dalam sebuah proyek FP memerlukan input berupa dokumen XMI hasil export diagram UML. Pada saat user memasukkan dokumen XMI ke dalam sistem analisis FP, sistem analisis FP secara otomatis menjalankan proses *parsing* file XMI menjadi komponen-komponen FP. Setelah semua komponen FP diverifikasi oleh user, hasil perhitungan FP dapat dilihat setelah sistem menjalankan proses *generate FP*. Proses *generate FP* merupakan proses perhitungan tiap komponen fungsi transaksi dan fungsi data yang dibuat sesuai dengan aturan perhitungan FP dari IFPUG [3].

2.3.3 Pembuatan laporan perhitungan FP

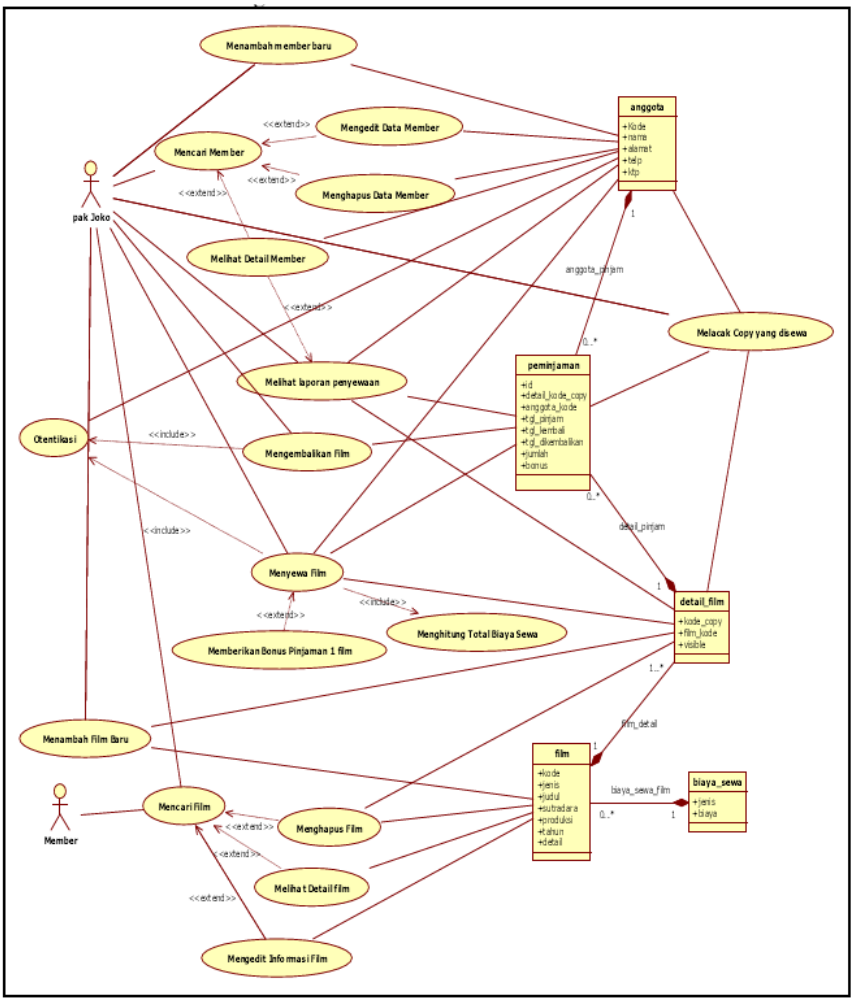
Dokumentasi perhitungan FP untuk proyek perangkat lunak dengan dokumentasi tertentu dapat dilihat setelahsistem menjalankan perintah untuk membuat laporan FP. Laporan tersebut memuat informasi mengenai komponen penting perhitungan FP yaitu nama proyek FP, diagram UML yang digunakan sebagai dasar perhitungan FP, nama file XMI yang dianalisa, waktu dilakukannya analisis FP, jumlah uFP (yang merupakan jumlah dari fungsi data dan fumgsi transaksi), dan perkiraan sumber daya proyek perangkat lunak yang dianalisa.

3. HASIL DAN PEMBAHASAN

Bagian ini dibagi menjadi dua bagian utama yaitu perhitungan FP menggunakan studi kasus dan bagian validasi. Pada bagian pertama dilakukan perhitungan Function Point pada sebuah perangkat lunak sesuai dokumentasi yang dimilikinya (studi kasus) menggunakan sistem analisis FP dan bagian kedua adalah melakukan perbandingan hasil perhitungan FP menggunakan sistem analisis FP dengan hasil perhitungan yang dilakukan oleh IFPUG untuk memeriksa validitas perhitungan.

Studi kasus yang dianalisa dan dihitung menggunakan sistem analisis FP adalah sistem Video Rental sederhana. Analisis FP dilakukan pada use case dan class diagram yang dimiliki

oleh sistem Video Rental. Gambar 2 merupakan contoh salah satu dokumentasi Video rental yaitu berupa diagram yang menggambarkan hubungan antara use case dan class-class yang digunakan untuk realisasi use case tersebut.



Gambar 2 Asosiasi antara use case dan class diagram sistem video rental

FUNCTION POINT ANALYSIS
Studi Kasus Rental Pak Joko

Function Point Detail For : pak Joko System (82FP)

Transactional Function (61FP)

No	Function	Type	DET	FTR	Complexity	uFP Size
1	Menambah member baru	EI	5	1	Low	3
2	Menyewa Film	EI	23	3	High	6
3	Mencari Film	EQ	10	1	Low	3
4	Menambah Film Baru	EI	10	1	Low	3
5	Mencari Member	EQ	5	1	Low	3
6	Melacak Copy yang disewa	EQ	23	3	High	6
7	Melihat laporan penyewaan	EQ	23	3	High	7
8	Mengembalikan Film	EI	23	3	High	6
9	Memberikan Bonus Pinjaman 1 film	EI	18	2	High	6
10	Menghapus Film	EI	10	1	Low	3
11	Mengedit Informasi Film	EI	10	1	Low	3
12	Menghapus Data Member	EI	5	1	Low	3
13	Melihat Detail Member	EQ	5	1	Low	3
14	Mengedit Data Member	EI	5	1	Low	3
15	Melihat Detail film	EQ	10	1	Low	3
	Total					61

Data Function (21FP)

No	Function	Type	DET	RET	Complexity	uFP Size
1	peminjaman	ILF	8	1	Low	7
2	anggota	ILF	5	1	Low	7
3	film - detail_film	ILF	10	2	Low	7
	Total					21

Gambar 3 Hasil perhitungan FP menggunakan hubungan antara use case dan class diagram untuk Actor Pak Joko.

FUNCTION POINT ANALYSIS
Studi Kasus Rental Pak Joko

Function Point Detail For : Member System (8FP)

Transactional Function (3FP)

No	Function	Type	DET	FTR	Complexity	uFP Size
1	Mencari Film	EQ	7	1	Low	3
		Total				3

Data Function (5FP)

No	Function	Type	DET	RET	Complexity	uFP Size
1	film - detail_film	EIF	7	1	Low	5
		Total				5

Gambar 4 Hasil perhitungan FP menggunakan hubungan antara use case dan class diagram untuk actor Member

PROJECT ESTIMATION
Using ISBSG Regression Equation
Estimated from software size only in function points (IFPUG)
Project Size : 90 function points

Project Work Effort (New)	906	hours
Project Work Effort (New, PC, 4GL)	589	hours
Project Duration (New)	3.4	months
Project Duration (New, PC, 4GL)	2.6	months
Speed of Delivery (New)	26.4	FP/month
Speed of Delivery (New, PC, 4GL)	35.2	FP/month

Gambar 5 Hasil perhitungan perkiraan sumber daya proyek yang berukuran 90 FP

Hasil perhitungan FP yang didapatkan berdasarkan dokumentasi pada Gambar 2 adalah sebesar 90 FP, yaitu 82 FP untuk aplikasi admin dan 8 FP untuk aplikasi customer. Pada Gambar 3 dan Gambar 4 dapat terlihat bahwa kompleksitas fungsi transaksi untuk kedua aplikasi (Pak Joko dan Member) adalah beragam yaitu *high* dan *low*, hal ini karena menyesuaikan jumlah FTR dan DET yang dimiliki oleh fungsi transaksi tersebut. Sedangkan kompleksitas untuk fungsi data yang teridentifikasi pada kedua aplikasi mempunyai kompleksitas low sesuai dengan DET dan RET yang dimilikinya.

Besaran 90 FP untuk project rental kemudian digunakan untuk memperkirakan sumber daya yang dihitung menggunakan persamaan (1), (2) dan (3). Hasil perhitungan suber daya yang dimaksud dapat dilihat pada Gambar 5.

Besarnya *Project Work Effort (PWE)* menunjukkan besarnya usaha yang diperlukan untuk menyelesaikan pengembangan proyek software, ditunjukkan dalam besaran *hours* (jam kerja). *Project Work Effort (New)* merupakan besarnya usaha yang diperlukan untuk mengembangakan proyek baru tanpa memperhatikan platform dan bahasa pemrograman yang digunakan, sedangkan *Project Work Effort (New, PC, 4GL)* merupakan besarnya usaha yang diperlukan untuk mengembangakan proyek baru menggunakan platform PC dan bahasa pemrograman generasi keempat.

Gambar 5 menunjukkan bahwa pengembangan proyek rental secara umum memerlukan 906 jam kerja (*hours)* dan jika dikerjakan berbasis PC dan menggunakan bahasa pemrograman generasi keempat hanya memerlukan sebesar 589 jam kerja (*hours).*

Besaran lain yang dapat diperkirakan adalah *durasi* yaitu banyaknya hari kerja yang diperlukan untuk menyelesaikan pengembangan proyek, ditunjukkan dalam satuan *months* (bulan). Gambar 5 menunjukkan bahwa untuk mengerjakan proyek dengan ukuran 90 FP diperlukan waktu kurang lebih 3,4 bulan (hari kerja) dan jika proyek tersebut dikerjakan berbasis PC dan menggunakan bahasa pemrograman generasi keempat akan dapat dikerjakan dalam waktu 2,6 bulan.

Speed of delivery merupakan besaran yang menunjukkan rata-rata kecepatan yang diperlukan oleh seluruh tim pengembang untuk menyelesaikan proyek software. *Speed of delivery* dinyatakan dalam satuan FP/month, dimana *month* merupakan keseluruhan waktu yaitu hari kerja ditambah dengan hari libur. Gambar 5 menunjukkan bahwa untuk mengerjakan proyek dengan ukuran 90 FP, kecepatan penyelesaian proyek secara umum kurang lebih sebesar 26,4 FP/month

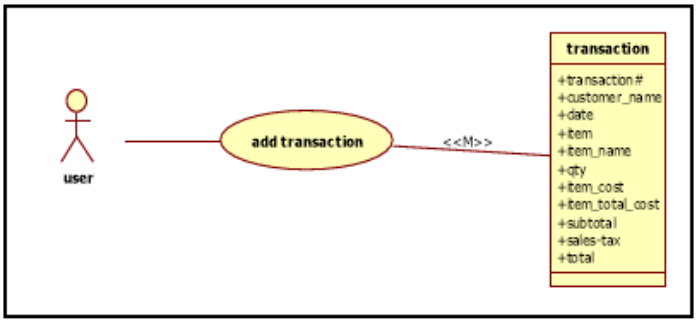
dan jika proyek tersebut dikembangkan berbasis PC dan menggunakan bahasa pemrograman generasi keempat kecepatannya kurang lebih sebesar 35,2 FP/month.

Berikutnya, sistem analisis FP juga diterapkan untuk contoh-contoh perhitungan FP yang terdapat dalam IFPUG CPM 4.3.1 untuk melihat tingkat keakuratan perhitungan dari sistem. Dokumentasi yang terdapat pada contoh perhitungan FP dibuat kembali dalam bentuk desain UML kemudian dijadikan sebagai input perhitungan pada sistem. Hasil perhitungan yang dilakukan dirangkum pada Tabel 2.

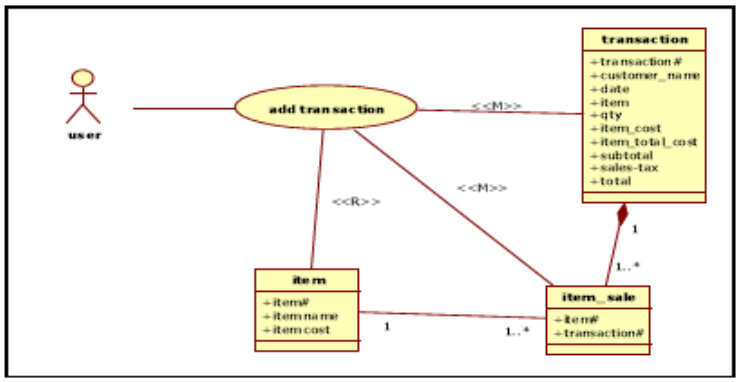
Tabel 2 Hasil validasi perhitungan FP

No	Studi Kasus (diambil dan 1FPUG CPM 4.3.1)	Validasi Komponen yang dikenali							
		Jenis Fungsi		Jumlah File Reference (FTR/RET)		Jumlah DET		Jumlah FP	
		Sistem	IFPUG	Sistem	IFPUG	Sistem	IFPUG	Sistem	IFPUG
1	SK 1-9	ILF	ILF	2	2	9	8	7	7
2	SKi-iS	ILF	ILF	1	1	4	4	7	7
3	SK 1-21	ILF	ILF	1	1	7	6 dan 3	7	7
4	SK 1-29	ILF	ILF	2	1	7	6 dan 3	7	7
5	SK 1-39	ElF	ELF	1	1	6	6	5	5
6	SK 1-43	ElF	ELF	1	1	2	2	5	5
7	SK 1-62	ElF	ELF	1	1	6	2	5	5
8	SK 2-73	El	El	1	1	5	6	3	3
9	SK2-77	El	El	3	3	12	11	6	6
10	SK2-85	El	El	2	1	15	11	4	3
11	SK2-91	El	El	3	3	21	13	6	6
12	SK2-105	EO	EO	4	4	15	5	7	5
13	SK2-108	EO	EO	3	3	13	8	5	5
14	SK2-131	EQ	EQ	1	1	11	6	3	3
15	SK2-145	EQ	EQ	3	3	15	10	4	4
16	SK2-159	EQ	EQ	1	1	4	4	3	3

Tabel 2 menunjukkan bahwa sistem yang dibangun dapat mengidentifikasi komponen FP dan dapat menghasilkan jumlah FP dengan benar. Perbedaan pada hasil perhitungan, misalnya pada SK 2-85, terjadi karena perbedaan presepsi mengenai penggambaran fungsi data antara analis. Penulis berasumsi bahwa setidaknya diperlukan 2 file untuk realisasi use case *add transaction*, yaitu file/class transaction dan file/class item. Pada *user requirements* dikatakan bahwa ketika item barang dimasukkan maka harga *(item cost)* dan barang tersebut akan muncul secara otomatis. mi berarti bahwa ketika user mulai me-*maintain* data item ke dalam file transaction maka aplikasi akan merujuk pada file item untuk mengambil harganya. Selain itu, hubungan antara class transaction dan class item merupakan relasi *many to many* dan menghasilkan class item_sale. Gambar 6 dan Gambar 7 merupakan contoh 2 diagram berbeda yang dihasilkan berdasarkan sebuah user *requfrement* yang sama (untuk SK 2-85).



Gambar 6 Use Case dan Class Diagram asumsi IFPUG untuk SK 2-85



Gambar 7 Use Case dan Class Diagram asumsi penulis untuk SK 2-85

4. KESIMPULAN

Implementasi sistem analisis FP yang dilakukan pada beberapa studi kasus menghasilkan beberapa kesimpulan sebagai berikut:

1) Hasil validasi perhitungan FP menunjukkan bahwa semakin lengkap dokumen UML yang dimiliki oleh perangkat lunak maka hasil perhitungan FP yang didapatkan akan semakin akurat.
2) Hasil validasi perhitungan FP yang telah dilakukan menunjukkan bahwa dokumentasi diagram UML dengan syarat tertentu bisa dilakukan perhitungan FP secara otomatis dan cepat dan dengan hasil yang akurat.
   Adapun syarat yang harus dipenuhi adalah:
   a. Use case yang digambar harus memenuhi syarat sebagai suatu proses dasar. Use case yang merupakan proses dasar harus terhubung pada actor tertentu, atau *extend* dari use case yang terhubung kepada actor.
   b. Penggambaran class diagram harus lengkap meliputi atribut, jenis dependensi dan multiplicity hubungan antar class.
   c. Diagram asosiasi antara use case dan class diagram harus dilengkapi dengan stereotype yang menggambarkan tipe atau cara acuannya (*read only*, *read and calculating*, atau *maintain).*
3) Perbedaan hasil perhitungan FP antara produk dan desain perangkat lunak bisa terjadi karena perbedaan jumlah DET. Pada produk perangkat lunak, DET yang dilibatkan bisa diverifikasi dengan lebih tepat karena dapat diketahui secara pasti data input atau data output yang terlibat untuk pemenuhan fungsionalitas tertentu.

5. SARAN

Saran yang diberikan dalam upaya pengembangan sistem analisis FP yang dibuat adalah:

1) Sistem analisis FP yang dibuat ini dapat dikembangkan lagi dalam hal intepretasi diagram UML dengan cara penggambaran diagram use case dan class yang bermacam-macam, versi dokumen XMI yang dihasilkan, serta *tools* yang digunakan.
2) Sistem analisis FP yang dibuat masih memerlukan verifikasi DET oleh user secara manual sehingga pada penelitian lanjutan dapat dilakukan pengembangan dalam hal identifikasi DET secara langsung oleh sistem.

DAFTAR PUSTAKA

[1] Bundschuh M. dan Dekkers C., 2008, *The IT Measurement Compendium : Estimating and Benchmarking Success with Functional Size Measurement*, Springer-Verlag Berlin Heidelberg, e-ISBN 978-3-540-68188-5.

[2] Longstreet, D., 2002, Function Point Analysis Training Course, *http://www.softwaremetrics.com/freemanual.htm,* diakses pada 17 Juli 2010.

[3] International Function Point Users Group (IFPUG), 2010, *Function Point Counting Practices Manual release 4.3.1*., ISBN: 978-0-9753783-4-2.

[4] Kusrini dan Iskandar , MD., 2006, *Pengukuran Volume Software berdasarkan Kompleksitasnya dengan Metode Function Point*, DASI.

[5] Mahar ,K. dan El-Deraa, A., 2006, Software Project Estimation Model Using Function Point Analysis With Cbr Support, *IADIS*, ISBN: 972- 8924-09-7.

[6] Chamundeswari A. dan Chitra B., 2010, an Extended Function Point Approach for Size Estimation of Object-Oriented Software, dalam proceeding *3rd India Software Engineering Conference (ISEC-ESE)*, 25-27 Pebruari, India.

[7] Fetcke T., Abran A., dan Nguyen T.H., 1997, Mapping the OO-Jacobson Approach to Function Point Analysis, dalam proceeding *Spring Conference IFPUG, pp. 134-142.*

[8] Giachetti G., Marin B., Condori-Fernandez N., dan Molina J.C., 2007, Updating OO Method Function Points, dalam proceeding *6th Conference on the Quality of Information and Communications Technology (QUATIC 2007)*, 12-14 September, Lisbon, Portugal, ISBN : 0-7695-2948-8.

[9] Cantone G., Pace D., dan Calavaro G., 2004, Applying Function Point to Unified Modeling Language : Conversion Model and pilot Study, dalam proceeding *10th International Symposium on Software Metrics (METRICS '04).*

[10] Del Bianco, V., Gentile, C. dan Lavazza, L., 2008, An Evaluation of Function Point Counting Based on Measurement-Oriented Models, dalam Proceeding *Evaluation and Assessment in Software Engineering*, June 26-27, Bari.

[11] Iorio T., 2004, IFPUG Function Point Analysis in a UML Framework, dalam proceeding *SMEF04*, Roma, Italia.

[12] The Netherlands Software Metrics Users Association (NESMA), 2008, *FPA Applied to UML/Use Cases Versi 1.0*, ISBN : 978-90-76258- 24-9.

[13] Uemura, T., Kusumoto S., dan Inoue, K., 1999, Function Point Measurement Tool For Uml Design Specification, Proc. Of *The Metrics '99*, Boca Raton, Florida.

[14] Van den Berg K., Dekkers T., dan Oudshoorn R., 2005, *Functional Size Measurement Applied to UML-based User Requirements*, 2nd Software Measurement European Forum, 16-18 Maret, Roma, Italia.

[15] International Software Benchmarking Standards Group (ISBSG), 2011, *Practical Software Project Estimation: A Toolkit for Estimating Software Development Effort & Duration*, The McGraw-Hill Companies, Inc., ISBN: 978-0-07-171792-2.

[16] Bennett S., McRobb S., dan Farmer R., 2006, *Object-Oriented System Analysis and Design Using UML*, 3rd Edition, The McGraw-Hill Companies, Inc.