

# Kajian Metode Perhitungan Metrik *Function-Point* dan Penerapannya pada Dua Perangkat Lunak yang Dipilih

Winangsari Pradani

Program Studi Teknik Informatika, Fakultas Sains dan Teknologi,  
Universitas Al Azhar Indonesia, Jl. Sisingamangaraja, Jakarta 12110

E-mail: [winangsari@uai.ac.id](mailto:winangsari@uai.ac.id)

**Abstrak** - *Function Point* (FP) adalah metoda pengukuran sekaligus satuan ukuran sebuah perangkat lunak (PL). FP mengukur PL dengan cara mengkuantifikasi fungsionalitas PL yang disediakan untuk *user* berdasar pada disain logik. Metrik FP dihitung dari 5 komponen, yakni jumlah *input* pemakai (*external input* - EI), jumlah *output* kepada pemakai (*external output* - EO), jumlah *inquiry* pemakai (*external inquiry* - EQ), jumlah *file* internal yang digunakan (*internal logical file* - ILF), dan jumlah antarmuka eksternal (*external file* - EIF). Penelitian ini bertujuan untuk mempelajari secara seksama cara menghitung FP. PL yang menjadi objek penghitungan dipilih 2 perangkat lunak yang telah ada yakni *Expert System Pembimbing Akademik* (ESPA) dan Sistem Negosiasi Lelang pada Industri Kelapa Sawit (SNLINKS). Berkas teknis yang tersedia dari dua perangkat lunak ini adalah *source code* dan *running application*. Pengukuran FP akan dilakukan dari sudut pandang aplikasi, yakni mengukur aplikasi yang sudah jadi. Dari hasil observasi didapatkan bahwa FP untuk PL ESPA = 82 dan FP untuk PL SNLINKS = 85,36. Perbedaan ukuran FP keduanya tidak terlalu besar sehingga dapat disimpulkan bahwa usaha yang diberikan untuk membangun kedua PL tersebut relatif sama.

**Abstract** - *Function Point* (FP) is a measurement method as well as a measurement unit of software. FP measured by quantifying software functionality provided to the user based on the logical design. FP metrics calculated from the 5 components, namely the amount of user input (*input external* - EI), the amount of output to the user (*external output* - EO), the amount of user inquiry (*external inquiry* - EQ), the number of files that are used internally (*internal logical*

*files* - ILF), and the number of external interfaces (*external file* - EIF). This research has objective to study carefully how to calculate the FP. Two softwares had been chosen to measured, that were *Academic Advisors Expert System* (ESPA) and *Auction and Negotiation System in Palm Oil Industry* (SNLINKS). It only source code and running program that were available as the technical documentation of them. FP measurements will be carried out from the viewpoint of application, ie, measuring the finished application. Obtained from the observation, ESPA software had FP=82 and SNLINKS software had FP= 85.36. Only a slight difference between them so it can be concluded that the effort exerted to build two softwares are relatively similar.

**Keywords** - *Function Point* (FP), *Software Metrics*, *Software Engineering* (RPL)

## I. PENDAHULUAN

Tujuan dari penelitian ini adalah untuk:

- Mempelajari metoda terbaru penghitungan metrik FP yang lebih akurat
- Menghitung FP dari 2 perangkat lunak yang telah ada yakni perangkat lunak *Expert System Pembimbing Akademik* (ESPA) dan Sistem Negosiasi Lelang pada Industri Kelapa Sawit (SNLINKS).

Penelitian ini diharapkan akan bermanfaat pada:

- Perkembangan metrik PL khususnya pada pengukuran volume PL skala kecil dengan memberikan gambaran hasil perhitungan metrik pada kasus nyata.
- Pengajaran topik pengukuran perangkat lunak yang sangat sedikit terpapar pada Teknik

Informatika dengan cara memberikan bahan ini sebagai contoh pengukuran.

Penelitian ini diharapkan dapat menjawab hal-hal berikut ini:

- Sebelum FP dapat dihitung, 5 komponen perhitungan harus dinilai dulu tingkat kompleksitasnya, apakah termasuk rendah, sedang, atau tinggi. Bagaimana menentukan tingkat kompleksitas komponen ?
- Apa kendala yang ditemukan pada saat menentukan kompleksitas komponen dan menghitung FP ?
- Berapa nilai FP untuk kedua PL hasil studi observasi ?

Penghitungan metrik FP akan menggunakan metoda dan petunjuk-petunjuk dari komunitas resmi pengembang metrik FP yakni *International Function Point User Group* (IFPUG).

## II. TINJAUAN PUSTAKA

### 2.1 Metrik Perangkat Lunak

Terdapat ratusan metrik yang telah dibuat untuk PL, tetapi tidak semuanya memberikan dukungan praktis bagi pembangun PL. Beberapa pengukuran terlalu kompleks, yang lainnya sangat esoterik (hanya sedikit yang mengetahuinya) [1]. Ejiogu mendefinisikan sejumlah atribut yang harus dimiliki sebuah metrik yang efektif, yaitu:

- Sederhana dan dapat dihitung. Metrik haruslah mudah untuk dipelajari bagaimana menggunakannya. Proses perhitungannya juga tidak memakan waktu lama.
- Meyakinkan secara empiris maupun intuitif. Metrik harus memenuhi intuisi pembangun mengenai atribut produk yang diukur, misalnya metrik yang mengukur tingkat kohesi akan menghasilkan nilai yang tinggi untuk PL yang memiliki tingkat kohesi tinggi.
- Konsisten pada penggunaan unit dan dimensi
- Bebas terhadap bahasa pemrograman
- Memberikan informasi yang berguna untuk menghasilkan produk berkualitas tinggi.

*Function Point* (FP) adalah unit pengukuran untuk PL seperti jam untuk mengukur waktu, mil untuk mengukur jarak atau Celsius untuk mengukur suhu. FP mengukur PL dengan cara mengkuantifikasi fungsionalitas PL yang disediakan untuk user berdasar pada disain lojik. Yang dimaksudkan *user* di sini adalah *sophisticated user* yang mengerti

sistem dari perspektif fungsional – lebih dari ‘*user*’ yang memberikan pernyataan kebutuhan atau melakukan uji terima (*acceptance testing*).

Metrik FP dapat digunakan secara efektif sebagai alat untuk memprediksi ukuran sebuah sistem yang akan dihasilkan dari model analisis. Model analisis yang digunakan harus dapat memberikan masukan berikut ini kepada metrik FP :

- Jumlah *input* pemakai
- Jumlah *output* kepada pemakai
- Jumlah *inquiry* pemakai
- Jumlah file yang digunakan
- Jumlah antarmuka eksternal

Kemudian FP dihitung dengan [1] :

$$FP = \text{count total} \times [0,65 + 0,01 \times \text{Nilai kompleksitas}]$$

Nilai kompleksitas akan berbeda dari satu PL ke PL lainnya. Nilai FP ini akan merupakan petunjuk terhadap besarnya ukuran PL. Semakin besar nilai FP akan semakin besar pula ukuran PL. Ukuran PL dapat dinyatakan sebagai *Line of Code* (jumlah baris statement dalam *source code*).

Terdapat beberapa metode yang dapat digunakan untuk menghitung FP, pada makalah ini digunakan aturan yang dibangun oleh Alan Albrecht yang kemudian direvisi oleh komunitas *International Function Point User Group* (IFPUG).

Salah satu dari miskonsepsi terbesar dari FP adalah memahami perbedaan fungsionalitas PL antara yang terekspos oleh end user dan fungsionalitas yang diberikan (*delivered*). Salah satu tren yang terjadi pada pengembangan PL sekarang adalah aplikasi swalayan seperti yang digunakan sebagian besar perusahaan penerbangan.

Jika seseorang mengunjungi sebuah website penerbangan, akan terlihat layar yang relatif sederhana. *End user* hanya diminta untuk memilih awal penerbangan dan tujuan dan tanggal keberangkatan. Selintas ini merupakan permintaan yang sederhana, tetapi sebenarnya pelaksanaannya kompleks. Proses ini mencakup 1000 proses elementer. Seluruh rute yang mungkin harus dikalkulasi, nama kota dikonversi menjadi tiga karakter internasional, antarmuka dikirimkan ke semua perusahaan penerbangan, semuanya merupakan proses yang kompleks dan robust. Ketika kita mengukur aplikasi PL, kita harus

mengerti mana yang terekspos dan mana yang di bawah permukaan.

## 2.2 Jenis-jenis Penghitungan FP

Terdapat tiga tipe Penghitungan FP PL: Development (Pembangunan), Enhancement (Perbaikan), dan Maintenance (Pemeliharaan). Penghitungan FP pada saat Pembangunan dilakukan pada pekerjaan pembangunan PL baru. Penghitungan FP pada saat Perbaikan merupakan penghitungan FP untuk mengukur proyek perbaikan. Penghitungan FP pada saat Pemeliharaan PL akan menilai FP aplikasi sesudah selesai produksi. Nilai 'baseline' ini dapat menjadi dasar bagi metrik aplikasi keseluruhan seperti jam pemeliharaan total. Metrik ini dapat digunakan untuk mencatat (*track*) jumlah jam pemeliharaan per FP.

## 2.3 Data Functions

Data Functions terdiri atas ILF dan EIF.

- **Internal Logical File (ILF):** data atau informasi kontrol yang perlu dipelihara melalui pengaksesan *insert*, *update*, atau *delete* di dalam sistem aplikasi.
- **External Interface File (EIF):** data atau informasi kontrol yang diperlukan oleh sistem aplikasi, tetapi disimpan atau dipelihara di luar sistem (oleh sistem lain)

## 2.4 Elementary Process (EP)

Proses elementer merupakan unit aktivitas terkecil yang berarti bagi user (dilakukan oleh user melalui software aplikasi). Proses elementer harus berdiri sendiri dan hasilnya tetap menjaga konsistensi sistem. Proses elementer terdiri atas 3: *External Input* (EI), *External Inquiry* (EQ), dan *External Output* (EO)

1. **External Input (EI):** proses elementer yang memproses data dari luar sistem dan menyimpan hasilnya ke dalam system
2. **External Inquiry (EQ):** proses elementer yang menyajikan data ke luar sistem (hanya menyajikan data apa adanya, tidak ada proses pengolahan data)
3. **External Output (EO):** proses elementer yang memproses data dan menyajikan informasi ke luar sistem (ada proses komputasi atau kalkulasi ketika menyajikan informasi ke luar) Logika proses yg mungkin terjadi dalam EI/EO/EQ

Tabel 1. Hubungan antara Logika Proses dengan kemungkinan Perhitungan pada EI/EO/EQ

Proses	EI	EO	EQ
1. Validasi	c	c	c
2. Kalkulasi rumus matematika	c	m*	n
3. Konversi nilai	c	c	c
4. Seleksi atau filtering data berdasarkan kriteria tertentu	c	c	c
5. Anallisa kondisi untuk menentukan kondisi mana yang berlaku	c	c	c
6. Meng-update paling sedikit 1 (satu) ILF	m*	m*	n
7. Membaca paling sedikit 1 (satu) ILF atau EIF	c	c	m
8. Membaca data atau informasi kontrol	c	c	m
9. Menghasilkan data turunan	c	m*	n
10. Mengubah perilaku sistem	m*	m*	n
11. Menyiapkan dan menyajikan informasi ke luar sistem	c	m	m
12. Menerima data atau informasi kontrol dari luar sistem	m	c	c
13. Mengurutkan ( <i>sorting</i> ) data	c	c	c

Catatan:

c=dapat melakukan (tidak harus); m=*mandatory*; m\* = *mandatory*, minimal salah satu dari yg m\*; n = tidak dapat melakukannya

Pada tabel 1 terlihat bahwa beberapa proses tipikal dapat memiliki proses elementer EI, EO dan EQ seperti tercantum.

## 2.5 Tingkat Kompleksitas EI, EQ dan EO berdasarkan nilai DET, RET dan FTR

Penghitungan nilai EI, EQ dan EO bergantung dari tingkat kompleksitas komponen-komponen tersebut. IFPUG telah membuat prosedur bagaimana menentukan nilai kompleksitas komponen agar dapat ditentukan apakah termasuk kategori rendah, sedang dan tinggi.

Pengkategorian komponen didasarkan pada perhitungan DET, RET dan FTR yang rinciannya adalah sebagai berikut:

1. **Data Element Type (DET):** adalah elemen data yang dikenal oleh user (*non-repeatable data field*). Contoh: Nama Pelanggan, No. Pelanggan, Tanggal Lahir, dll.
2. **Record Elemen Type (RET):** adalah subgrup data yang dikenal oleh user. Misalnya: data pelanggan terdiri dari biodata, data finansial, data tanggungan keluarga

3. **File Type Reference (FTR):** adalah ILF yang dibaca atau diakses oleh proses elementer (EI/EO/EQ), atau EIF yang dibaca oleh proses elementer.

Tabel 2 memperlihatkan bagaimana skor EI (3, 4, atau 6) ditentukan berdasarkan jumlah DET dan FTR. Tabel 3 memperlihatkan bagaimana skor EO dan EQ ditentukan berdasarkan jumlah DET dan FTR

### 2.6 Langkah Penghitungan FP

Berikut ini adalah langkah-langkah penghitungan FP:

1. Menentukan tipe penghitungan FP, apakah Pembangunan, Perbaikan, atau Pemeliharaan.
2. Menentukan batas-batas aplikasi
3. Mengidentifikasi dan menghitung nilai ILF dan EIF
4. Mengidentifikasi dan menghitung nilai EI, EQ, EO
5. Menjumlahkan nilai ILF, EIF, EI, EQ, EO sebagai nilai FP yang belum di-adjust (*Unadjusted Function Point – UFP*)
6. Menentukan nilai faktor penyetelan (*Value Adjustment Factor – VAF*)
7. Menghitung FP yang telah disetel (*adjusted FP*)

FP final (*adjusted FP*) merupakan kombinasi dari FP *unadjusted* (UFP) dan karakteristik sistem secara umum (*general system characteristics – GSC*).

Tabel 2. Skor EI berdasarkan DET dan FTR

FTR	DET		
	1-4	5-15	>15
<2	Rendah (3)	Rendah (3)	Sedang (4)
2	Rendah (3)	Sedang (4)	Tinggi (6)
>2	Sedang (4)	Tinggi (6)	Tinggi (6)

Tabel 3. Skor EO/EQ berdasarkan DET dan FTR

DET	1 - 5		5 - 19		> 19	
	Skor	FP	Skor	FP	Skor	FP
0-1	Rendah	4/3	Rendah	4/3	Sedang	5/4
2-3	Rendah	4/3	Sedang	5/4	Tinggi	7/6
>3	Sedang	5/4	Tinggi	7/6	Tinggi	7/6

### 2.7 General System Characteristics (GSC)

Sebelum penghitungan FP dapat dilakukan, tingkat kompleksitas sistem harus ditentukan terlebih dahulu dengan menjumlah skor yang didapatkan dari ke-14 kriteria. Setiap kriteria memiliki bobot 1 sampai 5. 1 untuk sistem yang paling sederhana dan 5 untuk sistem yang paling kompleks.

Berikut ini adalah ke-14 kriteria untuk penghitungan GSC:

1. **Data Communication:** tingkat kebutuhan komunikasi langsung antara aplikasi dan *processor*.
2. **Distributed Data Processing:** tingkat kebutuhan transfer data antara komponen-komponen aplikasi.
3. **Performance:** tingkat *response time* dan *throughput* yg perlu dipertimbangkan dalam pengembangan aplikasi.
4. **Heavily used configuration:** tingkat kebutuhan dimana setting konfigurasi komputer berpengaruh terhadap pengembangan aplikasi.
5. **Transaction rate:** tingkat kecepatan transaksi bisnis yang berpengaruh terhadap pengembangan aplikasi.
6. **Online Data Entry:** tingkat kebutuhan peng-input-an data secara interaktif.
7. **End-user efficiency:** tingkat kemudahan penggunaan aplikasi.
8. **Online Update:** tingkat kebutuhan ILF di-update secara online.
9. **Complex Process:** tingkat kesulitan logika proses yang mempengaruhi proses development.
10. **Reusability:** tingkat kebutuhan aplikasi dan kode program aplikasi dirancang dan dikembangkan untuk bisa digunakan pada aplikasi lain.
11. **Installation Ease:** tingkat kemudahan konversi ke sistem baru yg berpengaruh pada proses development.
12. **Operational Ease:** tingkat kemudahan aplikasi dalam aspek-aspek operasional, seperti *start-up*, *backup*, dan proses *recovery*.
13. **Multiple Sites:** tingkat kebutuhan aplikasi dapat dioperasionalkan pada lingkungan *hardware* dan *software* yang berbeda-beda.
14. **Facilitate Change:** tingkat kemudahan aplikasi untuk modifikasi logika proses maupun struktur data.

## 2.8 Formula TDI, VAF, UFP

Formula berikut ini dipakai untuk menghitung UFP (Unadjusted Function Point) [2]:

$$TDI = \sum_{i=1}^{14} GSC_i \quad (1)$$

$$VAF = TDI * 0,01 + 0,65 \quad (2)$$

$$UFP = \sum_{i=1}^5 TE_i * W_i \quad (3)$$

TDI adalah jumlah seluruh nilai dari 14 kriteria GSC. Ini merupakan skor sistem untuk PL yang diukur. VAF merupakan nilai penyesuaian yang telah ditentukan. UFP adalah skor akumulasi kelima komponen tanpa dikalikan dengan nilai kompleksitas sistem.

Berikut ini adalah formula FP yang dibedakan menjadi 3, yakni FP untuk jenis pengukuran PL pada saat Pembangunan (DFPC); pada saat Perbaikan (EFPC); dan pada saat PL telah jadi/ Aplikasi (AFPC) [2].

1. Development: DFPC = (UFP + CFP) \* VAF
2. Enhancement: EFPC = [(ADD + CHGA + CFP) \* VAFA] + (DEL \* VAFB)
3. Application: AFPC = [(UFPB + ADD + CHGA) - (CHGB + DEL)] \* VAFA

Keterangan:

- a. ADD : UFP dari elemen-elemen fungsional yang baru (tambahan) pada proyek enhancement
- b. CFP : UFP dari elemen-elemen fungsional yang diperlukan untuk kebutuhan konversi sistem
- c. CHGA : UFP dari elemen-elemen fungsional yang dimodifikasi oleh proyek enhancement (setelah modifikasi)
- d. CHGB : UFP dari elemen-elemen fungsional yang dimodifikasi oleh proyek enhancement (sebelum modifikasi)
- e. DEL : UFP dari elemen-elemen fungsional yang dihapus oleh proyek enhancement
- f. VAF : value adjustment factor
- g. VAFA : VAF setelah proyek enhancement selesai
- h. VAFB : VAF sebelum proyek enhancement dilakukan
- i. UFP : unadjusted function point
- j. UFPB : UFP aplikasi sebelum proyek enhancement dimulai

## III. METODA PENELITIAN

Terdapat 2 perangkat lunak yang akan diukur menggunakan ukuran FP dengan beberapa pertimbangan :

1. Perangkat Lunak yang mudah didapat sumber-sumber dokumentasinya adalah perangkat lunak yang sudah tersedia.
2. Mengumpulkan berkas-berkas teknis kedua software. Dokumen teknis yang tersedia adalah source code dan aplikasinya sendiri (running application).

Jadi diputuskan untuk memilih pengukuran kedua perangkat lunak ini berjenis aplikasi. Maksudnya, perangkat lunak akan diukur hanya dari tampilan hasil running aplikasi.

Langkah pertama adalah menjalankan semua fungsi aplikasi yang mungkin, kemudian menangkap tampilan yang dihasilkan. Untuk setiap tampilan aplikasi akan diidentifikasi dan dihitung EI, EO, EQ, ILF dan EIF nya. Jadi jika dianalogikan dengan produk baju, kita akan mengukur bahan yang diperlukan dari baju yang sudah jadi, bukan dari mengukur badan lalu memperkirakan besar bahan yang diperlukan.

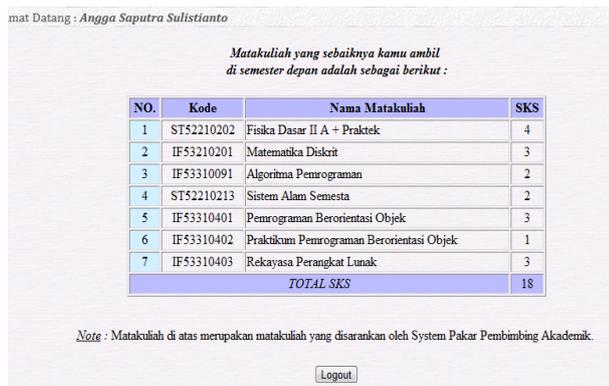
Untuk menghitung ILF dan EIF, cukup dilihat data yang betul-betul ditampilkan di layar dan data yang dimasukkan oleh user. Kemudian, dengan pengetahuan sebagai disainer data, kita membuat tabel data atau skema relasi yang telah normal (dinormalisasi).

Setelah nilai kelima komponen didapat, UFP dapat dihitung. Kemudian dengan petunjuk GSC kita dapat menghitung TDI dan VAF. AFP yang diinginkan didapat dengan mengkombinasikan nilai VAF dan UFP.

## IV. PEMBAHASAN

### 4.1 Pengukuran FP Program 1: Expert System Pembimbing Akademik (ESPA)

Gambar 1 merupakan tampilan utama PL ESPA



Gambar 1. Tampilan utama PL ESPA

ESPA adalah software cerdas yang dapat membantu mahasiswa UAI menentukan pengambilan matakuliah dan jumlah sks pada semester tertentu.

1. Input : Daftar Nilai Matakuliah yang sudah diambil dan IP Semester Terakhir
2. Output : Jumlah sks, Saran Matakuliah yang diambil (sudah berdasarkan prioritas)

Jumlah Layar ESPA : 19

Tabel 4 adalah hasil penghitungan 5 komponen ESPA:

Tabel 4. Hasil Perhitungan UFPC dari PL ESPA

Function Type	Number	Functional Complexity	Complexity Totals	Function Type Totals
ILF	7	Low	7	49
	0	Average	10	
	0	High	15	
EIF	0	Low	5	0
	0	Average	7	
	0	High	10	
EI	12	Low	3	36
	2	Average	4	
	0	High	6	
EO	0	Low	4	0
	0	Average	5	
	0	High	7	
EQ	5	Low	3	15
	0	Average	4	
	0	High	6	
<b>Unadjusted Function Point Count</b>				<b>108</b>

Berdasarkan penetapan GCS dan rumus (1), (2), dan (3) didapatkan perhitungan untuk PL ESPA sebagai berikut :

$$TDI = 11$$

$$VAF = 11 * 0.01 + 0.65 = 0,76$$

$$UFP = 108$$

$$DFPC = 108 * 0,76 = 82$$

#### 4.2 Pengukuran FP Program 2: Sistem Negosiasi Lelang pada Industri Kelapa Sawit (SNLINKS)

SNLINKS merupakan aplikasi yang digunakan untuk membantu kegiatan lelang kelapa sawit berbasis *supply chain management*. Pengguna sistem ini adalah penawar, pembeli, dan penyelenggara lelang. Gambar 2 menunjukkan tampilan utama PL SNLINKS. Jumlah Layar PL ini ada 15 buah.

Hasil perhitungan UFP dari PL SNLINKS adalah 97 yang merupakan penjumlahan dari kelima nilai ILF, EIF, EI, EO, dan EQ. Rinciannya terdapat pada Tabel 5.



Gambar 2. Tampilan utama PL SNLINKS

Tabel 5. Hasil Perhitungan UFPC dari PL SNLINKS

Function Type	Number	Functional Complexity	Complexity Totals	Function Type Totals
ILF	7	Low	7	49
	0	Average	10	
	0	High	15	
				49

EIF	0	Low	5	0	0
	0	Average	7	0	
	0	High	10	0	
EI	6	Low	3	18	22
	1	Average	4	4	
	0	High	6	0	
EO	3	Low	4	12	22
	2	Average	5	10	
	0	High	7	0	
EQ	0	Low	3	0	4
	1	Average	4	4	
	0	High	6	0	
<b>Unadjusted Function Point Count</b>					<b>97</b>

Berdasarkan penetapan GSC dan rumus (1), (2), dan (3) didapatkan perhitungan untuk PL SNLINKS sebagai berikut:

$$TDI = 23$$

$$VAF = 23 * 0.01 + 0.65 = 0,88$$

$$UFP = 97$$

$$DFPC = 97 * 0,88 = 85,36$$

## V. KESIMPULAN DAN SARAN

Penelitian ini dapat menjawab pertanyaan-pertanyaan di awal penelitian :

- Sebelum FP dapat dihitung, terdapat prosedur penilaian tingkat kompleksitas 5 komponen perhitungan, apakah termasuk rendah, sedang, atau tinggi. Masalahnya adalah: Bagaimana menentukan tingkat kompleksitas komponen ?

Tingkat kompleksitas komponen FP sangat bergantung pada perhitungan RET, FTR, DET. Ketiganya berkaitan dengan data yang diolah-disimpan-diambil. Artinya, **perhitungan FP ini cocok untuk perangkat lunak yang memiliki karakteristik 'mengolah data'**. Untuk perangkat lunak yang banyak memiliki karakteristik yang berbeda, seperti perangkat lunak kendali, perangkat lunak cerdas yang lebih banyak proses 'kalkulasi' nya daripada mengolah-menyimpan-dan mengambil data, unsur RET, FTR dan DET tidak lebih penting dari 'proses' nya sendiri.

- Apa kendala yang ditemukan pada saat menentukan kompleksitas komponen dan menghitung FP ?

Berkas teknis tidak lengkap. Agar perhitungan FP akurat, diperlukan berkas teknik rekayasa perangkat lunak yang lengkap. Selama ini, berkas teknis (dokumentasi teknis) belum terlalu mendapatkan perhatian untuk dilengkapi.

- Berapa nilai FP untuk kedua PL ?

FPC untuk PL ESPA = 82; FPC untuk PL SNLINKS = 85,36. Perbedaan ukuran FP keduanya tidak terlalu besar, dapat disimpulkan bahwa usaha yang diberikan untuk membangun kedua perangkat lunak relatif sama.

## Saran

- Untuk penelitian selanjutnya, disarankan untuk mengukur perangkat lunak jenis lainnya, seperti perangkat lunak *realtime*, perangkat lunak sistem, dan perangkat lunak jenis lainnya.
- Disarankan juga untuk dapat mengukur perangkat lunak dari berkas teknis yang lengkap sehingga dapat dibandingkan hasil pengukurannya jika diukur dari sisi awal rekayasa (dokumen user requirement), dan dari sisi produk yang sudah jadi.

## DAFTAR PUSTAKA

- , "Function Point Counting Practices Manual Release 4.3.1", International Function Point Users Group (IFPUG), 2010
- , "Function Points Analysis Training Course", SoftwareMetrics.com, Longstreet Consulting Inc
- , "Function Points Analysis Training Course ANSWERS", SoftwareMetrics.com, Longstreet Consulting Inc
- Singhal, Neelam Bawane nee' and Srikrishna, C.V., "A Case Study to Assess the Validity of Function Points", World Academy of Science, Engineering and Technology 42 2008
- Buglione, Luigi, "IFPUG Function Point Analysis (FPA) v4.2 Quick Guide", www.eng-it.it/spmiq
- Zulfikar, "Expert System Pembimbing Akademik", Laporan Tugas Akhir, Teknik Informatika Universitas Al Azhar Indonesia, 2010
- Fiki Arfiandi, "Sistem Negosiasi Lelang pada Industri Kelapa Sawit", Laporan Tugas Akhir, Teknik Informatika Universitas Al Azhar Indonesia, 2012