

ESTIMASI PENGELOLAAN SUATU PROYEK DALAM PENGEMBANGAN PERANGKAT LUNAK MENGUNAKAN ANALISA *FUNCTION POINT*

Fityan Aula Juyuspan¹⁾, Anita Hidayati²⁾

^{1,2}Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta

Jl. Prof. Dr. G.A.Siwabessy, Kampus UI, Depok

E-mail : ¹fityanaula@gmail.com, ²anita.hidayati@tik.pnj.ac.id

ABSTRAK

Upaya dalam pengembangan perangkat lunak merupakan hal yang penting dalam industri perangkat lunak. Pada *paper* ini, klasifikasi sistem perangkat lunak untuk menentukan estimasi usaha yang akan dilakukan dengan cara pengukuran berbasis *Function Point* dengan metode *Application FP Count* berbasis perhitungan *DET*, *RET*, dan *FTR*. Tujuan dari *paper* ini adalah untuk digunakan sebagai perkiraan pengukuran waktu dan biaya dalam pengerjaan proyek sehingga akan diperoleh hasil yang akurat dan dapat digunakan sebagai perbandingan dengan proyek baru yang serupa. Hal ini membuat upaya pengembang menjadi lebih mudah dalam pengelolaan suatu proyek dengan memperhitungkan *ILF*, *EIF*, *EI*, *EO*, dan *EQ* dari sebuah proyek lama dan kemudian didapatkan nilai *FP* yang digunakan untuk estimasi proyek baru. Untuk studi kasus digunakan sebuah software aplikasi sekolah (*Teacher Manager*). Dalam estimasi, diasumsikan proyek dikerjakan 200 jam/92.4FPs (2.16 jam/FP) dan dikerjakan oleh 2 pengembang. Pengembang mampu mengerjakan 2.5 jam/FP sehingga proyek dikerjakan selama 231 jam. *RUSA* merupakan proyek yang didalamnya terdapat fitur *Teacher Manager* dengan *FP* 115 dan dikerjakan oleh 2 orang. Modul ini paling lama akan dikerjakan selama 200 jam/115 FPs (1.74 jam/FP). Jika pengembang hanya mampu mengerjakan 2.5 jam/FP maka proyek ini akan dikerjakan selama 287.5 jam. Biaya pengerjaan *RUSA* juga dapat diperkirakan dengan melihat tabel produktivitas.

Kata kunci : Function Point, Estimasi Proyek, Produktivitas, Application FP Count.

ABSTRACT

Efforts in software development are important in the software industry. In this paper, the classification of software systems to determine estimates of the business is to be conducted in a manner based measurement method Application Function Point Count FP calculations which based upon DET, RET, and FTR. The purpose of this paper is to be used as an approximate measurement of time and cost in a project that will obtain accurate results which can be employed as a comparison with similar new projects. This result makes the efforts of developers become easier in the management of a project taking into account the ILF, EIF, EI, EO, and EQ of a long project and then obtained FP value used to estimate the new project. For the case study used a software application (Teacher Manager). In the estimation, it is assumed the project worked 200 hours / 92.4FPs (2.16 hours / FP) and carried out by two developers. Developers can do 2.5 hours / FP so that projects are carried out for 231 hours. RUSA is a project in which there are features of the FP 115 Teacher Manager and done by two people. This module is the longest to be done during the 200 hours / 115 fps (1.74 hours / FP). If the developer is only able to work 2.5 hours / FP, then this project will be done for 287.5 hours. RUSA construction costs can also be estimated by looking at the table of productivity.

Keywords: Function Point, Project Estimation, Productivity, Application FP Count.

PENDAHULUAN

Perangkat lunak yang dikembangkan sekarang ini jauh lebih kompleks dibanding dengan tahun 1970-an dimana perangkat lunak masih berjalan menggunakan *processor* tunggal. Perangkat lunak sekarang memiliki *Graphical User Interface* (GUI) dan arsitektur *client-server*. 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* [1]. Jadi, prediksi pengembangan perangkat lunak yang handal dan akurat merupakan masalah penting dalam pengelolaan proyek [2]. Teknik estimasi merupakan salah satu cara yang efektif dalam penentuan jadwal dan biaya dalam sebuah proyek [3]. Sebuah prediksi yang akurat dapat bermanfaat pada perencanaan suatu proyek sehingga pengelolaan menjadi lebih baik, dan menjamin kualitas pengembangan perangkat lunak. Jika tidak memiliki data yang cukup jelas, pengembang akan menghabiskan banyak upaya dalam mengumpulkan data pengembangan proyek.

Metode analisa *Function Point*, dapat digunakan untuk estimasi upaya penentuan jadwal, biaya, dan jumlah pegawai dalam sebuah proyek pengembangan perangkat lunak. Hal tersebut dilakukan dengan membagi sistem ke dalam beberapa komponen sehingga dapat dianalisa dan dimengerti. Masalah yang sulit mungkin akan tampak lebih sederhana setelah dipecah menjadi bagian yang lebih kecil kemudian dikelompokkan.

Dengan melakukan perhitungan *Function Point* pada proyek yang pernah dikerjakan sebelumnya, akan memudahkan pengembang dalam memprediksi penjadwalan dan biaya pada proyek serupa sehingga meminimalisir upaya dalam pengumpulan data pada skala besar. Meskipun penilaian ahli tetap banyak

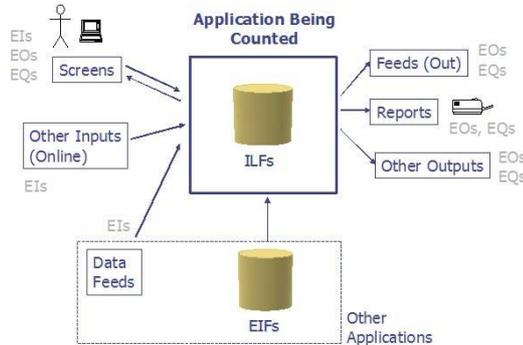
digunakan, ada pula yang berminat dalam penerapan statistik dan teknik *machine learning* untuk memprediksi usaha proyek perangkat lunak [4]. Biaya perangkat, biaya perjalanan dan pelatihan, dan biaya usaha adalah tiga komponen utama dari biaya usaha dominan [5]. Meskipun kompleksitas estimasi perangkat lunak kadang-kadang hanya dilakukan oleh seorang ahli estimasi itu sendiri. Dalam beberapa dekade terakhir ini, *Function Point* telah dikembangkan untuk estimasi proyek perangkat lunak [6].

Function Point (FP)

Analisa *Function Point* dikembangkan oleh Allan J. Albrecht pada tahun 1970-an. FP dapat digunakan untuk mengestimasi kompleksitas dari perangkat lunak [7], pengukuran waktu, dan biaya selama pengerjaan perangkat lunak. Jumlah total FP tergantung pada perhitungan yang berbeda (dalam hal format atau pengolahan logika) jenis. Berikut 5 jenis komponen tersebut yang terdiri dari 2 *data functions* dan 3 *transaction types*:

1. *Internal Logical File (ILF)*, merupakan logika data yang disimpan dan di maintained pada aplikasi *boundary*.
2. *External Interface File (EIF)*, merupakan pengelompokan logika data yang berada di luar aplikasi tetapi menyediakan data yang mungkin dapat digunakan pada aplikasi. EIF sama dengan ILF hanya saja ILF yang berada pada aplikasi lain disebut EIF.
3. *External Inputs (EI)*, merupakan proses data EI atau kontrol informasi yang datang dari luar *application boundary*. EI dapat berupa aktifitas unit terkecil dari sebuah aplikasi. Pada implementasi EI berupa *field-field* pada aplikasi.
4. *External Outputs (EO)*, merupakan proses dasar dimana data melintasi *boundary* dari dalam aplikasi ke luar aplikasi. EO dapat memperbaharui ILF. Pada implementasi EO bisa berupa laporan yang bisa dicetak.
5. *External Inquiry (EQ)*, merupakan sebuah proses dasar yang terdiri dari

gabungan *input-output* yang menghasilkan data kembali (*retrieve*) atau data yang bisa di-view.



Gambar 1. Lima Komponen Function Point

Dengan memahami kelima komponen pada Gambar 1, pengelola proyek dapat menentukan apakah tingkat kesulitan proyek tersebut berada pada kategori mudah, rata-rata ataupun kompleks. Sehingga pengelola proyek akan dimudahkan dalam pengelolaan proyek yang serupa. Berikut cara menentukan kategori apakah mudah, rata-rata ataupun kompleks. Jika RET yang diketahui maka digunakan Tabel 1, sedangkan jika FTR yang diketahui maka Tabel 2 yang akan digunakan.

Tabel 1. ILF/EIF Metriks Kompleksitas

RET	Data Element Type (DET)		
	1-19	20-50	51+
1	Low	Low	Average
2 to 5	Low	Average	High
6 or more	Average	High	High

Tabel 2. EI/EO/EQ Metriks Kompleksitas

FTR	Data Element Type (DET)		
	1-4	5-15	16+
0 – 1	Low	Low	Average
2	Low	Average	High
3 (if EI) or 4 (if EO, EQ) or more	Average	High	High

Total dari kalkulasi function point disebut *unadjusted function point (UFP) ISO Standard*. UFP adalah hasil perhitungan dari ILF + EIF + EI + EO + EQ. Setelah mengetahui ke lima komponen

tersebut, berikut adalah istilah penting dan definisi yang digunakan dalam menggambarkan kelima komponen tersebut:

- 1) *User Identifiable*: Istilah ini mengacu pada persyaratan yang ditetapkan untuk proses dan / atau kelompok data yang disepakati, dan dipahami oleh pengguna dan pengembang perangkat lunak.
- 2) *Control Information*: Merupakan proses dasar dari aplikasi yang sedang dihitung. Juga secara khusus menyebutkan apa, kapan, atau bagaimana data akan diproses.
- 3) *Elementary Process*: Unit terkecil dari aktivitas yang berarti bagi pengguna. *Elementary Process* harus serba lengkap dan meninggalkan proses bisnis aplikasi yang sedang dihitung dalam keadaan konsisten.
- 4) *Data Element Type (DET)*: Unik, pengguna dikenali, *field* tak-berulang.
- 5) *Record Element Type (RET)*: Pengguna subkelompok yang dapat dikenali dari elemen data dalam suatu ILF atau EIF.

Ada beberapa hal penting yang harus diperkenalkan dan mungkin dilakukan dalam proses analisis *function point* yaitu pengukuran dilihat dari sudut pandang pengguna, *multi-platform*, biaya, menganalisa ulang proyek, dan *function point* bekerja dengan baik pada *use case* [10]. Dengan hal-hal tersebut pengelola proyek dapat mengidentifikasi program dengan baik dan lebih akurat. Dalam kehidupan nyata *function point* juga digunakan sebagai dasar untuk menentukan biaya aplikasi perangkat lunak, membantu menentukan tingkat produktivitas tim pengembangan proyek, dan membantu untuk menentukan nilai ritel aset perangkat lunak.

METODE

Telah dibangun sebuah software aplikasi yang digunakan pada sekolah

untuk mengelola aktifitas-aktifitas guru seperti melihat jadwal, melihat jadwal pelajaran sekolah, dan menginput nilai-nilai murid masing-masing dari guru tersebut kemudian membuat output seperti cetak laporan nilai siswa. Aplikasi ini dibuat dengan menggunakan bahasa pemrograman Java dan memiliki beberapa proses seperti *create, read, update, delete* (CRUD) di beberapa fitur seperti nilai, jadwal, dan siswa itu sendiri.

Sebelum melakukan perhitungan *function point*, pengelola diharapkan memperhatikan langkah-langkah seperti menentukan tipe perhitungan, mengidentifikasi ruang lingkup perhitungan, menghitung UFP, menentukan faktor-faktor *value adjustment*, dan menghitung *value adjusted* itu sendiri [10]. Tabel 3 digunakan untuk menentukan nilai dalam perhitungan function point pada 5 jenis komponen.

Tabel 3. Bobot kompleksitas dengan angka yang sesuai dengan UFP

	<i>ILF</i>	<i>EIF</i>	<i>EI</i>	<i>EO</i>	<i>EQ</i>
Low	7	5	3	4	3
Average	10	7	4	5	4
High	15	10	6	7	6

Langkah Pertama: Menentukan Tipe Pengukuran

Dalam penentuan tipe perhitungan function point terdiri dari tiga jenis yaitu:

1. *Development Project FP Count*, mengukur fungsi yang disediakan untuk pengguna pertama kali.
2. *Enhancement Project FP Count*, mengukur modifikasi dari aplikasi.
3. *Application FP Count*, mengukur fungsi yang disediakan oleh pengguna pada aplikasi yang sudah ada.

Karena aplikasi *Teacher Manager* merupakan aplikasi yang sudah ada maka tipe pengukuran yang digunakan adalah "*Application FP Count*".

Langkah Kedua: Menentukan Ruang Lingkup Pengukuran

Teacher Manager merupakan aplikasi yang penyimpanan datanya

menggunakan database dan masih menggunakan jaringan intranet untuk mengkoneksikan satu *desktop* guru ke database server (*Not Standalone*).

Langkah Ketiga: Menghitung *Unadjusted Function Point*

Walaupun pada tahap pertama dan kedua merupakan tahap yang penting. Pada tahap ketiga ini merupakan perhitungan *data functions* dan *transaction types*.

Menentukan Perhitungan *Internal Logical File (ILF)*: Dalam perhitungan ILF pengelola proyek terlebih dahulu harus mengidentifikasi tabel apa saja yang digunakan pada proyek tersebut. Pada proyek *Teacher Manager*, terdapat tabel:

- Nilai Siswa
- Data Siswa
- Mata Pelajaran
- Kelas
- Jadwal
- Data Guru
- Akun Admin

Dengan mengetahui tabel-tabel yang digunakan untuk perhitungan *Data Element Type (DET)* Pengelola proyek dapat dengan mudah menentukan perhitungan yang akan dilakukan selanjutnya. Berikut adalah rincian penghitungan ILF dari tiap tabel:

Tabel 4. Perhitungan ILF pada Nilai Siswa

<i>Field</i>	<i>Deskripsi</i>	<i>DET</i>
IDMark	Autogenerate System	No
NIS	Foreign Key	Yes
IDCourse	Foreign Key	Yes
Score	Nilai yang diberikan	Yes
Semester	Tingkatan semester siswa	Yes
Total DET		4

Tabel 5. Perhitungan ILF pada Data Siswa

<i>Field</i>	<i>Deskripsi</i>	<i>DET</i>
NIS	Autogenerate System	No
IDClass	Foreign Key	Yes
Name	Nama siswa	Yes
Gender	Jenis kelamin siswa	Yes
DOB	Tanggal Lahir Siswa	Yes
Total DET		4

Tabel 6. Perhitungan ILF pada Mata Pelajaran

Field	Deskripsi	DET
IDCourse	Autogenerate System	No
Course	Mata pelajaran	Yes
Majors	Jurusan (IPA/IPS)	Yes
Total DET		2

Tabel 7. Perhitungan ILF pada Kelas

Field	Deskripsi	DET
IDClass	Autogenerate System	No
Class	Nama kelas	Yes
HomeRoomTeacherNIP	Foreign key	Yes
Majors	Jurusan	Yes
Total DET		3

Tabel 8. Perhitungan ILF pada Jadwal

Field	Deskripsi	DET
IDSchedule	Autogenerate System	No
Class	Foreign key	Yes
IDCourse	Foreign key	Yes
NIP	Foreign Key	Yes
Day	Hari	Yes
Time	Jam berapa	Yes
Total DET		5

Tabel 9. Perhitungan ILF pada Data Guru

Field	Deskripsi	DET
NIP	Nomor Induk Pegawai	Yes
IDCourse	Foreign key	Yes
Password	Password untuk log in	Yes
Name	Nama guru	Yes
Gender	Jenis kelamin guru	Yes
DOB	Tanggal lahir guru	Yes
Total DET		6

Tabel 10. Perhitungan ILF pada Admin

Field	Deskripsi	DET
ID	Autogenerate System	No
Password	Password admin	Yes
Total DET		1

Pada aplikasi ini, tidak ada *one-to-one relationship*. Tetapi ada dua tipe dalam tabel pada perhitungan ini yaitu tabel yang “terpisah” dan tabel yang secara logika “tergabung”. Analogi tersebut disebut “*schema*” pada SQL Server. Pada aplikasi ini, tingkat kesulitan pada tabel dikategorikan “Rendah (*Low*)” karena tidak terlalu kompleks dalam pengerjaannya.

Pada perhitungan tabel yang “terpisah” berarti pengelola proyek menghitung semua tabel untuk mendapatkan nilai FP. Tingkat kesulitan rendah bernilai 7, sedangkan total tabel adalah 7. Jadi, $7 \times 7 = 49$ FPs. Sangatlah berbeda dengan hasil yang secara logika “tergabung”. Tabel yang secara logika tergabung adalah:

- Data Siswa
 - Nilai Siswa
- Data Guru
 - Mata Pelajaran
 - Kelas
 - Jadwal
- Akun Admin

Dengan demikian, perhitungan menjadi $3 \times 7 = 21$ FPs. Jadi, perhitungan ILF yang seharusnya adalah:

Tabel 11. Perhitungan Total ILF

ILF	RET	DET	Complexity	FP
Data Siswa	2	8	Low	7
Data Guru	4	15	Low	7
Akun Admin	1	1	Low	7
Total				21

Menentukan Perhitungan External Interface File (EIF): Pada kasus ini, aplikasi Teacher Manager memiliki eksternal aplikasi yang digunakan untuk monitoring aplikasi yang digunakan oleh guru, yaitu aplikasi untuk admin. External Interface File adalah Internal Logical File pada aplikasi lain. Jadi, total FP EIF yaitu $3 \text{ Low} \times 5 = 15$ FPs.

Menentukan Perhitungan External Input (EI): Tabel 12 adalah daftar External Input pada aplikasi Teacher Manager. Dibawah ini pula terdapat angka untuk DET, FTR, dan kompleksitas pada setiap proses.

Tabel 12. External Inputs (EIs)

Proses	D		FTR Name	F		Kompleksitas	F
	E	T		T	R		
AddHT	4		LoginAdmin, HTManager	2		Low	3
EditHT	1		LoginAdmin, HTManager	2		Low	3
AddTeacher	6		Login, Teacher	2		Average	4
EditTeacher	6		LoginAdmin, Teacher	2		Average	4

DeleteTeacher	6	LoginAdmin, Teacher	2	Average	4
AddStudent	5	LoginAdmin, Student	2	Average	4
EditStudent	5	LoginAdmin, Student	2	Average	4
DeleteStudent	5	LoginAdmin, Student	2	Average	4
LoginAdmin	2	LoginAdmin	1	Low	3
LoginTeacher	2	LoginTeacher	1	Low	3
AddMark	5	LoginTeacher, Mark	2	Average	4
EditMark	5	LoginTeacher, Mark	2	Average	4
AdminSetting	2	LoginTeacher, AdminSetting	2	Low	3
AdminAddSchedule	6	LoginAdmin, AdminAddSchedule	2	Average	4
Setting	2	LoginTeacher, Setting	2	Low	3
Total:					54

Menentukan Perhitungan External Output (EO): Pada Teacher Manager hanya terdapat satu External Output yaitu laporan nilai siswa. Yang terdiri dari 6 DET yaitu:

- a) NIS
- b) Nama Siswa
- c) Mata Pelajaran
- d) Nilai
- e) HomeRoomTeacher
- f) Kelas

Tabel13. External Output (EO)

Proses	DET	FTR	Kompleksitas	FP
Laporan Nilai Siswa	6	2	Average	4
Total:				4

Menentukan Perhitungan External Inquiry (EQ): Tabel 14 merupakan proses EQ yang ada pada aplikasi Teacher Manager.

Tabel 14. External Inquiries (EQs)

Proses	DET	FTR	Kompleksitas	FP
AdminViewStudent	5	2	Average	4
AdminViewTeacher	5	2	Average	4
AdminViewHT	3	2	Low	3
ViewMarks	6	2	Average	4
ViewSchedule	5	2	Average	4
Total:				19

Menentukan Unadjusted Function Point: Dengan mengetahui perhitungan ILF, EIF, EI, EO, EQ maka sekarang pengelola proyek dapat menghitung UFP.

Tabel 15. Unadjusted Function Point

Tipe Fungsi	Kompleksitas	Total Kompleksitas		Total
ILF	3 Low	x 7	21	21
	0 Average	x 10	0	
	0 High	x 15	0	
EIF	3 Low	x 5	15	15
	0 Average	x 7	0	
	0 High	x 10	0	
EI	6 Low	x 4	24	60
	9 Average	x 4	36	
	0 High	x 6	0	
EO	0 Low	x 4	0	5
	1 Average	x 5	5	
	0 High	x 7	0	
EQ	1 Low	x 3	3	19
	4 Average	x 4	16	
	0 High	x 6	0	
Unadjusted Function Point				120

Menentukan Nilai VAF (Value Adjustment Factor): VAF adalah perhitungan yang berbasis 14 General System Characteristics (GSCs) yang mengukur fungsi umum dari aplikasi yang akan di hitung [8]. VAF terdiri dari 14 GSCs.

Dari ke 14 GSCs pengelola proyek harus memperkirakan nilai dari masing-masing karakteristik, nilai terdiri dari 0 (low) sampai 5 (high). Pada kasus proyek Teacher Manager, penulis mendefinisikan a = 2, b = e = 0, f = h = 5, i = j = 0, k = m = 3, n = 2. Total dari penaksiran dari tiap GSCs adalah TDI (Total Degree Influence) yaitu 12. VAF memiliki rumus (1).

$$VAF = (TDI * 0.01) + 0.65 \quad (1)$$

Pada hasilnya, VAF memiliki nilai yang berkisar antara 0.65 – 1.35. Dan TDI memiliki nilai yang berkisar antara 0 – 70. Jadi, nilai VAF dari proyek Teacher Manager adalah 0.77.

Menentukan Nilai Function Point: FP merupakan nilai yang didapat dari nilai

VAF * UFP. UFP dari proyek *Teacher Manager* adalah 120 dan VAF 0.77. Jadi, FP dari proyek ini adalah 92.4.

HASIL DAN PEMBAHASAN

Estimasi Waktu Pengerjaan

Dalam estimasi ini, diasumsikan proyek ini dikerjakan paling lama 200 jam/92.4FPs (2.16 jam/FP) dan dikerjakan oleh 2 pengembang. Masing-masing pengembang mengerjakan 100 jam (200/2). Dalam satu minggu, pengembang memiliki waktu sebanyak 30 jam untuk mengerjakan proyek *Teacher Manager*. Jadi, 100 jam per orang / 30 = 3.3 minggu / 46.2 FPs. Jika diketahui pengembang hanya mampu mengerjakan selama 2.5 jam/FP maka proyek paling lama akan dikerjakan selama 231 jam.

Dengan mengetahui jumlah function point, maka dapat di-estimasi proyek yang serupa yang memiliki konten yang hampir mirip dengan yang pernah dikerjakan. Sebagai contoh, proyek RUSA (RUjak System for Academy) merupakan proyek yang akan dikerjakan oleh pengelola. Walaupun tidak semua perhitungan pada *Teacher Manager* sama dengan beberapa modul yang ada pada RUSA akan tetapi pengelola dapat mengestimasi waktu pengerjaan suatu modul pada RUSA berdasarkan pengalaman pengelola dalam pengerjaan proyek *Teacher Manager*. Sebagai

contoh, pada proyek RUSA terdapat fitur *Teacher Manager*, sehingga pengelola proyek dapat menggunakan tipe pengukuran "*Application FP Count*". Jadi, tanpa menghitung lagi pengelola yang memiliki pengalaman banyak atau memiliki data yang banyak tentang perhitungan *Function Point* yang pernah dikerjakan, akan lebih mudah mendapatkan hasil yang akurat. Hanya saja mungkin ada beberapa data dan input, output bahkan inquiry yang sedikit berbeda. Tetapi, untuk estimasi tidaklah terlalu jauh. RUSA memiliki modul *Teacher Manager* dengan FP 115 dan akan dikerjakan oleh 2 orang pengembang. Modul ini paling lama akan dikerjakan

selama 200 jam/115 FPs (1.74 jam/FP). Jika pengembang hanya mampu mengerjakan selama 2.5 jam/FP maka proyek ini akan dikerjakan selama 287.5 jam. Dengan menggunakan perkiraan waktu pengerjaan *Teacher Manager* dengan salah satu modul RUSA yang hampir serupa, pengembang dapat memperkirakan berapa FP dan perhitungan ILF, EIF, EI, EO, EQ yang hampir serupa. Tentu perkiraan ini akan lebih akurat jika memiliki pengalaman yang banyak dan pengembang pun akan makin produktif karena modul yang dikerjakan hampir serupa sehingga bisa menggunakan 1.74 jam/FP sebagai acuan dikarenakan pengelola proyek dan tim pernah mengerjakan proyek yang seperti ini. Hal tersebut disebabkan oleh faktor produktivitas [12].

Estimasi Biaya

Untuk memperkiraan biaya yang akan dikeluarkan dalam suatu proyek, pengelola juga dapat melihat catatan pengerjaan pada proyek *Teacher Manager* dikarenakan proyek ini merupakan proyek yang sejenis dengan RUSA. Jika pada proyek *Teacher Manager* dapat dikerjakan dalam waktu 200 jam dan menggunakan pengukuran faktor produktivitas, maka produktivitas proyek akan seperti Tabel 16.

Tabel 16. Produktivitas Proyek

Peran (A)	Harga/Peran (Dalam .000 Rupiah) (B)	Usaha yang dilakukan (Jam) (C)	(C*100/ Total Jam) (D)	Subtotal Dibayar (Dalam .000 Rupiah) (B*D)
Pengelola Proyek	1000	10	5	5000
System Analyst	180	40	20	3600
Pengembang	50	100	500	2500
Spesialis Keamanan	300	20	10	3000
Tester	100	30	15	1500
Total	-	200	685	15600

Teknik ini sangat baik digunakan untuk merencanakan alokasi tim dan juga dapat digunakan untuk memperkirakan berapa jam yang akan digunakan oleh masing-masing peran. Dengan demikian

pengerjaan proyek RUSA juga dapat diperkirakan biayanya dengan melihat tabel hasil produktivitas proyek Teacher Manager.

SIMPULAN

Analisa Function Point dapat digunakan untuk estimasi waktu dan biaya dalam pengerjaan suatu proyek dan juga dapat digunakan untuk mengukur aplikasi yang dikerjakan, dan pengukuran merupakan faktor penting untuk menentukan produktivitas. Dengan menggunakan analisa function point dan perhitungan *DET*, *RET* dan *FTR* pada proyek lama yang serupa tentu akan mempermudah pengelola proyek dalam estimasi proyek yang serupa dan pengelola proyek yang memiliki banyak pengalaman mengerjakan aplikasi dapat melihat catatan pengerjaan proyek yang pernah dikerjakan untuk dijadikan perbandingan estimasi waktu dan biaya pada proyek yang serupa dengan memperhatikan perhitungan ILF, EIF, EI, EO, dan EQ lalu menghitung FP yang didapat dari perhitungan UFP dan VAF. Dengan begitu pengelola proyek dapat mengestimasi tanpa harus memperhitungkan FP untuk mendapatkan estimasi waktu dan biaya dengan memperhatikan faktor produktivitas dari pengembang.

DAFTAR PUSTAKA

- [1] Sun-Jen Huang, Chieh-Yi Lin, Nan-Hsing Chiu, "Fuzzy Decision Tree Approach for Embedding Risk Assessment Information into Software Cost Estimation Model", *Journal of Information Science and Engineering*, pp. 297-313, 2006.
- [2] Vishal Sharma, Harsh Kumar Verma, "Optimized Fuzzy Logic based Framework for Effort Estimation in software Development", *International Journal of Computer Science*, Pp 30-38, 2010.
- [3] Jyoti Mahajan, D., Dhruve, K, "REBEE – Reusability Based Effort Estimation Technique Using Dynamic Neural Network", *Global Journal Computer Science Technology*, 11, (7), pp. 55–60, 2011.
- [4] Deng, J.D., Purvis, M.K., Purvis, M.A., "Software Effort Estimation: Harmonizing Algorithms and Domain Knowledge in An Integrated Data Mining Approach" *Inf. Sci. Discuss. Pap. Ser.*, (5), pp. 1–13, 2009.
- [5] Mittal, H., Bhatia, P., "A Comparative Study of Conventional Effort Estimation and Fuzzy Effort Estimation Based On Triangular Fuzzy Numbers", *International Journal Computer Science Security.*, (4), pp. 36–47, 2002.
- [6] Benton, A., Bradly, M, The International Function Point User Group (IFPUG) in Function Point Counting Practices Manual – release 4.1'. (SA), 1999.
- [7] Schatzberg, D.R., *Total Quality Management for Maintenance Process Improvement, Journal Software Maintenance. Res. Pract.*, 5, (1), pp. 1–12, 1993.
- [8] Behrens, C. A., "Measuring the Productivity of Computer Systems Development Activities with Function Points", *IEEE Trans. Software Eng.*, SE-9, 648-652, 1989.
- [9] Eijogu, L. O., "Beyond Structured Programming: An Introduction to the Principles of Applied Software Metrics", *Journal .Struct. Progr.* 11, 1990.
- [10] Alvin Alexander, How I estimate Software Development Project, 2014.
- [11] Yuri Marx Pereira Gomez., Functional Size, Effort and Cost of the SOA Projects with Function Points, 2011, diambil dari <http://www.servicetechmag.com/system/application/views/I68/1112-2.pdf>