



Rancang Bangun Aplikasi Berbasis Android Untuk Mengestimasi Usaha dan Biaya Proyek Pengembangan Perangkat Lunak

Design and Build Android-Based Applications of Software Development Projects Effort and Cost Estimation

Juliana Kristi^{*1}, Renny Sari Dewi¹

¹Program Studi Sistem Informasi, Fakultas Teknologi Informasi dan Kreatif, Universitas Internasional Semen Indonesia

ARTICLE INFO

Article history:

Diterima 02-10-2020
Diperbaiki 17-12-2020
Disetujui 28-12-2020

Kata Kunci:

Estimasi Biaya
Estimasi Perangkat Lunak
Kodular
Function Points
Use Case Points

Keywords:

Cost Estimation
Software Estimation
Kodular
Function Points
Use Case Points

ABSTRAK

Estimasi usaha dan biaya perangkat lunak perlu dilakukan dengan hati-hati dan terukur secara jelas sehingga keberhasilan proyek pengembangan perangkat lunak dapat tercapai. Terdapat 2 metode yang sudah lazim digunakan untuk mengestimasi usaha dan biaya pengembangan perangkat lunak, diantaranya adalah *Function Points* (FP) dan *Use Case Points*. Tahapan penelitian yang dilakukan ada 4, yaitu studi literatur, analisis kebutuhan, perancangan, dan implementasi (pengembangan aplikasi hingga uji coba). Analisis dan perancangan aplikasi menggunakan tools Signavio. Sedangkan pengembangan aplikasinya penulis menggunakan open source *rapid application development* (RAD) yaitu Kodular. Tahap uji coba dilakukan dengan cara melakukan verifikasi terhadap perhitungan manual yang proyeknya sudah selesai dikerjakan. Hasil dari penelitian ini adalah terciptanya aplikasi berbasis Android yang dapat difungsikan dalam mempercepat para pelaku bisnis di bidang pengembangan perangkat lunak dalam melakukan proses estimasi harga produk. Penelitian ini memiliki batasan yaitu belum dilakukan pengujian validitas terhadap proyek pengembangan lunak yang masih dalam tahap perencanaan.

ABSTRACT

Prediction of software business and costs needs to be done carefully and clearly measured so that the success of a software development project can be achieved. There are 2 methods that are commonly used to estimate software development efforts and costs, such *Function Points* (FP) and *Use Case Points*. There were 4 stages of the research carried out: literature study, needs analysis, design, and implementation (application development to testing). Analysis and application design using the Signavio tools. Meanwhile, the author uses open source *rapid application development* (RAD), called Kodular. The trial phase is carried out by verifying manual calculations whose projects have been completed. The result of this research is the creation of an Android-based application that can be used to accelerate business players in the software development sector in the process of estimating product prices. This research has limitation is not tested the validity of the soft development project which is still in the planning stage.

1. Pendahuluan

Sebuah proyek pengembangan perangkat lunak dikatakan berhasil apabila dapat diserahkan dengan tepat waktu sesuai dengan biaya dan kualitas yang diharapkan. Hal tersebut membuktikan bahwa estimasi usaha dan biaya pengembangan perangkat lunak merupakan salah satu penentu keberhasilan [1]. Pada proses pengembangan perangkat lunak, estimasi usaha dan biaya pengembangan perangkat lunak memiliki peranan penting untuk menjaga proses pengembangan tetap

dalam kendali pengembang. Proyek perangkat lunak akan sulit dikendalikan jika sebelumnya tidak memiliki estimasi dan perencanaan. Dalam melakukan estimasi usaha dan biaya pengembangan perangkat lunak diperlukan penilaian kelayakan dari perencanaan proyek dengan cara meninjau nilai effort [2]. Terdapat beberapa metode untuk melakukan estimasi usaha dan biaya pengembangan perangkat lunak yang dikembangkan diantaranya metode *Function Points* dan *Use Case Points*.

Metode *Function Points* (FP) dan *Use Case Points* (UCP) merupakan dua metode estimasi biaya perangkat lunak yang memiliki deviasi yang cukup kecil [3]. Metode *Function Points* cukup populer digunakan di kalangan pengembang karena memiliki nilai akurasi yang tinggi [4]. Metode ini dapat digunakan acuan perhitungan estimasi usaha pengembangan bahkan sampai pengembangan game [5]. Selain itu, metode ini juga dapat digunakan diberbagai sektor perusahaan [6]. Metode *Use Case Points* lebih sering diaplikasikan pada pengembangan perangkat lunak yang berdasarkan *object oriented* [7]. Metode ini dapat diperluas dengan beberapa variasi dengan disesuaikannya pada pengembangan perangkat lunak yang diestimasi [8]. Banyak penelitian mengalami dilema terhadap estimasi biaya perangkat lunak yang semakin berkembang kebutuhannya sehingga sangat sulit memprediksi secara akurat ukuran perangkat lunak yang dikembangkan [9]. Metode *Function Points* dan *Use Case Points* dapat membantu hal tersebut untuk melakukan estimasi biaya berdasarkan kebutuhan perangkat lunak. Metode *Function Points* digunakan bagi perangkat lunak yang tidak memerlukan hasil analisis sistem dalam bentuk narasi menggunakan skenario kasus dan konsep *database*, sedangkan metode *Use Case Points* memerlukan hal tersebut [10]. Untuk mendukung hal tersebut maka dibutuhkan *tools* yang dapat mempermudah dan mempercepat proses perhitungan estimasi usaha dan biaya pengembangan perangkat lunak bagi orang-orang yang tidak mendalami skenario sistem dan *database* perangkat lunak maupun mengetahui ruang lingkup dari masing-masing fungsi pada perangkat lunak.

Di sisi lain, pengembangan aplikasi kini marak dengan pendekatan *rapid application development* (RAD). RAD tools terdiri dari berbagai platform, mulai dari *web-based application*, *desktop*, hingga *mobile*. Justifikasi penggunaan RAD tools adalah kecepatan dalam pengembangan aplikasi yang analisis kebutuhan dan perancangannya sudah lengkap hingga konsep basis datanya. RAD tools yang digunakan dalam penelitian ini adalah Kodular. Kodular merupakan platform *open source* untuk membuat aplikasi Android dengan menggunakan *block programming* berbasis *website*. Kodular memiliki keunggulan dapat menguji/mencoba aplikasi yang telah dibuat secara langsung tanpa harus mengeksplor aplikasi terlebih dahulu.

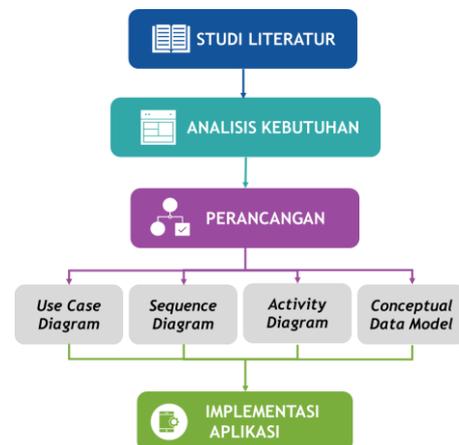
Adapun batasan dari penelitian ini adalah:

- 1) Secara teknis, pengembangan aplikasi Android menggunakan platform basis data MySQL dan webserver lokal XAMPP
- 2) Secara non-teknis, konversi dari satuan usaha (orang/jam) yang dihasilkan dari metode FP dan UCP menjadi satuan biaya (Rupiah) menggunakan tarif standar yang dikeluarkan oleh Kelly Services Indonesia Salary Guide tahun 2020.

Kontribusi yang dihasilkan dalam penelitian ini adalah bersifat praktis. Para profesional di bidang pengembangan perangkat lunak dapat memanfaatkan aplikasi ini sebagai alternatif sekunder dalam penentuan harga produknya.

2. Metode Penelitian

Metode penelitian digunakan sebagai penyelesaian penelitian ini, sehingga proses pengerjaan lebih terarah dan sistematis. Gambar 1 merupakan tahapan metode penelitian yang perlu dilakukan pada saat membangun aplikasi pada penelitian ini.



Gambar 1 Metode penelitian

Secara rinci, penjelasan Gambar 1 dapat dilihat pada poin-poin berikut ini.

2.1 Studi Literatur

Tahapan ini dilakukan untuk mencari teori yang relevan dengan permasalahan yang ditemukan. Adapun input dari tahapan ini adalah referensi berupa artikel ilmiah, laporan atau buku mengenai metode *Function Points*, *Use Case Points*, dan hal-hal yang berkaitan dengan penelitian ini.

2.2 Analisis Kebutuhan

Pada tahapan analisis kebutuhan telah didefinisikan kebutuhan-kebutuhan fungsional aplikasi untuk menentukan fitur yang diperlukan dalam pembangunan aplikasi ini.

2.3 Perancangan

Tahapan ini dilakukan perancangan arsitektur perangkat lunak meliputi pembuatan *use case diagram*, *sequence diagram*, *activity diagram*, dan *conceptual data model*.

2.4 Implementasi Aplikasi

Tahapan ini dilakukan pembangunan aplikasi dengan menggunakan pendekatan blok kode pemrograman khas RAD tools yaitu Kodular. Setelah dikembangkan, aplikasi diujicobakan dengan cara membandingkan antara perhitungan manual yang dilakukan manajer proyek menggunakan MS Excel dengan luaran dari aplikasi estimator pengembangan perangkat lunak berbasis Android ini.

3. Hasil dan Pembahasan

Pada bagian ini akan dijelaskan mengenai hasil dari penelitian ini.

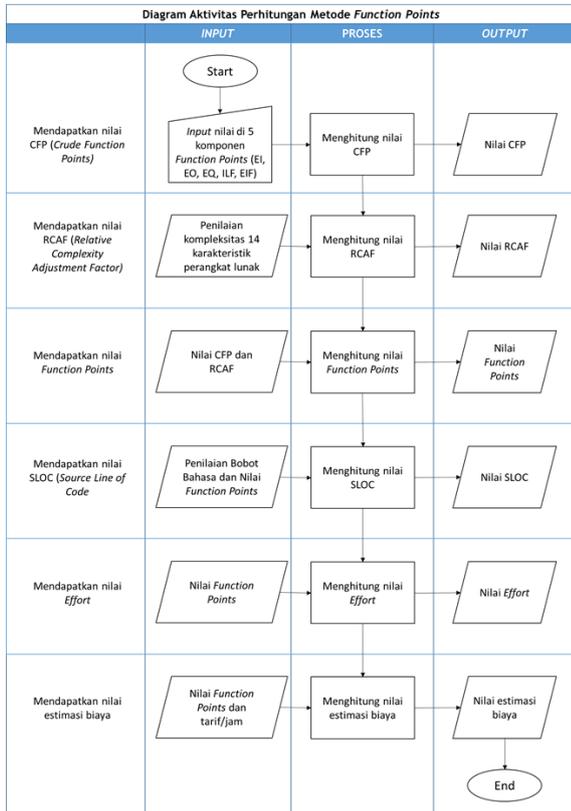
3.1 Alur Proses Perhitungan

Dalam membangun aplikasi ini diperlukan alur proses bisnis perhitungan menggunakan metode *Function Points* dan

Use Case Points untuk mempermudah dalam pembangunan aplikasi sebagai berikut:

3.1.1 Metode Function Points (FP)

Gambar 2 menjelaskan mengenai alur proses perhitungan menggunakan metode Function Points sebagai berikut:



Gambar 2 Alur proses perhitungan metode Function Points

Alur proses perhitungan pada metode Function Points (lihat Gambar 2) mengadopsi secara keseluruhan pada penelitian Allan J. Albrecht tahun 1979 [11].

1) Menghitung CFP (Crude Function Points)

CFP digunakan untuk menghitung bobot dari nilai komponen-komponen Function Points yang dikaitkan dengan perangkat lunak yang akan dikembangkan. Kemudian menghitung nilai Crude Function Points pada setiap kategori dan setiap nilai perhitungan dikalikan dengan faktor kompleksitas yang ditunjukkan oleh Tabel 1 berdasarkan kriteria dari setiap kategori.

Tabel 1. Perhitungan Bobot Kompleksitas Crude Function Points

Tipe Fungsi	Bobot Kompleksitas		
	Simple	Average	Complex
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6
Internal Logical File	7	10	15
External Logical File	5	7	10

Bobot kompleksitas yang disajikan pada Tabel 1 merupakan faktor pengali bagi jumlah yang diperoleh saat sebuah sistem dianalisis.

2) Menghitung RCAF (Relative Complexity Adjustment Factor)

RCAF digunakan untuk menghitung bobot kompleksitas berdasarkan karakteristik perangkat lunak. Penilaian kompleksitas memiliki skala antara 0 sampai 5. Nilai 0 jika faktor tersebut tidak menimbulkan pengaruh dan 5 jika faktor tersebut sangat penting pada perangkat lunak yang diukur.

Tabel 2. 14 Karakteristik Perangkat Lunak

No.	Faktor Kompleksitas
1.	Tingkat Kompleksitas Komunikasi Data
2.	Tingkat Kompleksitas Pemrosesan Terdistribusi
3.	Tingkat Kompleksitas Performance
4.	Tingkat Kompleksitas Konfigurasi
5.	Tingkat Frekuensi Penggunaan Software
6.	Tingkat Frekuensi Input Data
7.	Tingkat Kemudahan Penggunaan Bagi Pengguna
8.	Tingkat Frekuensi Update Data
9.	Tingkat Kompleksitas Processing Data
10.	Tingkat Kemungkinan Penggunaan Kembali(Reusable) Kode Program
11.	Tingkat Kemudahan dalam Instalasi
12.	Tingkat Kemudahan Operasional Software (Backup, Recovery)
13.	Tingkat Software dibuat untuk Multi Organisasi/Perusahaan/Client
14.	Tingkat Kompleksitas dalam Mengikuti Perubahan/Fleksibel

3) Menghitung Nilai Function Points

Untuk menghitung Function Points dapat digunakan rumus (1) sebagai berikut:

$$FP = CFP*(0.65+(0.01*RCAF)) \tag{1}$$

Nilai CFP diperoleh dari total bobot kompleksitas, sedangkan nilai RCAF diperoleh dari jumlah 14 faktor karakteristik perangkat lunak yang bernilai 0 sampai 5.

4) Mendistribusikan Nilai Usaha ke Aktivitas

Untuk mengubah nilai Function Points menjadi nilai usaha maka nilai Function Points harus dikalikan dengan dengan nilai konstanta PR (Productivity Rate) dapat digunakan rumus (2) sebagai berikut:

$$Effort = FP*PR \tag{2}$$

Nilai PR memiliki satuan orang/jam. Di penelitian sebelumnya, nilai kritis yang sesuai dengan pola pengembangan perangkat lunak adalah 8.2 orang/jam.

5) Perhitungan Estimasi Biaya Perangkat Lunak

Pada tahap akhir melakukan estimasi biaya perangkat lunak. Untuk estimasi biaya disesuaikan dengan tarif/jam dapat digunakan rumus (3) sebagai berikut:

$$Estimasi\ Biaya = Effort*Tarif\ per\ jam \tag{3}$$

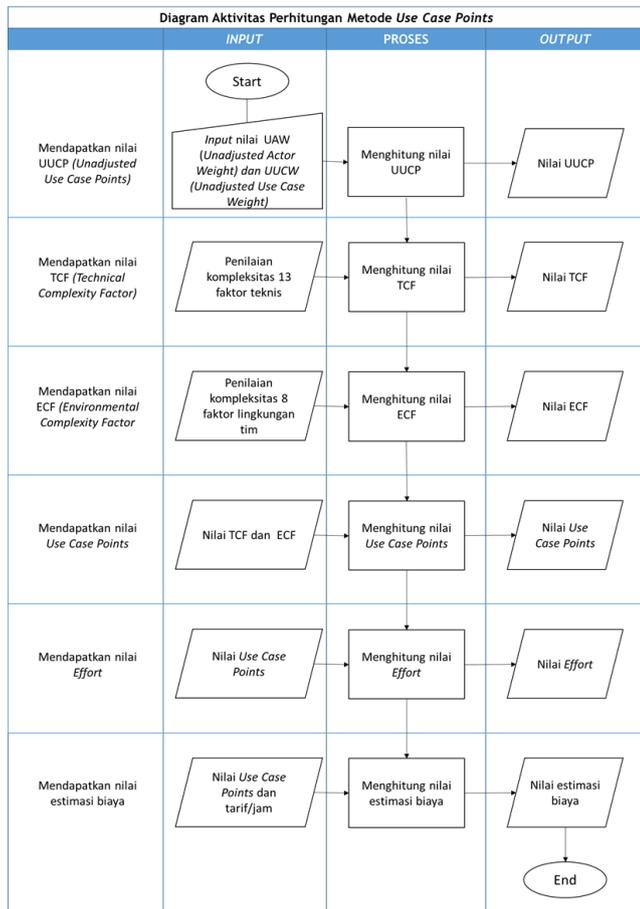
3.1.2 Use Case Points (UCP)

Tahapan dalam perhitungan usaha dan biaya proyek pengembangan perangkat lunak menggunakan metode UCP dapat dilihat pada Gambar 3.

1) Menghitung UUCW (Unadjusted Use Case Weight)

UUCW (Unadjusted Use Case Weight) didapatkan dari total aktivitas yang ada dalam setiap skenario use case.

Terdapat 3 kategori dalam menentukan nilai *use case*, yaitu mudah, sedang, dan kompleks. Setiap *use case* dikategorikan dari jumlah skenario yang terkandung didalamnya. Selanjutnya *use case* dijumlahkan dan dikalikan dengan bobot di masing-masing kategori.



Gambar 3 Alur proses perhitungan metode UCP

Tabel 4 menjelaskan mengenai klasifikasi bobot pada UUCW.

Tabel 4. Klasifikasi Bobot UUCW

Kategori	Jumlah Transaksi	Bobot
Mudah	1-3	5
Sedang	4-7	10
Kompleks	>7	15

Untuk menghitung nilai UUCW digunakan rumus (4) sebagai berikut:

$$UUCW = \Sigma(\text{jumlah transaksi} * \text{bobot}) \quad (4)$$

2) Menghitung UAW (*Unadjusted Actor Weight*)

UAW (*Unadjusted Actor Weight*) didapatkan berdasarkan kompleksitas dari semua aktor dalam *Use Case*. UAW mengkategorikan aktor berdasarkan kompleksitasnya dan diklasifikasikan kedalam bentuk *simple*, *average*, dan *complex* dengan setiap kategori memiliki bobot nilai. Tabel 5 menjelaskan mengenai klasifikasi bobot pada UAW.

Tabel 5. Klasifikasi Bobot UAW

Kategori	Tipe Aktor	Bobot
Mudah	Berinteraksi melalui API, contoh <i>Command Prompt</i>	1
Sedang	Berinteraksi melalui Protokol, contoh TCP/IP	2
Kompleks	Berinteraksi melalui GUI atau	3

Untuk menghitung nilai UAW digunakan rumus (5) sebagai berikut:

$$UAW = \Sigma(\text{jumlah tipe aktor} * \text{bobot}) \quad (5)$$

3) Menghitung UUCP (*Unadjusted Use Case Points*)

UUCP (*Unadjusted Use Case Points*) didapatkan melalui perancangan *Use Case* pada perangkat lunak yang dikembangkan. Untuk menghitung nilai UUCP digunakan rumus (6) sebagai berikut:

$$UUCP = UUCW + UAW \quad (6)$$

4) Menghitung TCF (*Technical Complexity Factor*)

TCF (*Technical Complexity Factor*) digunakan untuk melakukan penilaian terhadap faktor-faktor teknis yang berdampak terhadap produktivitas dari pengerjaan proyek. Setiap faktor teknis dievaluasi dengan tim pengembang berdasarkan tingkat kompleksitasnya. Untuk penilaian terhadap tingkat kompleksitas dari setiap faktor, diberikan bobot dari 0 sampai 5 berdasarkan pengaruh terhadap perangkat lunak. Bobot tersebut dikalikan dengan nilai masing-masing kompleksitas faktor. Tabel 6 menjelaskan mengenai TF (*Technical Factor*) dan pembobotannya.

Tabel 6. Klasifikasi Bobot *Technical Factor*

Kode	<i>Technical Factor</i>	Bobot
T1	<i>Distributed System Required</i>	2
T2	<i>Response Time Is Important</i>	1
T3	<i>End User Efficiency</i>	1
T4	<i>Complex Internal Processing Required</i>	1
T5	<i>Reusable Code Must Be A Focus</i>	1
T6	<i>Installation Easy</i>	0.5
T7	<i>Usability</i>	0.5
T8	<i>Cross-Platform Support</i>	2
T9	<i>Easy To Change</i>	1
T10	<i>Highly Concurrent</i>	1
T11	<i>Custom Security</i>	1
T12	<i>Dependence On Thrid-Part Code</i>	1
T13	<i>User Training</i>	1

Untuk menghitung TCF dapat digunakan rumus (7) sebagai berikut:

$$TCF = 0.6 + (0.01 * TF) \quad (7)$$

5) Menghitung ECF (*Enviromental Complexity Factor*)

Selain faktor kompleksitas teknis, faktor kompleksitas lingkungan ECF (*Enviromental Complexity Factor*) juga digunakan dalam menentukan estimasi biaya perangkat lunak. Untuk penilaian terhadap tingkat kompleksitas dari setiap faktor, diberikan bobot dari 0 sampai 5 berdasarkan pengaruh terhadap perangkat lunak. Bobot tersebut dikalikan dengan nilai masing-masing kompleksitas faktor.

Untuk menghitung *TCF* dapat digunakan rumus (8) sebagai berikut:

$$ECF = 1.4 + (-0.03 * EF) \tag{8}$$

6) Menghitung UCP

Perhitungan UCP membutuhkan hasil perhitungan UUCP, *TCF*, dan *ECF* (lihat pada rumus sebelumnya). Untuk menghitung UCP dapat digunakan rumus (9) sebagai berikut:

$$UCP = UUCP * TCF * ECF \tag{9}$$

7) Perhitungan Estimasi Biaya Perangkat Lunak

Pada tahap akhir melakukan estimasi biaya perangkat lunak. Untuk estimasi biaya disesuaikan dengan tarif/jam dikalikan dengan *Effort* masing-masing proyek perangkat lunak dapat digunakan rumus (10) sebagai berikut:

$$\text{Estimasi Biaya} = (UCP * 8.2) * \text{Tarif per jam} \tag{10}$$

3.2 Analisis Kebutuhan

Tahap pembangunan aplikasi dapat dilakukan apabila telah dilakukan analisis terhadap kebutuhan pengguna. Berdasarkan studi literatur yang telah dilakukan akan dijelaskan analisa kebutuhan pada Tabel 7 sebagai berikut.

Tabel 7. Analisis Kebutuhan Pengguna

No.	Kebutuhan Pengguna	Solusi Yang Ditawarkan
1.	Perhitungan estimasi pengembangan perangkat lunak lebih mudah.	Aplikasi berbasis <i>mobile</i> Android sehingga memudahkan pengguna dalam melakukan perhitungan dan pengguna melakukan input nilai-nilai yang diperlukan di aplikasi.
2.	Metode estimasi pengembangan perangkat lunak dapat digunakan pada beberapa proyek pengembangan perangkat lunak.	Metode yang digunakan pada aplikasi menggunakan dua metode yaitu metode <i>Function Points</i> dan <i>Use Case Points</i> .
3.	Dapat melakukan perubahan data/input saat proses estimasi.	Pengguna dapat mengubah data sebelum melakukan penyimpanan.
4.	Melihat rekap hasil perhitungan estimasi.	Terdapat fitur riwayat hasil perhitungan.
5.	Biaya proyek dapat disesuaikan oleh pengguna berdasarkan kebutuhan dalam pengembangan proyek perangkat lunak.	Pengguna dapat menentukan tarif/jam pada proyek pengembangan perangkat lunak.
6.	Estimasi perhitungan biaya perangkat lunak digunakan oleh pengguna yang ingin mengembangkan sebuah aplikasi dan mengetahui estimasi harga perangkat lunak.	Metode yang digunakan yaitu metode <i>Function Points</i> dan <i>Use Case Points</i> yang terdapat penjelasan singkat saat melakukan perhitungan sehingga pengguna dapat menyesuaikan dengan kebutuhan.

3.3 Kebutuhan Fungsional

Pada kebutuhan fungsional ini dijelaskan fitur-fitur atau layanan yang terdapat pada aplikasi. Tabel 8 berikut ini merupakan spesifikasi kebutuhan fungsional yang berhasil diidentifikasi.

Tabel 8. Kebutuhan Fungsional

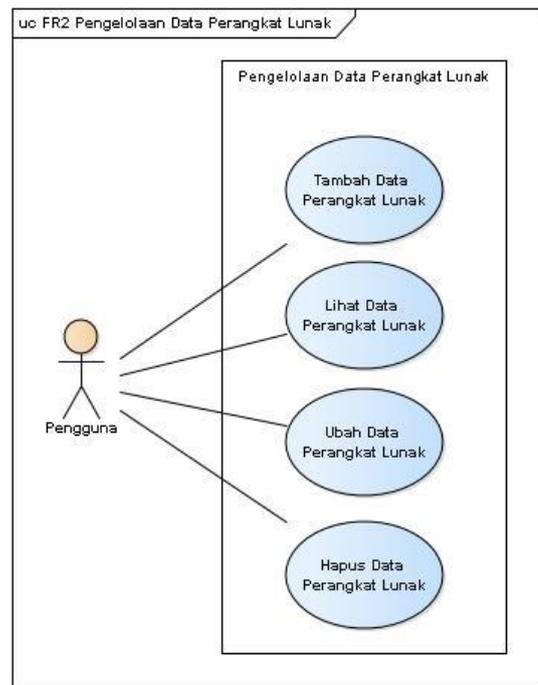
Kode	Kebutuhan Fungsional	Nama Use Case
FR1	Pengelolaan Akun/Autentifikasi	UC 1.1 <i>Login</i>
		UC 1.2 <i>Logout</i>
		UC 1.3 <i>Register</i>
FR2	Pengelolaan Data Perangkat Lunak	UC 2.1 Tambah Data Perangkat Lunak
		UC 2.2 Lihat Data Perangkat Lunak
		UC 2.3 Ubah Data Perangkat Lunak
		UC 2.4 Hapus Data Perangkat Lunak
FR3	Pengelolaan Data Perhitungan <i>Function Points</i> (FP)	UC 3.1 Tambah Data Perhitungan FP
		UC 3.2 Lihat Data Perhitungan FP
		UC 3.3 Ubah Data Perhitungan FP
FR4	Pengelolaan Data Perhitungan <i>Use Case Points</i> (UCP)	UC 4.1 Tambah Data Perhitungan UCP
		UC 4.2 Lihat Data Perhitungan UCP
FR5	Riwayat Perhitungan	UC 5.1 Lihat Riwayat Perhitungan
		UC 5.2 Hapus Riwayat Perhitungan

3.4 Perancangan

Dalam membangun aplikasi ini diperlukan perancangan arsitektur perangkat lunak meliputi *use case diagram*, *sequence diagram*, *activity diagram*, dan *conceptual data model* sebagai berikut:

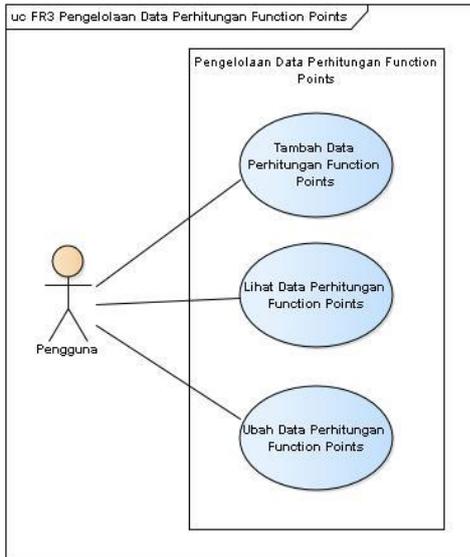
3.4.1 Use Case Diagram

Use Case Diagram mendeskripsikan sebuah interaksi antara aktor dengan sistem yang akan dibangun. *use case diagram* yang diidentifikasi berdasarkan kebutuhan fungsional FR2 dapat dilihat pada Gambar 4.



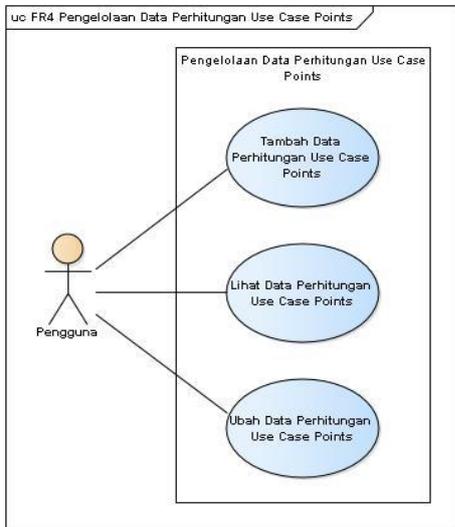
Gambar 4 Use case diagram pengelolaan data proyek

Gambar 5 menunjukkan use case diagram untuk kebutuhan fungsional dengan kode FR3 yaitu pengelolaan data perhitungan *Function Points*.



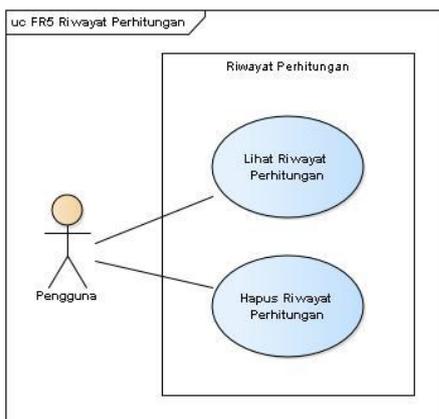
Gambar 5 Use case diagram perhitungan function points

Gambar 6 menunjukkan use case diagram untuk kebutuhan fungsional dengan kode FR4 yaitu pengelolaan data perhitungan Use Case Points.



Gambar 6 Use case diagram perhitungan use case points

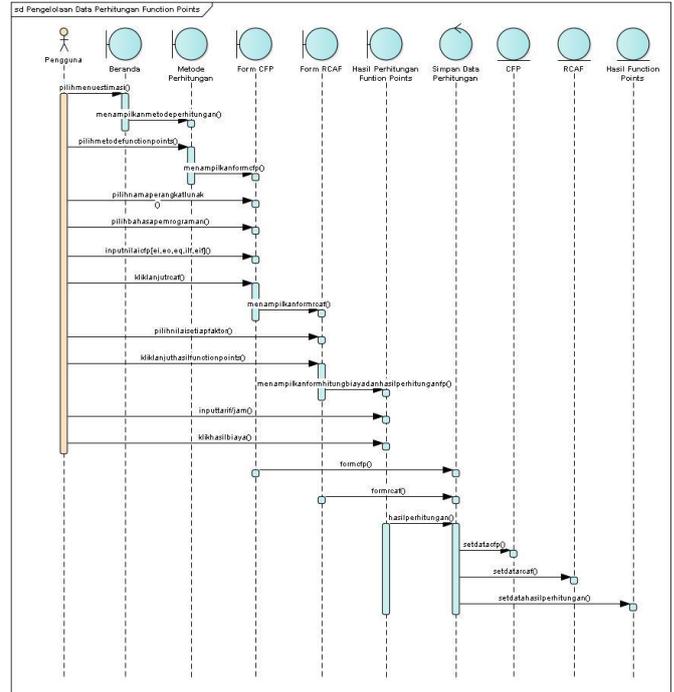
Gambar 7 menunjukkan use case diagram kebutuhan fungsional dengan kode FR5 yaitu riwayat perhitungan.



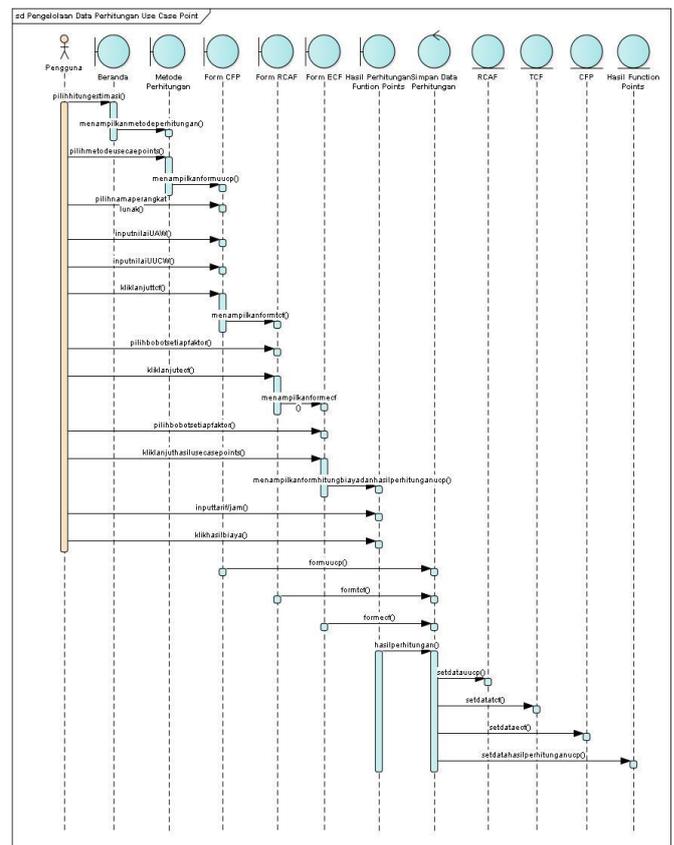
Gambar 7 Use case diagram riwayat perhitungan

3.4.2 Sequence Diagram

Sequence diagram merupakan hubungan antar objek dari pengguna kepada sistem. Berikut ini adalah cuplikan sequence diagram pengelolaan data perhitungan metode Function Points (Gambar 8). Sedangkan Gambar 9 merupakan sequence diagram untuk perhitungan data Use Case Points.



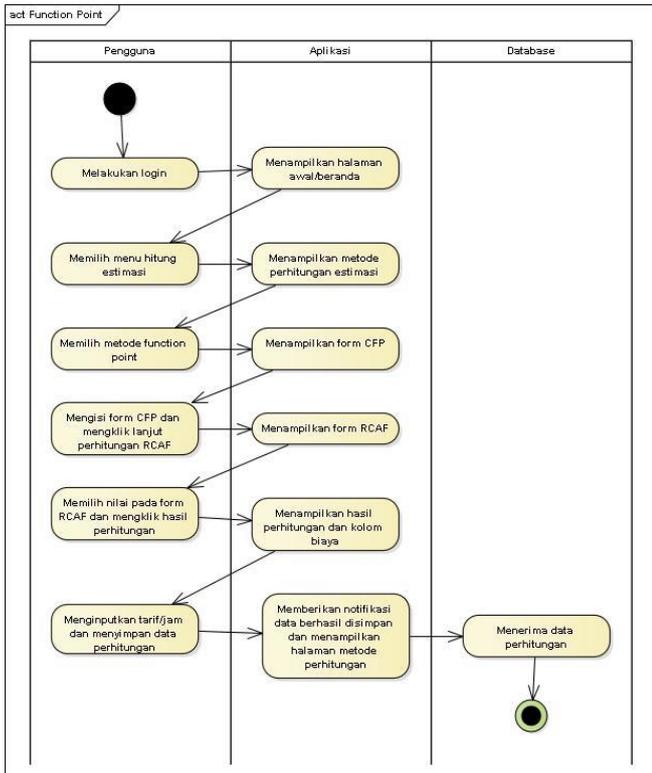
Gambar 8 Sequence diagram perhitungan function points



Gambar 9 Sequence diagram perhitungan use case points

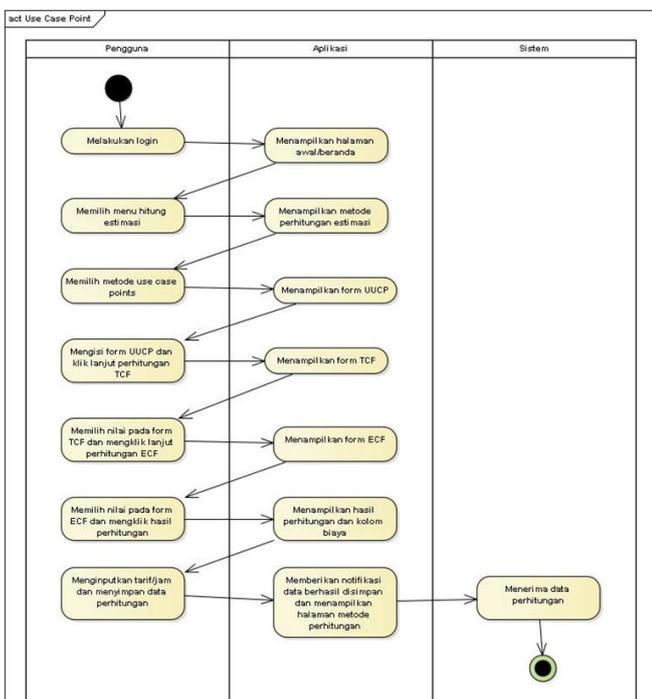
3.4.3 Activity Diagram

Activity Diagram menggambarkan berbagai alur aktivitas dalam sistem dari awal hingga akhir alur. Gambar 10 menunjukkan activity diagram pengelolaan perhitungan data dengan metode Function Points.



Gambar 10 Activity diagram perhitungan Function Points

Activity diagram untuk perhitungan data dengan metode Use Case Points dapat dilihat pada Gambar 11.



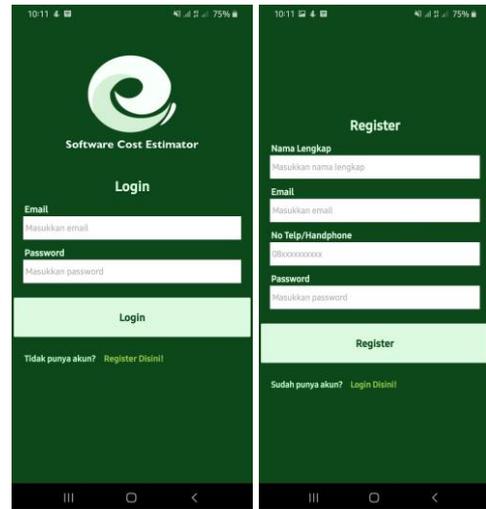
Gambar 11 Activity diagram perhitungan Use Case Points

3.5 Implementasi Aplikasi

Pada tahap ini digunakan kode program untuk menghasilkan aplikasi sesuai dengan fitur yang diharapkan. Berikut adalah tampilan fitur aplikasi estimasi usaha dan biaya pengembangan software menggunakan metode Function Points dan Use Case Points.

3.5.1 Fitur Autentikasi

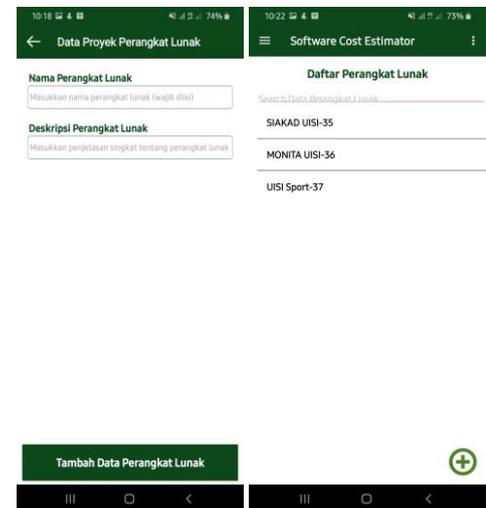
Fitur autentikasi merupakan proses untuk memvalidasi pengguna saat mengakses aplikasi dan sebelum melakukan perhitungan estimasi usaha dan biaya pengembangan perangkat lunak. Tampilan fitur autentikasi dapat dilihat pada Gambar 12 (a) dan (b).



Gambar 12 Fitur autentikasi

3.5.2 Fitur Data Perangkat Lunak

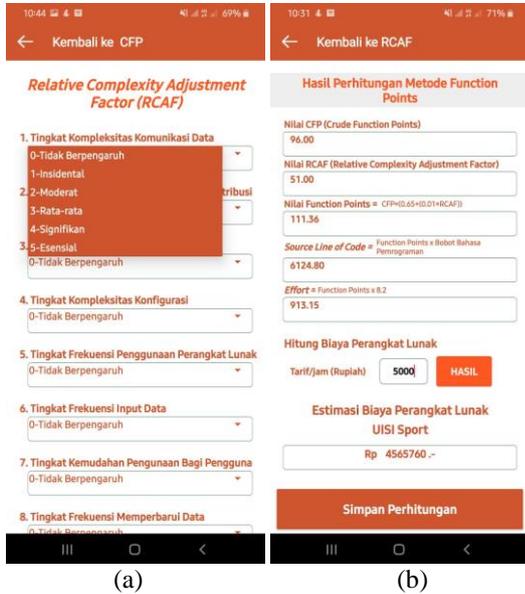
Fitur ini digunakan untuk menyimpan data perangkat lunak yang berupa nama dan deskripsi perangkat lunak. Gambar 13 (a) dan (b) menunjukkan tampilan fitur pendataan proyek pengembangan perangkat lunak yang akan dilakukan perhitungan estimasi usaha dan biayanya.



Gambar 13 Fitur data perangkat lunak

3.5.3 Fitur Perhitungan Estimasi Metode *Function Points*

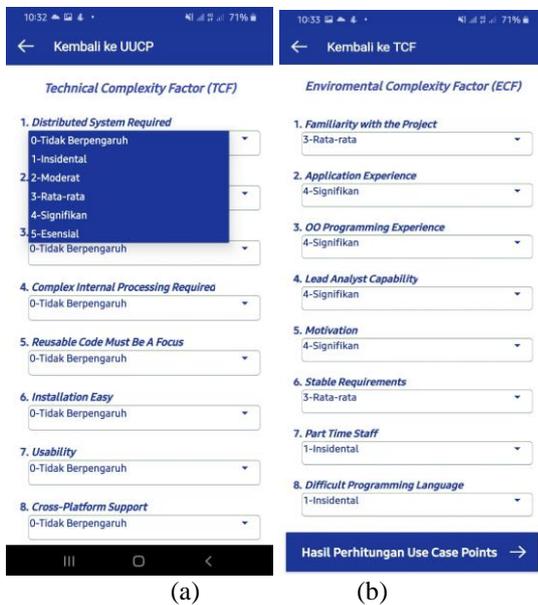
Pada fitur ini berisikan tahapan-tahapan pada perhitungan metode *Function Points* meliputi perhitungan CFP, RCAF, hingga menghasilkan nilai *Function Points*, *Effort*, dan hasil estimasi biaya perangkat lunak. Tampilan fitur perhitungan estimasi dengan metode FP dapat dilihat pada Gambar 14 (a) untuk RCAF dan (b) hasil perhitungan effort dan konversi biayanya.



Gambar 14 Fitur perhitungan estimasi metode *function points*

3.5.4 Fitur Perhitungan Estimasi Metode *Use Case Points*

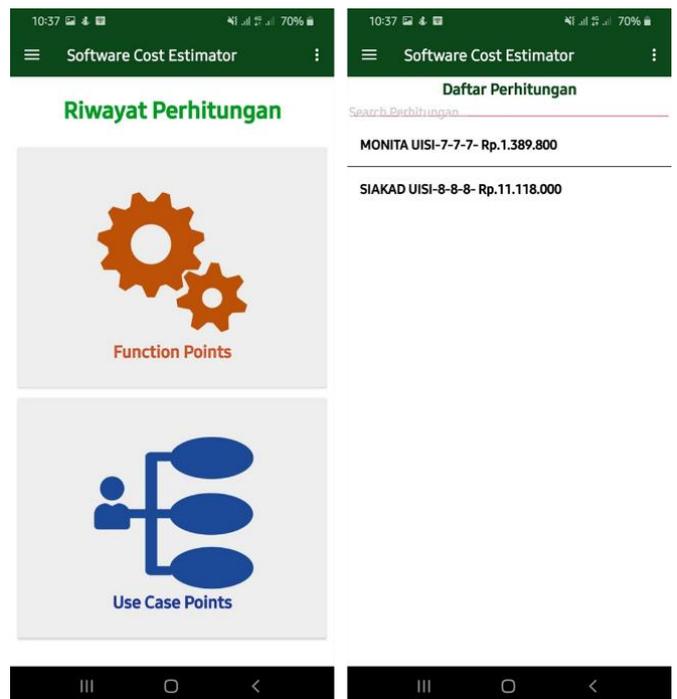
Pada fitur ini berisikan tahapan-tahapan pada perhitungan metode *Use Case Points* meliputi perhitungan UUCP, TCF, ECF hingga menghasilkan nilai *Use Case Points*, *Effort*, dan hasil estimasi biaya perangkat lunak. Tampilan fitur perhitungan estimasi dengan metode UCP dapat dilihat pada Gambar 15 (a) faktor kompleksitas teknis, (b) faktor kompleksitas lingkungan, dan (c) hasil perhitungan effort dan konversi biayanya.



Gambar 15 Fitur perhitungan estimasi metode *use case points*

3.5.5 Fitur Riwayat Perhitungan

Pada fitur riwayat perhitungan terdapat data hasil perhitungan perangkat lunak yang telah disimpan pengguna. Tampilan fitur riwayat perhitungan yang telah disimpan dapat dilihat pada Gambar 16.



Gambar 16 Riwayat Perhitungan

3.6 Ujicoba Aplikasi

Ujicoba dilakukan secara kuantitatif yaitu melakukan verifikasi data perhitungan dengan 2 lapis. Verifikasi lapis pertama adalah membandingkan nilai perhitungan usaha pada kertas kerja MS Excel dari perhitungan manajer proyek dengan nilai yang dihasilkan dalam aplikasi ini. Setelah dilakukan verifikasi perhitungan usaha (*effort*) pengembangan 21 aplikasi, penulis tidak menemukan perbedaan nilai (margin error = 0).

Verifikasi lapis kedua adalah membandingkan nilai estimasi biaya pengembangan perangkat lunak yang telah dihitung manajer proyek dengan hasil kalkulasi pada aplikasi ini. Ujicoba kembali dilakukan pada 21 aplikasi dan penulis tidak menemukan perbedaan data.

4. Kesimpulan

Dari hasil penelitian di atas, maka penulis dapat menyimpulkan beberapa temuan sebagai berikut:

1. Aplikasi pada penelitian ini menghasilkan satuan *effort* (orang/jam) berdasarkan metode *Function Points* dan *Use Case Points*
2. Estimasi biaya yang dapat disesuaikan oleh kebutuhan pengguna dengan konversi tarif/jam berdasarkan Kelly Service Indonesia Salary Guide tahun 2020.
3. Aplikasi telah berhasil diujicobakan pada 21 proyek pengembangan aplikasi

Kedepan, penelitian ini akan dikembangkan berdasarkan aktivitas dan tim pengembang perangkat lunak sehingga dapat mengetahui biaya setiap aktivitas yang dilakukan oleh tim pengembang sesuai pengalaman yang dimiliki (*work experience*).

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Departemen Sistem Informasi Universitas Internasional Semen Indonesia (UISI) yang telah memberi fasilitas dan dukungan moral terhadap penelitian ini.

Referensi

- [1] Y. Pratama and E. Rasywir, "Automatic Cost Estimation Analysis on Datawarehouse Project with Modified Analogy Based Method," in *Proceedings of 2018 International Conference on Electrical Engineering and Computer Science, ICECOS 2018*, 2019.
- [2] S. Muklis, K. Kusriani, and A. Sunyoto, "Perancangan Sistem Estimasi untuk Menentukan Biaya Kelayakan Proyek IT," *Creat. Inf. Technol. J.*, vol. 5, no. 2, p. 114, 2019.
- [3] Sholiq, R. S. Dewi, and A. P. Subriadi, "A Comparative Study of Software Development Size Estimation Method: UCPabc vs Function Points," in *Procedia Computer Science*, 2017, vol. 124, pp. 470–477.
- [4] N. Rachmat and S. Saparudin, "Estimasi Ukuran Perangkat Lunak Menggunakan Function Point Analysis-Studi Kasus Aplikasi Pengujian dan Pembelajaran Berbasis Web," *Annu. Res. Semin.*, vol. 3, no. 1, pp. 57–60, 2017.
- [5] R. S. Dewi, T. W. Andari, M. B. A. Rasyid, and R. C. AP, "Ekstraksi Faktor Kompleksitas Game Menggunakan Metode Function Points," *J. Teknol. dan Sist. Inf.*, vol. 4, no. 3, pp. 115–122, 2018.
- [6] J. Kristi, S. N. Aisah, and R. S. Dewi, "Estimasi Biaya Software FAS (Financing Analysis System) Menggunakan Metode Function Point (Studi Kasus Pada PT BPRS Lantabur Tebuireng)," *JURIKOM (Jurnal Ris. Komputer)*, 2020.
- [7] Rifki Adhitama;Condro Kartiko, "Effort Estimation Menggunakan Metode Use Case Point untuk Pengembangan Perangkat Lunak," *J. Informatics, Inf. Syst. Softw. Eng. Appl.*, vol. 1, no. 1, pp. 63–70, 2018.
- [8] R. S. Dewi, A. P. Subriadi, and S. Sholiq, "Use Case Point - Activity-Based Costing: Metode Baru untuk Mengestimasi Biaya Pengembangan Perangkat Lunak," *Sisfo*, 2015.
- [9] M. R. Z. Alnobeta, M. C. Saputra, and A. D. Herlambang, "Estimasi Biaya Perangkat Lunak Menggunakan Metode Function Point (Studi Kasus : CV Aptikma Indonesia)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 1, pp. 40–46, 2018.
- [10] R. S. Dewi, A. P. Subriadi, and Sholiq, "A Modification Complexity Factor in Function Points Method for Software Cost Estimation Towards Public Service Application," *Procedia Comput. Sci.*, vol. 124, pp. 415–422, 2017.
- [11] A. J. Albrecht, "Measuring application development productivity," in *IBO Conference on Application Development*, 1979, pp. 83–92.