

PERANCANGAN UJI APLIKASI KODE DIAGNOSA DAN TINDAKAN *DISEASES AND PROCEDURE OF EYE AND ADNEXA* DENGAN METODE NORMALISASI DATA PADA HASIL EVALUASI RCAF (*RELATIVE COMPLEXITY ADJUSTMENT FACTOR*)

Puguh Yudho Trisnanto¹, Ganif Djuwadi², Arif Mulyadi³

¹Perekam Medis dan Informasi Kesehatan-Poltekkes Kemenkes Malang

²Promosi Kesehatan-Poltekkes Kemenkes Malang

³Keperawatan Blitar-Poltekkes Kemenkes Malang

INFO ARTIKEL

Riwayat Artikel:

Diterima: 20-5-2017

Disetujui: 20-6-2017

Kata kunci:

*international function point user
group;
general system characteristics;
klomogorov-smirnov test;
crude function points*

ABSTRAK

Abstract: Application of Diagnostic and Action Codes Diseases and Procedure of eye and adnexa indicates DAD of FP test implementation has 3 (three) source data terminators which include: Terminator data source action, diagnostic search and diagnostics, each terminator produces source data consisting of. The action terminator produces the input data source action. Terminator search diagnostics generate diagnostic code source data and action codes. The diagnostic terminator produces the diagnostic input source data, the three terminators are processed into the Diagnostic Code and Action application, generating output output in the form of destination data terminals. Disease diagnosis output. Diagnostic output of action, process Application diagnostic code and action make connecting connection to IFPUG storage source with GSC information to D1 CFP as source data to process 1.1. Simple level, with destination data yielding 53.94 as output output to developer knowledge, IFPUG also generates GSC / RCAF to D2 data normality with sig> 0.1 as source data to process 1.2. Test of normality with sig 0.200 * using klomogorov-Smirnov test as output output to GSC information as application development diagnose code and action by system user at RSI UNISMA Malang.

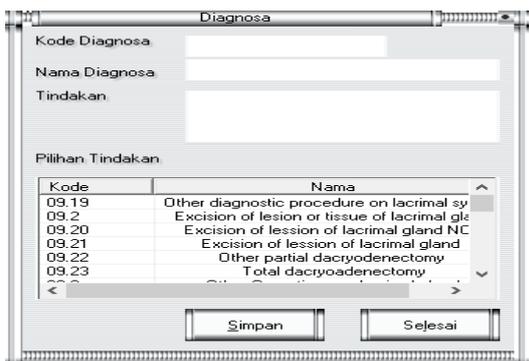
Abstrak: Penerapan kode perilaku diagnostik dan tindakan, prosedur mata dan adneksa menunjukkan bahwa DAD pelaksanaan uji FP memiliki tiga terminator data sumber, meliputi tindakan sumber data terminator, penelusuran diagnostik, dan diagnostik. Terminator tindakan menghasilkan tindakan sumber data masukan. Diagnostik pencarian terminator menghasilkan kode sumber kode diagnostik dan kode tindakan. Terminator diagnostik menghasilkan data sumber input diagnostik, ketiga terminator tersebut diproses ke dalam aplikasi Diagnostic Code and Action, menghasilkan output output dalam bentuk terminal data tujuan. Hasil diagnosis penyakit. Hasil diagnostik tindakan, proses Aplikasi kode diagnostik dan tindakan membuat koneksi penghubung ke sumber penyimpanan IFPUG dengan informasi GSC ke D1 CFP sebagai data sumber untuk memproses 1.1. Tingkat sederhana, dengan data tujuan menghasilkan 53,94 sebagai keluaran output untuk pengetahuan pengembang, IFPUG juga menghasilkan data normalitas GSC / RCAF ke D2 dengan sig> 0,1 sebagai data sumber untuk memproses 1.2. Uji normalitas dengan sig 0.200 * menggunakan uji klomogorov-Smirnov sebagai output output terhadap informasi GSC sebagai pengembangan aplikasi diagnosa kode dan tindakan oleh pengguna sistem pada RSI UNISMA Malang.

Alamat Korespondensi:

Puguh Yudho Trisnanto
Perekam Medis dan Informasi Kesehatan
Poltekkes Kemenkes Malang
Jalan Besar Ijen No.77C Malang
E-mail: jkonsultasirm@gmail.com

Desain sistem informasi merupakan penyampaian informasi secara tidak langsung, meliputi data dan informasi yang terbentuk menjadi komponen-komponen untuk mencapai tujuan tertentu John Burch dan Gari Grundnitski (2000:28). Dengan kata lain, Desain sistem informasi merupakan hal mendasar dalam menyampaikan informasi ke user. Hasil Desain sistem informasi ini masuk ke dalam PL (Perangkat Lunak) tidak dipungkiri PL merupakan hal yang terlewatkan di organisasi atau institusi ini dikarenakan ketika orang awam melihat PL hanya sekedar sebagai operator tidak bertindak sebagai analis sistem. Desain sistem informasi PL ini Aplikasi Kode Diagnosa dan Tindakan *Diseases and Procedure of eye and adnexa* merupakan aplikasi kode diagnosa dan tindakan untuk membantu petugas kode mencari data yang dibutuhkan secara cepat dan akurat. Di luar dari manajemen sistem tersebut ada hal mendasar yang dilewatkan oleh pembuat Aplikasi Kode Diagnosa dan Tindakan *Diseases and Procedure of eye and adnexa*. PL yang digunakan belum memiliki uji PL dengan permasalahan tersebut penulis melakukan uji metode FP dengan tabel RCAF yang dianalisa dengan metode normalisasi data untuk menghasilkan uji PL yang sesuai dengan kebutuhan pengguna di RSI Malang UNISMA.

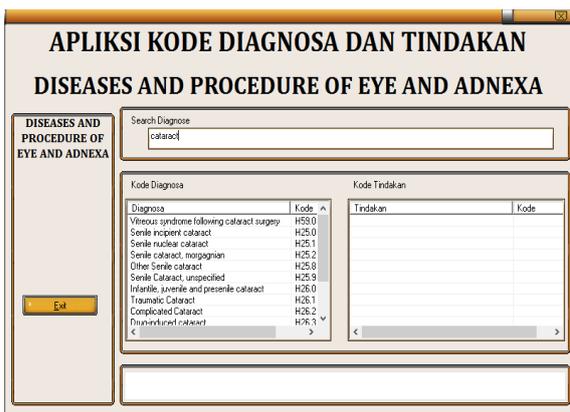
Aplikasi kode diagnosa merupakan sistem informasi yang berhubungan dengan manajemen kode diagnosa dan tindakan di RSI Malang UNISMA. Memiliki modul Aplikasi, meliputi search diagnosa, input diagnosa, dan input tindakan. Pentingnya dilakukan uji PL dengan metode FP dikarenakan Akan dibangun sebuah software dalam pengelolaan Sistem Informasi Kode Diagnosa dan Tindakan *Diseases and Procedure of eye and adnexa*, yang rencanakan untuk mengelola Kode Diagnosa dan Tindakan untuk menghasilkan berbagai macam laporan, seperti laporan kunjungan pasien, laporan diagnosa penyakit, dan laporan tindakan diagnosa penyakit memungkinkan berinteraksi dengan perangkat keras printer serta menyediakan *web service* agar Software Sumber Daya Manusia ini dapat berkolaborasi dengan Software Manajemen di RSI UNISMA Malang. Dari hasil modul Aplikasi Kode Diagnosa dan Tindakan dalam bentuk Menu page tersebut dilakukan dengan menggunakan Uji PL dengan metode FP (*Functions Point*) menggunakan tabel RCAF dengan analisa uji Normalisasi data. Hasil uji PL ini digunakan untuk Pengembangan Aplikasi kode diagnosa dan tindakan sebagai *Knowledge* pengembang dan sebagai laporan hasil uji PL ke RSI UNISMA Malang.



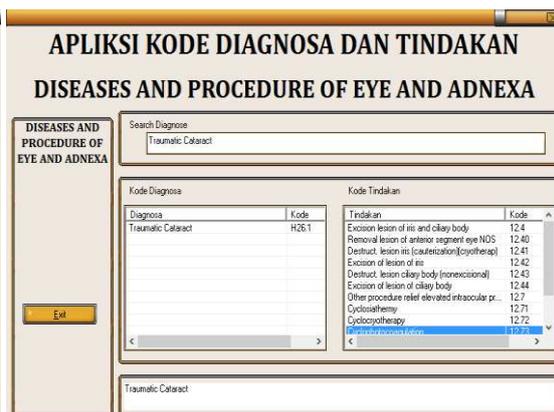
Gambar 1. Input Diagnosa



Gambar 2. Input Tindakan

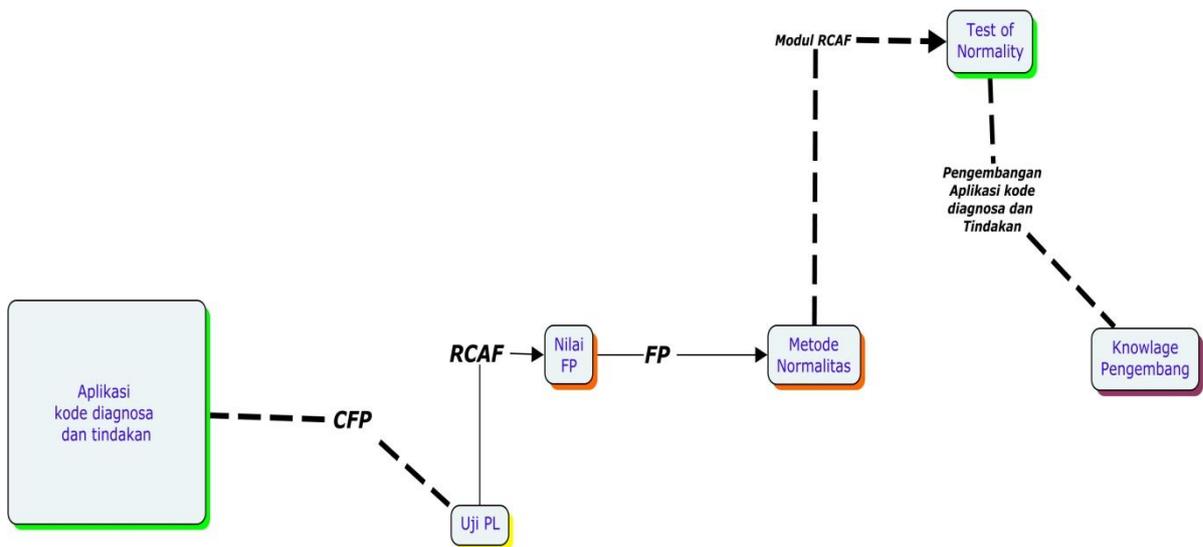


Gambar 3. Pencarian diagnosa penyakit



Gambar 4. Pencarian tindakan medis

METODE



Gambar 5. Konsep Design Map Uji PL

Aplikasi Kode Diagnosa dan Tindakan

Aplikasi kode diagnosa dan tindakan memiliki beberapa modul yang terdiri atas satu paket modul dengan menggunakan GSC (General System Characteristics), dalam bentuk level derajat kompleksitas CFP dengan metode uji PL dengan menggunakan formulir RCAF perhitungan dari hasil RCAF, di gabungkan dengan level derajat kompleksitas CFP untuk menghasilkan nilai FP. Angka 0.65 dan 0.01 adalah ketetapan atau konstanta yang dibuat oleh *Internasional Function Point User Group (IFPUG)*. Sedangkan nilai RCAF digunakan untuk mengetahui hasil normalitas data untuk menghitung bobot kompleksitas dari software berdasarkan 14 karakteristik. Penilaian kompleksitas memiliki skala 0 s/d 5. Keterangan 0 = Tidak Pengaruh, 1 = Insidental, 2 = Moderat, 3 = Rata-rata, 4 = Signifikan, dan 5 = Essential hasil uji normalitas data ini ditujukan kepada knowledge Pengembangan Aplikasi kode diagnosa dan tindakan di RSI UNISMA Malang.



Gambar 6. Tahapan Perhitungan Function Point

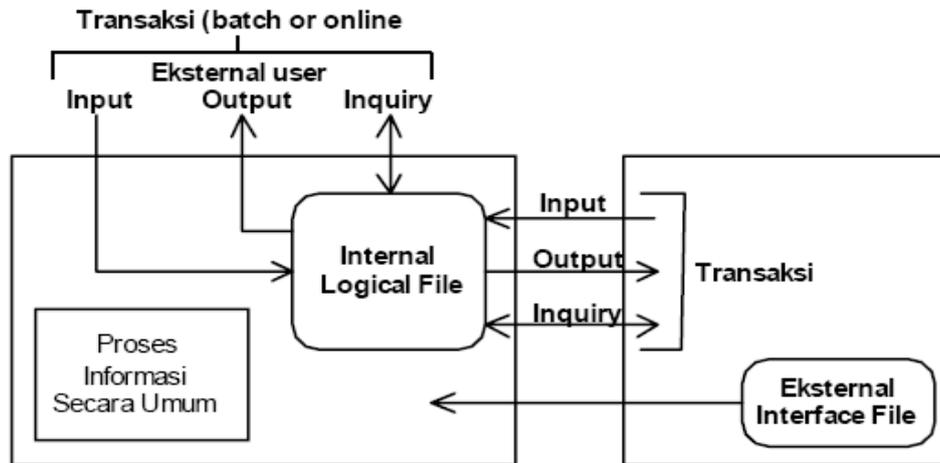
Gambar 2. Tahapan perhitungan FP merupakan Metode Penelitian PL (Perangkat Lunak) Aplikasi Kode Diagnosa dan Tindakan *Diseases and Procedure of eye and adnexa. Function Point* pertama kali di terbitkan pada tahun 1979. Pada tahun 1984 Albrecht menyempurnakan metode *Function Point*. *Internasional Function Point User Group (IFPUG)* didirikan, beberapa versi *function point* sebagai Pedoman telah diterbitkan oleh IFPUG, untuk mengukur perangkat lunak maka dapat menggunakan *Function Point* yang biasa disingkat dengan FP. *Function Point* teknik terstruktur dalam memecahkan masalah dengan cara memecah sistem menjadi komponen yang lebih kecil dan menetapkan beberapa karakteristik dari sebuah perangkat lunak sehingga dapat lebih mudah dipahami dan dianalisis. *Function Point* mengukur dari perspektif *functional* dari perangkat lunak yang akan dibangun, terlepas dari bahasa program, metode *development* atau *platform* perangkat keras yang digunakan, *function point* harus dilakukan oleh orang terlatih dan berpengalaman dalam pengembangan perangkat lunak karena dalam memberikan nilai-nilai dari setiap komponen *function point* bersifat subjektif, dan akan wajar apabila hasil perhitungan *function point* seseorang akan berbeda dengan yang lain.

Pengerjaan *function point* harus dimasukkan sebagai bagian dari rencana proyek secara keseluruhan. Artinya harus dijadwalkan dan direncanakan pengerjaannya. Hasil dari pengukuran menggunakan *Function Point* dapat digunakan untuk mengestimasi biaya dan *effort* yang diperlukan dalam pengembangan perangkat lunak.

Tabel 1. General System Characteristics

No. Uji	Jenis Variabel Pengujian	Subjek Pengujian	Keterangan Subjek Pengujian	Bobot
1	X1	Tingkat kompleksitas kehandalan backup/recovery	Apakah operasi seperti <i>backup</i> , <i>startup</i> , dan <i>recovery</i> dilakukan secara otomatis?	[0/1/2/3/4/5]
2	X2	Tingkat kompleksitas komunikasi data	Berapa banyak fasilitas komunikasi yang ada untuk membantu pertukaran informasi dengan penerapan <i>system</i> aplikasi?	[0/1/2/3/4/5]
3	X3	Tingkat kompleksitas pemrosesan terdistribusi	Bagaimana data di distribusikan dan pengolahan fungsi ditangani?	[0/1/2/3/4/5]
4	X4	Tingkat kompleksitas kebutuhan akan kinerja	Seberapa lama waktu yang diperlukan dan performa secara keseluruhan	[0/1/2/3/4/5]
5	X5	Tingkat kebutuhan lingkungan operasional	Bagaimana platform perangkat keras yang digunakan saat ini dimana aplikasi akan dieksekusi?	[0/1/2/3/4/5]
6	X6	Tingkat kebutuhan knowledge pengembang	Apakah spesifikasi aplikasi didesain, dikembangkan dan didukung untuk berbagai pengembangan kode diagnosa dan tindakan	[0/1/2/3/4/5]
7	X7	Tingkat kompleksitas updating file master	Berapa banyak data di ubah secara online?	[0/1/2/3/4/5]
8	X8	Tingkat kompleksitas instalasi	Apakah konversi dan instalasi dilakukan secara otomatis?	[0/1/2/3/4/5]
9	X9	Tingkat kompleksitas aplikasi input, output, query online dan file	Berapa persentase dari informasi yang dimasukkan secara online?	[0/1/2/3/4/5]
10	X10	Tingkat kompleksitas pemrosesan data	Apakah proses internal yang dilakukan kompleks?	[0/1/2/3/4/5]
11	X11	Tingkat ketidakmungkinan penggunaan kembali dari kode (<i>reuse</i>)	Apakah aplikasi didesain dan dikembangkan untuk memudahkan pengguna?	[0/1/2/3/4/5]
12	X12	Tingkat variasi organisasi pelanggan	Apakah spesifikasi aplikasi didesain, dikembangkan dan didukung untuk berb agai situs dengan berbagai organisasi?	[0/1/2/3/4/5]
13	X13	Tingkat kemungkinan perubahan/fleksibilitas	Apakah aplikasi yang dirancang untuk pengguna efisien?	[0/1/2/3/4/5]
14	X14	Tingkat kebutuhan kemudahan penggunaan	Apakah spesifikasi aplikasi didesain, dikembangkan dan didukung untuk memfasilitasi perubahan dan kemudahan penggunaan oleh <i>user</i> ?	[0/1/2/3/4/5]

Tabel 1. GSC (*General System Characteristics*) dihitung berdasarkan pada keseluruhan kompleksitas sistem. Cara menghitung VAF (*Value Adjustent Factor*) dengan menggunakan 14 (empat belas) GSC (*General System Characteristics*), dimana nila masing-masing dari GSC berskala 0 (nol) sampai 5 (lima). Skala 0 (nol) menunjukkan tidak adanya pengaruh dan skala 5 (lima) menunjukkan adanya pengaruh yang luas terhadap keseluruhan proyek. RCAF digunakan untuk menghitung bobot kompleksitas dari software berdasarkan 14 karakteristik. Penilaian kompleksitas memiliki skala 0 s/d 5. Keterangan 0 = Tidak Pengaruh, 1 = Insidental, 2 = Moderat, 3 = Rata-rata, 4 = Signifikan, dan 5 = Essential.



Gambar 7. Elemen Analisis Function Point

Perhitungan CFP digunakan untuk mengukur proses informasi. Ada beberapa komponen yang dilibatkan dalam pengukuran ini. Komponen ini memiliki kategori "sederhana", "menengah" atau "kompleks" tergantung pada karakteristik kompleksitas yang dimiliki. Perhitungan CFP melibatkan lima komponen dalam analisis sistem, yaitu (1) jumlah macam aplikasi input, (2) jumlah macam aplikasi output, (3) jumlah macam aplikasi query online/inquiry – aplikasi ini berhubungan dengan query terhadap data yang tersimpan, (4) jumlah macam file/tabel logic yang terlibat, dan (5) jumlah macam interface eksternal – output atau input yang dapat berhubungan dengan komputer lewat komunikasi data, CD, disket, dan lain-lain. Gambar 2 menggambarkan komponen-komponen analisis tersebut dalam bentuk diagram. Sebelum mengukur CFP, terlebih dahulu diidentifikasi komponen-komponen yang dalam rancangan software tersebut. Dalam hal ini suatu diagram arus data dapat digunakan. Komponen-komponen yang telah teridentifikasi tersebut selanjutnya dikelompok-kelompokkan menjadi "sederhana", "sedang" dan "kompleks" berdasarkan kompleksitasnya. Setelah itu, jumlah masing-masing komponen yang telah dikategorisasi dapat dimasukkan ke dalam tabel CFP.

Tabel 2. Crude Function Points (CFP)

Komponen Sistem Software	Level Kompleksitas									Total CFP
	Sederhana			Menengah			Kompleks			
	Count	Faktor Bobot	Point	Count	Faktor Bobot	Point	Count	Faktor Bobot	Point	
	A	B	C=AxB	D	E	F=DxE	G	H	I=GxH	J=C+F+I
Input	-	3	-	-	4	-	-	6	-	-
Output	-	4	-	-	5	-	-	7	-	-
Query Online	-	3	-	-	4	-	-	6	-	-
File logic	-	7	-	-	10	-	-	15	-	-
Interface Eksternal	-	5	-	-	7	-	-	10	-	-
Total CFP										

Crude Function Points (CFP) adalah untuk menghitung bobot nilai dari komponen-komponen *Function Point* yang dikaitkan dengan software yang akan dibuat. Komponen-komponen *Function Point* terdiri atas lima buah, yaitu (1) tipe input, berkaitan dengan interface yang lakukan pengguna/user dalam memasukan data pada aplikasi, (2) tipe output, berkaitan dengan output yang dihasilkan aplikasi untuk pengguna/user yang dapat berupa laporan di print atau yang ditampilkan pada layar, (4) tipe query/search/view, berkaitan dengan *query* terhadap data yang tersimpan, (4) tipe file/tabel/database, berkaitan dengan logic penyimpanan data yang dapat berupa file atau semacam database relational, dan (5) tipe interface eksternal. Berkaitan dengan

komunikasi data pada perangkat/mesin yang lain, contohnya adalah membuat aplikasi SMS Server yang membutuhkan. Koneksi pada perangkat keras Modem telepon adalah proses melakukan perhitungan untuk mendapat nilai *function point* dari perangkat lunak yang akan dibangun.

Rumus FP = CFP x (0.65 + 0.01 x RCAF), Angka 0.65 dan 0.01 adalah ketetapan atau konstanta yang dibuat oleh *Internasional Function Point User Group* (IFPUG).

Tabel 3. Ukuran Sistem FP

LINES	Ukuran
0 .. 9999	Kecil
0.000 .. 49.999	Menengah
50.000 .. 99.999	Semi-besar
100.000 .. 499.999	Besar
500.000 ..	Sangat besar

HASIL DAN PEMBAHASAN

Tahapan ini bertujuan untuk mengetahui komponen pelaksanaan uji PL Aplikasi Kode Diagnosa dan Tindakan *Diseases and Procedure of eye and adnexa*, sesuai dengan design sistem informasi yang digunakan dalam bentuk menu search kode diagnosa dan tindakan. Dengan komponen uji PL dalam bentuk metode uji: FP (RCAF dan CFP) dan Normalisasi data. Bagian metode tersebut dikelompokkan ke dalam RCAF sesuai dengan empat belas modul uji yang dianalisa dengan menggunakan metode normalisasi data.

Tabel 4. Hasil Nilai RCAF

No. Uji	Jenis Variabel Pengujian	Subjek Pengujian	Keterangan Variabel Pengujian	Nilai
1	X1	Tingkat kompleksitas kehandalan backup/recovery	Tidak Pengaruh	0
2	X2	Tingkat kompleksitas komunikasi data	Rata-rata	3
3	X3	Tingkat kompleksitas pemrosesan terdistribusi	Moderat	2
4	X4	Tingkat kompleksitas kebutuhan akan kinerja	Insidental	1
5	X5	Tingkat kebutuhan lingkungan operasional	Insidental	1
6	X6	Tingkat kebutuhan knowledge pengembang	Tidak Pengaruh	0
7	X7	Tingkat kompleksitas updating file master	Tidak Pengaruh	0
8	X8	Tingkat kompleksitas instalasi	Tidak Pengaruh	0
9	X9	Tingkat kompleksitas aplikasi input, output, query online dan file	Moderat	2
10	X10	Tingkat kompleksitas pemrosesan data	Moderat	2
11	X11	Tingkat ketidakmungkinan penggunaan kembali dari kode (reuse)	Rata-rata	3
12	X12	Tingkat variasi organisasi pelanggan	Moderat	2
13	X13	Tingkat kemungkinan perubahan/fleksibilitas	Insidental	1
14	X14	Tingkat kebutuhan kemudahan penggunaan	Essential	5

Aplikasi Kode Diagnosa dan Tindakan

Hasil analisa tabel 4 nilai RCAF sesuai dengan jenis variabel pengujian menunjukkan X1,X6,X7, dan X8 memiliki keterangan variabel pengujian Tidak pengaruh (tidak memiliki pengaruh terhadap tabel uji (RCAF) dikarenakan memiliki bobot nilai kompleksitas = 0 X2 dan X11 memiliki keterangan variabel rata-rata bobot kompleksitas rata-rata penilaian = 3 sesuai dengan keterangan_subjek pengujian tabel 2.3 X4, X5 dan X13 memiliki keterangan variabel pengujian Insidental (Sistem memiliki kepastian informasi baik di lingkungan operasional prosedural dan manajemen) dikarenakan memimiliki bobot nilai kompleksitas = 1 X3,X9,X10 dan X12 memiliki keterangan Moderat (tidak memiliki ukuran kecil maupun besar dalam suatu ukuran jumlah, derajat, dan kekuatan) keterangan variabel moderat memiliki nilai kompleksitas = 2X14 memiliki keterangan_variabel pengujian Essential (Apakah spesifikasi aplikasi didesain, dikembangkan dan didukung untuk memfasilitasi perubahan dan kemudahan penggunaan oleh user) keterangan variabel Essential memiliki nilai kompleksitas = 5.

Tabel 5. Crude Function Points (CFP)

Komponen Sistem Software	Level kompleksitas									Total CFP
	Sederhana			Menengah			Kompleks			
	Count	Faktor Bobot	Point	Count	Faktor Bobot	Point	Count	Faktor Bobot	Point	
	A	B	C=AxB	D	E	F=DxE	G	H	I=GxH	J=C+F+I
Input	2	3	6	3	4	12	-	6	-	18
Output	2	4	8	3	5	15	-	7	-	23
Query Online	-	3	-	-	4	-	-	6	-	-
File logic	-	7	-	-	10	-	-	15	-	-
Interface Eksternal	-	5	-	3	7	21	-	10	-	21
Total CFP										62

Metode uji FP yang digunakan ini mengevaluasi RCAF dan CFP untuk mengetahui faktor pengubah kompleksitas relatif/relative complexity adjustment factor dari 14 (empat belas) subyek yang diuji dan hasil yang diuji dengan menghasilkan nilai RCAF = 22 dan nilai CFP = 62 total nilai FP = 53,94 nilai FP yang dihasilkan ini digunakan untuk mengetahui Normalisasi datanya dengan menggunakan tabel RCAF sehingga akan diketahui jumlah subjek yang bisa digunakan untuk diteliti lebih lanjut. Sebagai langkah akhir, dengan menggunakan persamaan (2) maka dapat dihitung nilai dari Function Point dari Aplikasi Kode Diagnosa dan Tindakan Diseases and Procedure of eye and adnexa ini, yaitu: $FP = CFP \times (0.65 + 0.01 \times RCAF) = 62 \times (0.65 + 0.01 \times 22) = 53,94$ sesuai dengan Tabel 2.3 ukuran sistem FP menurut teori (Gorla dan Benander) nilai FP yang dihasilkan 53,94 memiliki ukuran line 0.9999 dengan ukuran kecil sehingga Aplikasi Kode Diagnosa dan Tindakan Diseases and Procedure of eye and adnexa, memiliki level kompleksitas Aplikasi Sederhana.

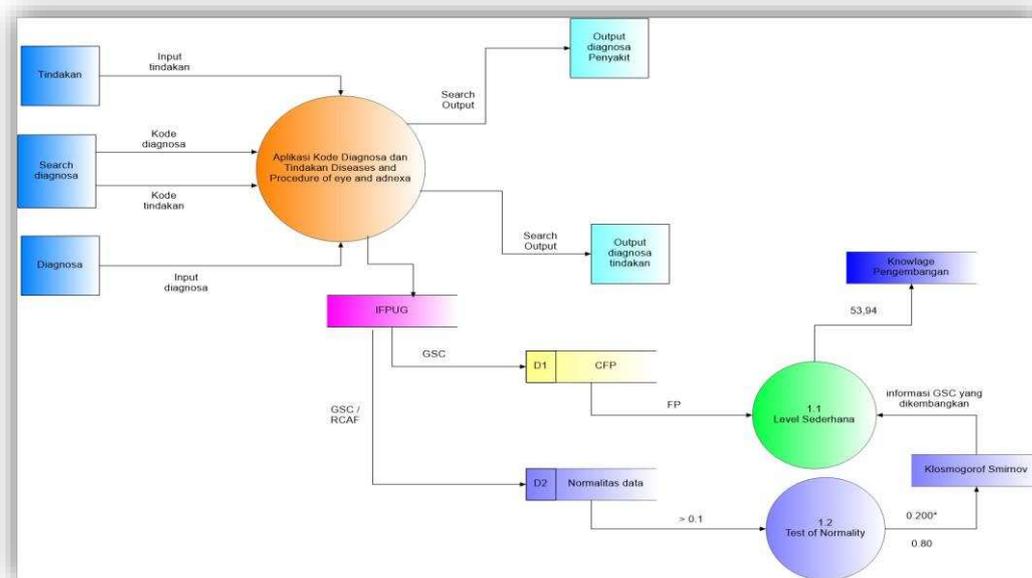
Tabel 6. Hasil Uji Normalitas Data

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
nilai	,170	14	,200 [*]	,890	14	,080

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Uji normalitas data ini dilakukan untuk mengetahui nilai subjek yang dihasilkan dari tabel RCAF yang memiliki fungsi untuk pengembangan Aplikasi kode dan tindakan di RS UNISMA Malang. Sehingga peneliti dapat mengetahui gambaran sistem yang dibuat ketika sistem Aplikasi ini digunakan oleh user dalam konteks organisasi atau unit pelaksana di RS UNISMA. Data uji RCAF dapat dilihat di kolom Sig 0.80. Jika sig atau p lebih dari 0.1 maka kita simpulkan hipotesis nol gagal ditolak, yang berarti data RCAF yang diuji memiliki distribusi yang tidak berbeda dari data yang normal. Atau dengan kata lain data yang diuji memiliki distribusi normal.



Gambar 8. DAD Hasil Uji FP

Tabel 5 hasil FP menunjukkan DAD dari pelaksanaan uji FP memiliki tiga terminator data sumber, yaitu terminator data sumber tindakan, *search* diagnosa, dan diagnosa masing-masing terminator menghasilkan data sumber yang terdiri atas (1) terminator tindakan menghasilkan data sumber input tindakan; (2) terminator *search* diagnosa menghasilkan data sumber kode diagnosa dan kode tindakan; (3) terminator diagnosa menghasilkan data sumber input diagnosa. Ketiga terminator tersebut di proses ke dalam aplikasi kode diagnosa dan tindakan dengan menghasilkan *output* dalam bentuk terminator data tujuan, yaitu (1) *output* diagnosa penyakit dan (2) *output* diagnosa tindakan. Proses aplikasi kode diagnosa dan tindakan membuat koneksi penghubung ke *source* penyimpanan IFPUG dengan informasi GSC ke D1 CFP sebagai data sumber ke proses 1.1. Level sederhana, dengan data tujuan menghasilkan nilai 53,94 sebagai *output* keluaran ke *knowledge* pengembang, IFPUG juga menghasilkan GSC/RCAF ke D2 normalitas data dengan sig >0.1 sebagai data sumber ke proses 1.2. Test of normality dengan sig 0.200* menggunakan uji Klomogorov-Smirnov sebagai *output* ke informasi GSC sebagai pengembangan Aplikasi kode diagnosa dan tindakan oleh pengguna sistem di RSI UNISMA Malang.

SIMPULAN

Berdasarkan hasil dan Pembahasan Aplikasi Kode Diagnosa dan Tindakan *Diseases and Procedure of eye and adnexa* tersebut memiliki tiga terminator data sumber, yaitu *search* diagnosa, diagnosa, dan tindakan dengan input data dari data sumber yang terproses ke Aplikasi Coding. Dalam bentuk *output* diagnosa penyakit dan *output* tindakan penyakit sehingga menghasilkan data RCAF dengan hasil uji 53,94 data normalisasi menghasilkan nilai > 0.1 Test of Normality 1 variabel < 0.1 untuk uji stem an leaf plot untuk data menempel pada garis Normal Q-Q Plots dengan hasil Defrended Normal Q-Q Plots dengan banyak titik yang tersebar jauh dari garis. Hasil evaluasi PL menyatakan Aplikasi Kode Diagnosa dan Tindakan *Diseases and Procedure of eye and adnexa* memiliki level derajat kompleksitas sederhana menengah.

Saran pengembangan sistem ke depan, setelah melakukan uji PL bagi RS UNISMA, meliputi (1) melakukan evaluasi kebutuhan sistem informasi terlebih dahulu dengan melakukan analisa evaluasi kebutuhan sistem informasi kode diagnosa dan tindakan, di level *enterprise* data dan *entity* data informasi, (2) mengevaluasi sistem informasi kode dan tindakan dengan menggunakan tabel RCAF sesuai dengan standar IEEE, (3) memberikan gambaran PL untuk pengembangan kode dan tindakan sesuai dengan struktur data yang dibutuhkan, (4) menghasilkan *output* kode dan tindakan sesuai dengan tabel CFP derajat kompleksitas PL Aplikasi kode dan tindakan, dan (5) mengurangi kesalahan dalam menganalisa hasil evaluasi PL untuk pengembangan PL sesuai dengan kebutuhan.

DAFTAR RUJUKAN

- Burch, J & Gary Grudnitski.1986. *Information Systems Theory and Practice*. New York: John Wiley & Sons.
 Caldiera, G., Antonioli, G., Fiuterm, R & Lokan, C. 1998. *Definition and Experimental Evaluation of Function Points for Object-Oriented Systems*. Proceedings of The Fifth International Software Metrics Symposium, California, US.
 Fathoni. 2009. Pengukuran Kualitas Perangkat Lunak Berdasarkan Kompleksitas Menggunakan Metode Function Point. Palembang: *Jurnal Sistem Informasi*, 1 (2):79—87.

- Gramus, D. dan Herron, D.1996. *Measuring the Software Process – A Practical Guide to Functional Measurements*, New Jersey: Yourdon Press.
- Pratiwi, D. 2013. Implementation of Function Point Analysis in Measuring the Volume Estimation of Software System in Object Oriented and Structural Model of Academic System. Jakarta: *International Journal of Computer Applications*, 70 (10):0975—8887.
- Trisnanto, P.Y. 2016. Perancangan Sistem Informasi Laboratorium Komputer pada Program Studi D-III PMIK Poltekkes Kemenkes Malang. *Jurnal Pendidikan: Teori, Penelitian, dan Pengembangan*. (Online), 1 (11):2152—2157, (<http://journal.um.ac.id/index.php/jptpp/article/view/8007/3650>, diakses 10 April 2017).