

Pengukuran Sistem Informasi Menggunakan Metode *Function Point Analysis* (Studi Kasus: *Software FrontAccounting*)

Information System Measurement Using Function Point Analysis Method (Case Study: FrontAccounting Software)

Natasha Christita Setiyono¹, Bisma Nanda Satria², Soetam Rizky Wicaksono³

^{1,2,3}Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Ma Chung

321910017@student.machung.ac.id¹, 321910016@student.machung.ac.id², soetam.rizky@machung.ac.id³

Received: 07.10.2021	Revised: 15.10.2021	Accepted: 21.01.2021	Available online: 31.10.2021
-------------------------	------------------------	-------------------------	---------------------------------

Abstrak: FrontAccounting merupakan *software* sistem akuntansi berbasis web-multi lokasi, multi-pengguna, multi-bahasa, multi-mata uang, dan manajemen stok yang bersifat *open-source* yang dikembangkan oleh tim internasional, senior *software developer* Joe Hunt dari Swedia dan Janusz Dobrowolski yang berada di Polandia, sementara yang lainnya berasal dari AS, Kanada, Mesir, Dubai, India, Australia, dan tempat lainnya. *Open-source software* yang memiliki fitur yang mirip serta *platform* yang sama dengan FrontAccounting adalah SQL-Ledger, berlisensi gratis dibawah GNU General Public. Dari hasil analisis kelengkapan fitur yang mendukung fungsional proses bisnis penjualan dan distribusi di Mini Market 7 Putra, diperoleh hasil bahwa *software* akuntansi yang paling cocok untuk digunakan adalah FrontAccounting Selain itu pada perusahaan startup pertanian di India, Farm Basket, FrontAccounting sukses diadopsi sebagai sistem informasi digital yang membantu manajemen bisnis yang efektif. Mulai dari penjualan dan pemantauan kegiatan pembelian, manajemen keuangan hingga pengelolaan hubungan pelanggan dari berbagai unit operasional yang terintegrasi ke dalam satu sistem. Dalam penelitian ini, akan dilakukan pengukuran terhadap *software* FrontAccounting dengan menggunakan metode *Function Point Analysis* (FPA). Hasil perhitungan yang akan didapatkan dari penerapan metode FPA dapat digunakan untuk mengembangkan *software* tersebut.

Kata kunci: FrontAccounting, *Function Point Analysis*, *open-source*, Pengukuran, *software*

Abstract: FrontAccounting is an *open-source*, multi-location, multi-user, multi-language, multi-currency, and stock management web-based accounting system software developed by an international team, senior software developers Joe Hunt from Sweden and Janusz Dobrowolski based in in Poland, while others are from the US, Canada, Egypt, Dubai, India, Australia and elsewhere. *Open-source software* that has similar features and the same platform as FrontAccounting is SQL-Ledger, licensed for free under the GNU General Public License. From the analysis of the completeness of the features that support the functional sales and distribution business processes at Mini Market 7 Putra, the results show that the most suitable accounting software to use is FrontAccounting. In addition, at an agricultural startup company in India, Farm Basket, FrontAccounting has been successfully adopted as a digital information system. which helps in effective business management. Starting from sales and purchasing activities monitoring, financial management to customer relationship management from various operational units integrated into one system. In this study, measurements of the FrontAccounting software will be carried out using the *Function Point Analysis* (FPA) method. The calculation results that will be obtained from the application of the FPA method can be used to develop the software.

Keywords: FrontAccounting, *Function Point Analysis*, *open-source*, Pengukuran, *software*

1. PENDAHULUAN

FrontAccounting merupakan *software* sistem akuntansi berbasis web-multi lokasi, multi-pengguna, multi-bahasa, multi-mata uang, dan manajemen stok yang bersifat *open-source* yang dikembangkan oleh tim internasional, senior *software developer* Joe Hunt dari Swedia dan Janusz Dobrowolski yang berada di Polandia, sementara yang lainnya berasal dari AS, Kanada, Mesir, Dubai, India, Australia, dan tempat lainnya. Rilis pada tahun 2004 dengan nama OpenAccounting kemudian pada tahun 2005 berganti nama menjadi FrontAccounting. Saat ini FrontAccounting support di Mac OS X, Linux, Unix, dan Microsoft Windows ^[3](ERP, 2015). *Open-source software* yang memiliki fitur yang mirip serta *platform* yang sama (*web-based*) dengan FrontAccounting adalah SQL-Ledger, berlisensi gratis dibawah GNU General Public ^[1](Agittins, 2007). SQL-Ledger support pada Mac OS X, Unix, dan Microsoft Windows ^[13](SQL-Ledger, 2021). SQL-Ledger memiliki banyak fitur penting, seperti *account receivable*, *account payable*, *inventory control*, *billing/invoicing*, *payroll*, *purchase/sales order*, *vendors*, dll. SQL-Ledger dapat menggunakan MySQL dan PostgreSQL sebagai sistem manajemennya. Namun, SQL-

Ledger tidak memiliki fitur *Supply Chain Management (SCM)* seperti FrontAccounting^[10](Sampaio & Bernardino, 2016a).

Pada penentuan *Enterprise Resource Planning (ERP)* sistem pembelian, penjualan, dan inventory barang di Mini Market 7 Putra, dilakukan perbandingan *software* akuntansi gratis, antara lain Openbravo, FrontAccounting, dan Compiere. Dari hasil analisis kelengkapan fitur yang mendukung fungsional proses bisnis penjualan dan distribusi di Mini Market 7 Putra, diperoleh hasil bahwa *software* akuntansi yang paling cocok untuk digunakan adalah FrontAccounting^[8](Nofriandi & Kamil, 2015). Selain itu pada perusahaan *startup* pertanian di India, Farm Basket, FrontAccounting sukses diadopsi sebagai sistem informasi *digital* yang membantu manajemen bisnis yang efektif. Mulai dari penjualan dan pemantauan kegiatan pembelian, manajemen keuangan, manajemen rantai pasok, kegiatan manufaktur, perencanaan sumber daya, manajemen sumber daya manusia hingga pengelolaan hubungan pelanggan dari berbagai unit operasional yang terintegrasi ke dalam satu sistem. Singkatnya, dengan pengintegrasian ERP menggunakan FrontAccounting pada industri ini telah meningkatkan kinerja bisnis yang signifikan karena dapat mengurangi pengeluaran uang, lebih menghemat waktu dan mengurangi pemborosan sumber daya^[15](Y. Praneetha & Anu Choudhary, 2020).

Dengan berhasilnya pemanfaatan FrontAccounting oleh perusahaan berskala kecil hingga menengah dan juga dengan fitur-fitur penting yang dimiliki seperti: (1) *Purchase order*; (2) *Goods receivable notes*; (3) *Billing/invoicing*; (4) *Supplier*; (5) *Supplier invoices/credit notes*; (6) *Payments*; (7) *Deposits*; (8) *Sales*; (9) *Journal Entry*; (10) *Profit and Loss Drilldown, etc*^[11](Sampaio & Bernardino, 2016b). Maka *open-source software* FrontAccounting dinilai layak dijadikan studi kasus untuk melakukan pengukuran.

2. TINJAUAN PUSTAKA

2.1. Sistem Informasi

Sistem Informasi merupakan interaksi antara data, manusia, dan prosedur untuk menghasilkan penyelesaian berupa informasi yang berguna untuk mengambil keputusan jangka pendek, jangka menengah maupun jangka panjang dalam sebuah organisasi. Dapat dikatakan sistem informasi juga merupakan sebuah pendukung kegiatan operasi, manajerial, dan pengambilan keputusan strategis suatu organisasi^[14](Yulianto et al., 2009).

2.2. Pengukuran Sistem Informasi

Pengukuran perangkat lunak merupakan suatu metode yang ampuh untuk mengelola perangkat lunak, mengizinkan penerapan prinsip-prinsip rekayasa pengembangan perangkat lunak, dan memberikan tujuan kuantitatif untuk keputusan manajemen^[2](Buglione et al., 2015). Pengukuran sangat penting untuk setiap kegiatan uji coba. Faktor pengujian/penilaian kualitas perangkat lunak yang dikemukakan oleh McCall, terdiri dari faktor efisiensi, kehandalan, kegunaan, pemeliharaan, dan media pembelajaran^[5](Hapsari & Husen, 2015).

2.3. Function Point Analysis (FPA)

Function Point Analysis adalah salah satu metode pengukuran perangkat lunak yang menggunakan pendekatan tidak langsung dengan memperkirakan ukuran perangkat lunak ke dalam satuan *Function Point*. Nilai *Function Point* didapatkan dari perkalian antara *Unadjusted Function Point* dengan nilai *Technical Complexity Adjustment*. Kemudian hasil dari perkiraan *Function Point* akan digunakan dalam menghitung *Line of Code* yang didapatkan dari hasil perkalian nilai *Function Point* dengan nilai *productivity faktor* bahasa pemrograman yang digunakan dalam pengembangan aplikasi^[12](Saptono & Anggrainingsih, 2016).

2.4. FrontAccounting

FrontAccounting merupakan *software* sistem akuntansi berbasis web-multi lokasi, multi-pengguna, multi-bahasa, multi-mata uang, dan manajemen stok yang bersifat *open-source* yang dikembangkan oleh tim internasional, senior *software developer* Joe Hunt dari Swedia dan Janusz Dobrowolski yang berada di Polandia, sementara yang lainnya berasal dari AS, Kanada, Mesir, Dubai, India, Australia, dan tempat lainnya. Rilis pada tahun 2004 dengan nama OpenAccounting kemudian pada tahun 2005 berganti nama menjadi FrontAccounting. Penggunaan FPA dalam penelitian ini dinilai tepat karena hasil pengukuran yang didapatkan

akan memberikan informasi yang dapat digunakan untuk mengembangkan aplikasi tersebut^[4](FrontAccounting, 2020).

3. METODE

Pengukuran yang dilakukan pada FrontAccounting kali ini menggunakan metode *Function Point Analysis* (FPA) yang diterbitkan *International Function Point User Group* (IFPUG) karena metode ini memiliki nilai keakuratan yang tinggi dan populer di kalangan *developer*. Terdapat tiga tahapan yang harus dilakukan untuk mengukur perangkat lunak, antara lain: (1) Mengkalkulasi adalah untuk menghitung bobot nilai dari komponen-komponen *Function Point* yang dikaitkan dengan software yang akan dibuat, atau yang biasa disebut dengan *Crude Function Points* (CFP); (2) Menghitung *Relative Complexity Adjustment Factor* (RCAF) atau dalam Bahasa Indonesia dinamakan Faktor Penyesuaian Kompleksitas Relatif; (3) Setelah dilakukan penghitungan terhadap CFP dan RCAF baru bisa dilakukan penghitungan terhadap *Function Point* (FP) dengan rumus^[9](Pp_pankaj, 2020):

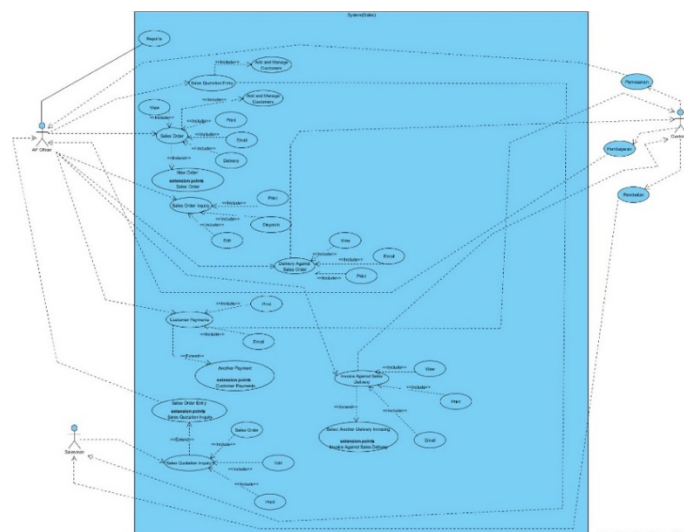
$$FP = CFP * (0.65 + 0.01 * RCAF)$$

4. HASIL DAN PEMBAHASAN

Hasil dari analisis *software* akuntansi FrontAccounting, maka didapatkan diagram *use case* sebagai berikut.

Tabel 1. Modul FrontAccounting

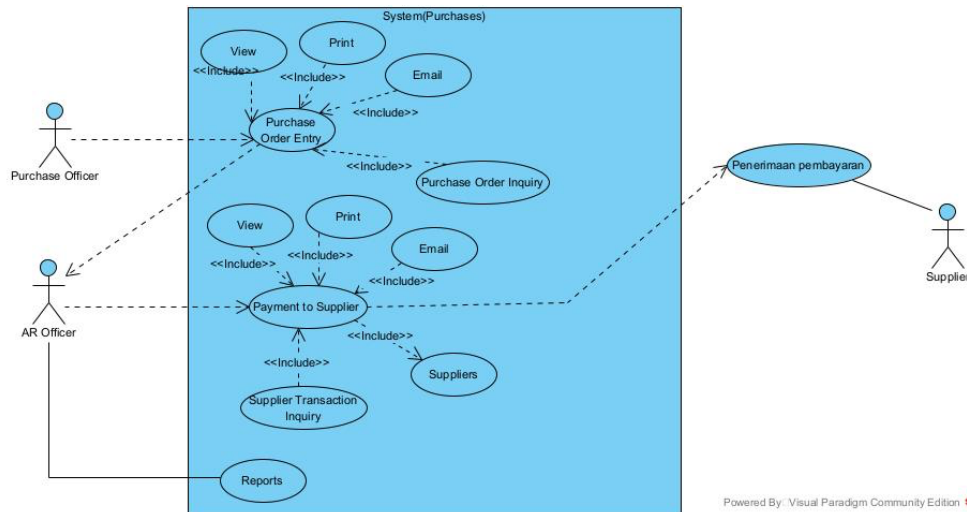
Sales	Purchase	Banking & General Ledger
Sales Quotation Entry	Purchase Order Entry	Payments
Sales Order Entry	Purchase Order Inquiry	Deposits
Delivery Against Sales Order	Supplier Transaction Inquiry	Reports
Invoice Against Sales Delivery	Reports	Journal Entry
Customer Payment	Suppliers	Journal Inquiry
Reports	Payments to Suppliers	GL Inquiry
Add & Manage Customer		Trial Balance
Sales Quotation Inquiry		Balance Sheet Drilldown
Sales Order Inquiry		Profit and Loss Drilldown
		Bank Account Inquiry
		Bank Account



Gambar 1. Sales Use Case

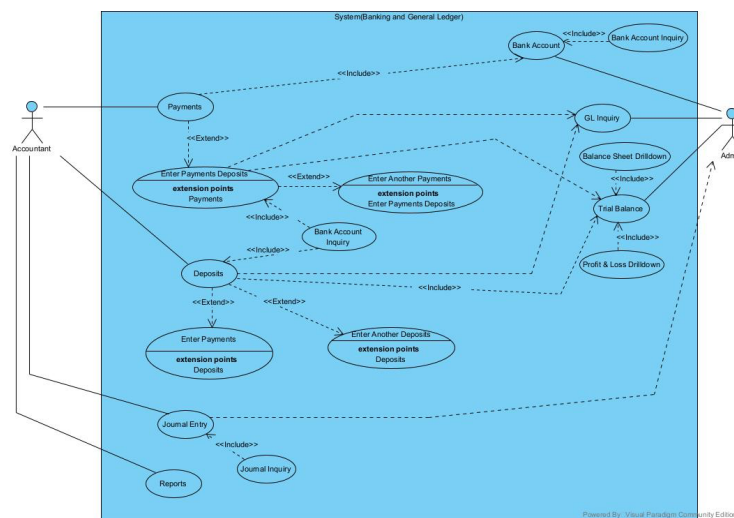
Ketika *customer* melakukan pemesanan maka *AP Officer* akan menginput master *Customer* pada *Add and Manage Customers* lalu menginput pemesanan *customer* pada *Sales Quotation Entry* yang akan diteruskan ke *Salesman*. Ketika *customer* melakukan pembelian maka *Salesman* mengakses pemesanan melalui *Sales Quotation Inquiry* dan meneruskannya ke *sales order* atau menginput pembelian *Sales Order Entry*. Input dari *Sales Order Entry* masuk ke *Sales*

Order Inquiry akan diteruskan ke *AP Officer* untuk disetujui untuk dilakukan pembelian melalui *Sales Order*. *Input* dari *Sales Order* akan masuk ke *Delivery Against Sales Order* untuk diproses *AP Officer*, setelah diproses maka *AP Officer* akan menyalurkan dokumen pembelian & pengiriman melalui *Print/Email* ke *Customer*. Setelah *Customer* melakukan pembayaran, *AP Officer* menginput pembayaran *Customer* melalui *Customer Payments*, maka *AP Officer* akan melakukan perintah *dispatch* pada *Delivery Against Sales Order*. Transaksi yang telah di-*dispatch* akan masuk ke *Invoice Against Sales Delivery* lalu *AP Officer* mengirimkan *invoice* transaksi ke *Customer* melalui *Print/Email*.



Gambar 2. Purchase Use Case

Ketika telah selesai dilakukan pemesanan produk pada *Supplier*, maka akan muncul data pembelian beserta harga-harganya. Pada awalnya, *AR Officer* akan menginputkan mastering *Supplier*. Setelah data *Supplier* tersimpan, *Purchase Officer* dapat mengakses *Supplier* yang sudah ditambahkan ke dalam *database Supplier* dan menginputkan data pembelian tersebut ke dalam *Purchase Order Entry*. Setelah di-save, maka data yang sudah diinputkan di *Purchase Order Entry* akan masuk ke dalam data *Purchase Order Inquiry*. Data yang sudah diinputkan di *Purchase Order Entry* juga dapat diteruskan kepada *AR Officer* untuk diproses ke dalam pembayaran, yaitu di *Payment to Supplier*. Lalu, setelah *AR Officer* memasukkan data pembayaran ke dalam *Payment to Supplier*, data akan tersimpan pada *Supplier Transaction Inquiry* dan akan diperoleh hasil berupa *remittance*/bukti pengiriman uang yang akan diberikan kepada *Supplier*. Data seperti *Purchase Orders* dan *Payment* akan tersimpan secara otomatis ke dalam *Reports* dan juga data *remittance* akan dapat dilihat kembali di *Remittance Report*.



Gambar 3. Banking and General Ledger Use Case

Proses pertama kali yang dilakukan adalah *Admin* menginputkan data *Bank Account*. *Admin* mengisikan data dari rekening bank yang digunakan untuk segala proses administrasi. Kemudian, *Accountant* dapat menginputkan *Deposit*. Setelah *Deposit* diproses, *Accountant* dapat melakukan pembayaran baik kepada *customer* maupun *supplier* di menu *Payment*. Data *Payment* yang berhasil diinputkan dan di *save* akan langsung tersimpan otomatis ke dalam *Journal Inquiry*. Namun, *Accountant* tidak dapat mengakses detail jurnal karena tidak ada izin untuk *Accountant* mengakses. Akses untuk melihat detail dari setiap *Journal Inquiry* hanya dimiliki oleh *Admin*. Untuk penginputan data Jurnal dapat dilakukan secara manual di dalam fitur *Journal Entry*. Berdasarkan hasil input baik di *Deposit* dan *Payment* akan terakumulasi secara otomatis sehingga menghasilkan *Balance Sheet Drilldown*, *Trial Balance*, dan *Profit & Loss Drilldown*. Semua data mulai dari *Deposit*, *Payment*, *Bank Account*, *Journal Inquiry*, *Balance Sheet Drilldown*, *Trial Balance*, dan *Profit & Loss Drilldown* akan masuk secara otomatis ke dalam *Reports*.

Tahap Melakukan Pengukuran

1. Menghitung *Crude Function Point*.

Crude Function Points (CFP) adalah tahap untuk menghitung bobot nilai dari komponen-komponen *Function Point* dari *software* FrontAccounting ^[6](Khairani, 2015). Dari hasil perhitungan, diperoleh tabel sebagai berikut.

Tabel 2. Keterangan Rasio Kompleksitas

<i>Simple</i>	<i>Average</i>	<i>Complex</i>
< 5 input	5-7 input	> 7 input

Tabel 3. External Input (EI)

No.	Modul	Deskripsi	Kompleksitas
1.	Sales	Sales Quotation Entry	Average
2.		Sales Order Entry	Average
3.		Delivery Against Sales Order	Simple
4.		Invoice Against Sales Delivery	Simple
5.		Customer Payment	Average
6.		Reports(Sales Summary Report, Price Listing, Salesman Listing, Print Invoices, Print Deliveries, Print Sales Order, Print Sales Quotations, Print Receipts)	8 Simple
7.	Purchase	Add & Manage Customer	Complex
8.		Sales Quotation Inquiry	Simple
9.		Sales Order Inquiry	Simple
10.		Purchase Order Entry	Average
11.		Purchase Order Inquiry	Simple
12.		Supplier Transaction Inquiry	Simple
13.	Banking & General Ledger	Reports (Payment Report, Print Purchase Orders, Print Remittances)	3 Simple
14.		Suppliers	Complex
15.		Payments to Suppliers	Complex
16.		Payments	Average
17.		Deposits	Average
18.		Reports (Bank Statement, List of Journal Entries, Annual Expense Breakdown, Balance Sheet, Profit and Loss Statement, Trial Balance)	6 Simple
19.		Journal Entry	Average
20.		Journal Inquiry	Average
21.		GL Inquiry	Average
22.		Trial Balance	Average
23.		Balance Sheet Drilldown	Average
24.		Profit and Loss Drilldown	Average
25.		Bank Account Inquiry	Average
26.		Bank Account	Average

Tabel 4. External Output (EO)

No.	Modul	Deskripsi	Kompleksitas
1.	Sales	Sales Quotation Entry	2 Simple & 2 Average
2.		Sales Order Entry	2 Simple & 2 Average
3.		Delivery Against Sales Order	Simple & Average
4.		Invoice Against Sales Delivery	Simple & Average
5.		Customer Payment	Simple & Average
6.		Reports(Sales Summary Report, Price Listing, Salesman Listing, Print Invoices, Print Deliveries, Print Sales Order, Print Sales Quotations, Print Receipts)	8 Average
7.		Add & Manage Customer	-
8.		Sales Quotation Inquiry	Simple & Complex
9.		Sales Order Inquiry	Simple & Complex
10.	Purchase	Purchase Order Entry	5 Average
11.		Purchase Order Inquiry	4 Average
12.		Supplier Transaction Inquiry	6 Average
13.		Reports (Payment Report, Print Purchase Orders, Print Remittances)	2 Average & 1 Complex
14.		Suppliers	-
15.	Banking & General Ledger	Payments to Suppliers	5 Average
16.		Payments	2 Simple
17.		Deposits	2 Simple
18.		Reports (Bank Statement, List of Journal Entries, Annual Expense Breakdown, Balance Sheet, Profit and Loss Statement, Trial Balance)	2 Average & 4 Complex
19.		Journal Entry	2 Simple
20.		Journal Inquiry	2 Simple
21.		GL Inquiry	Complex
22.		Trial Balance	Complex
23.		Balance Sheet Drilldown	Average
24.		Profit and Loss Drilldown	Average
25.		Bank Account Inquiry	Complex
26.		Bank Accounts	2 Simple

Tabel 5. Internal Logical File (ILF)

No.	Modul	Deskripsi	Kompleksitas
1.	Sales	Sales Quotation Entry	Complex
2.		Sales Order Entry	Complex
3.		Delivery Against Sales Order	Complex
4.		Invoice Against Sales Delivery	Complex
5.		Customer Payment	Average
6.		Reports(Sales Summary Report, Price Listing, Salesman Listing, Print Invoices, Print Deliveries, Print Sales Order, Print Sales Quotations, Print Receipts)	6 Simple & 2 Average
7.		Add & Manage Customer	2 Average & 3 Complex
8.		Sales Quotation Inquiry	Complex
9.		Sales Order Inquiry	Complex
10.	Purchase	Purchase Order Entry	Complex
11.		Purchase Order Inquiry	Average
12.		Supplier Transaction Inquiry	Complex
13.		Reports (Payment Report, Print Purchase Orders, Print Remittances)	3 Simple
14.		Suppliers	Simple, 2 Average, & 2 Complex
15.		Payments to Suppliers	Simple
16.	Banking & General Ledger	Payments	Simple

17.	Deposits	Simple
18.	Reports (Bank Statement, List of Journal Entries, Annual Expense Breakdown, Balance Sheet, Profit and Loss Statement, Trial Balance)	3 Simple & 3 Average
19.	Journal Entry	Complex
20.	Journal Inquiry	Average
21.	GL Inquiry	Simple
22.	Trial Balance	Average
23.	Balance Sheet Drilldown	Simple
24.	Profit and Loss Drilldown	Average
25.	Bank Account Inquiry	Complex
26.	Bank Accounts	Complex

Tabel 6. External Interface File (EIF)

No.	Modul	Deskripsi	Kompleksitas
1.	Sales	Sales Quotation Entry	Average
2.		Customer Payment	Average
3.	Purchase	Purchase Order Entry	Average
4.		Direct GRN	Average
5.		Direct Supplier Invoice	Average
6.		Suppliers	Complex
7.	Banking & General Ledger	Bank Account	Average
8.		Deposit	Simple

Tabel 7. External Inquiry (EQ)

No.	Modul	Deskripsi	Kompleksitas
-	-	-	-

Tabel 8. Perhitungan Complexity Level (CFP)

User Function Types	Complexity Level									Total
	Simple			Average			Complex			
	Count	Weight factor	Points	Count	Weight factor	Points	Count	Weight factor	Points	
	A	B	C=A×B	D	E	F=D×E	G	H	I=G×H	
External Input (EI)	23	2	46	14	5	70	3	7	21	137
External Output (EO)	19	3	57	41	4	164	10	5	50	271
Internal Logical File (ILF)	19	4	76	13	5	65	16	6	96	237
External Interface File (EIF)	1	2	2	6	3	18	1	4	4	24
External Inquiry (EQ)	0	5	0	0	7	0	0	9	0	0
CFP Total										669

2. Menghitung Relative Complexity Adjustment Factor.

Relative Complexity Adjustment Factor (RCAF) digunakan untuk menghitung bobot kompleksitas dari *software* FrontAccounting berdasarkan 14 Karakteristik Umum Sistem ^[7](Kristi, Aisah, & Dewi, 2020). RCAF memiliki skala penelitian yaitu mulai 0 s/d 5, dengan keterangan sebagai berikut.

- 0 = Tidak Berpengaruh
- 1 = Insidental
- 2 = Moderat

3 = Rata-Rata
4 = Signifikan
5 = Esensial

Tabel 2. Perhitungan 14 Karakteristik Umum Sistem

No.	Karakteristik	Bobot
1.	Tingkat Kompleksitas Komunikasi Data	3
2.	Tingkat Kompleksitas Pemrosesan Terdistribusi	1
3.	Tingkat Kompleksitas <i>Performance</i>	5
4.	Tingkat Kompleksitas Konfigurasi	5
5.	Tingkat Frekuensi Penggunaan <i>Software</i>	5
6.	Tingkat Frekuensi <i>Input Data</i>	5
7.	Tingkat Kemudahan Penggunaan Bagi <i>User</i>	3
8.	Tingkat Frekuensi <i>Update Data</i>	4
9.	Tingkat Kompleksitas <i>Processing Data</i>	5
10.	Tingkat Kemungkinan Penggunaan Kembali/ <i>Reusable</i> Kode Program	5
11.	Tingkat Kemudahan Dalam Instalasi	1
12.	Tingkat Kemudahan Operasional <i>Software</i> (<i>Backup, Recovery, dsb</i>)	3
13.	Tingkat <i>Software</i> Dibuat Untuk Multi Organisasi/ <i>Perusahaan/Client</i>	5
14.	Tingkat Kompleksitas Dalam Mengikuti Perubahan/ <i>Fleksibel</i>	5
TOTAL		55

Untuk menghitung *Function Point*, digunakan rumus:

$$FP = CFP \times (0.65 + 0.01 \times RCAF)$$

Keterangan:

FP = *Function Point*

CFP = *Crude Function Point*

RCAF = *Relative Complexity Adjustment Factor*

0.65 + 0.01 = Konstanta bawaan

Maka;

$$\begin{aligned}\text{Nilai FP} &= 669 \times (0.65 + 0.01 \times 55) \\ &= 802.8\end{aligned}$$

Jadi, nilai atau poin kompleksitas yang dimiliki oleh modul *Sales, Purchase, dan Banking and General Ledger* pada *software* FrontAccounting adalah sebesar 802.8.

5. KESIMPULAN

Berdasarkan penghitungan menggunakan metode *Function Point Analysis (FPA)* pada *software* akuntansi *open-source* FrontAccounting dengan modul sebagai berikut: (1) *Sales*; (2) *Purchase*; dan (3) *Banking and General Ledger*. Dihasilkan nilai *Function Point (FP)* sebesar 802.8 sehingga dapat disimpulkan bahwa *software* ini dinilai kompleks untuk diterapkan penggunaannya ke Usaha Kecil Mikro Kecil Menengah (UMKM). Selain itu *software* FrontAccounting ini memiliki fitur *Maintenance* yang dapat digunakan untuk melakukan pengaturan seperti *Customer, Supplier, dan Bank Account* yang menjadikan *software* FrontAccounting ini layak untuk digunakan.

DAFTAR PUSTAKA

- [1] Agittins. (2007). *When GPL software goes bad*. 10 April. <https://web.archive.org/web/20070618032337/http://www.purple.dropbear.id.au/sql-ledger-license-changed-no-longer-gpl-when-gpl-software-goes-bad/>
- [2] Buglione, L., Abran, A., Von, C. G., Caffery, F. M., Carlo, J., & Hauck, R. (2015). *Software Measurement* (Vol. 230).
- [3] ERP, F. (2015). *History of FrontAccounting*. 8 January. <https://frontaccounting.com/fawiki/index.php?n=Main.HistoryOfFrontAccounting>

- [4] FrontAccounting. (2020). FrontAccounting. Retrieved November 26, 2021, from 20 Maret website: <https://frontaccounting.com/fawiki/>
- [5] Hapsari, R. K., & Husen, M. J. (2015). Estimasi Kualitas Perangkat Lunak Berdasarkan Pengukuran Kompleksitas Menggunakan Metrik Function Oriented. Seminar Nasional Sains Dan Teknologi Terapan III, 425–434.
- [6] Khairani, D. (2015). Studi Kasus Pengukuran Sistem Informasi Menggunakan Function Point (Fp). Jurnal Teknik Informatika, 8(2), 1–7. <https://doi.org/10.15408/jti.v8i2.2442>
- [7] Kristi, J., Aisah, S. N., & Dewi, R. S. (2020). Estimasi Biaya Software FAS (Financing Analysis System) Menggunakan Metode Function Point (Studi Kasus Pada PT BPRS Lantabur Tebuireng). JURIKOM (Jurnal Riset Komputer), 7(1), 97. <https://doi.org/10.30865/jurikom.v7i1.1891>
- [8] Nofriandi, A., & Kamil, H. (2015). Penerapan Enterprise Resource Planning (ERP) pada Sistem Pembelian, Penjualan dan Inventory Barang di Mini Market 7 Putra. In TEKNOSI (Vol. 01, Issue 01).
- [9] Pp_pankaj. (2020). Software Engineering | Calculation of Function Point (FP). Retrieved November 26, 2021, from 28 June website: <https://www.geeksforgeeks.org/software-engineering-calculation-of-function-point-fp/>
- [10] Sampaio, D., & Bernardino, J. (2016a). Open-source accounting software for SMEs. International Journal of Business Information Systems, 23(3), 287–306. <https://doi.org/10.1504/IJBIS.2016.10000349>
- [11] Sampaio, D., & Bernardino, J. (2016b). Open-source accounting software for SMEs. International Journal of Business Information Systems, 23(3), 287–306. <https://doi.org/10.1504/IJBIS.2016.10000349>
- [12] Saptono, R., & Anggrainingsih, R. (2016). Development of Software Size Estimation Application using Function Point Analysis (FPA) Approach with Rapid Application Development (RAD).
- [13] SQL-Ledger. (2021). SQL-Ledger. <https://www.sql-ledger.com/cgi-bin/nav.pl?page=source/index.html&title=Download>
- [14] Yulianto, A. A., Gartina, I., Astuti, R., Dewi, S., Sari, S. K., & Witanti, W. (2009). Analisis dan Desain Sistem Informasi. Politeknik Telkom Bandung, 53(9), 1–176.
- [15] Y. Praneetha, V. V. S. K., & Anu Choudhary, S. K. (2020). Front Accounting: An ERP for Small and Medium Agri Enterprises. International Journal of Current Microbiology and Applied Sciences, 9(3), 3287–3294. <https://doi.org/10.20546/ijcmas.2020.903.377>