

# METODE FUNCTION POINT UNTUK ESTIMASI BIAYA PROYEK PENGEMBANGAN APLIKASI E-PRES SAJA

Nurul Amalia<sup>1</sup>, Nur Ika Royanti<sup>2</sup>, Indrayanti<sup>3</sup>  
<sup>1,2,3</sup>STMIK Widya Pratama, Pekalongan

E-mail: [amalia.0121@gmail.com](mailto:amalia.0121@gmail.com)

## Abstrak

Kemajuan teknologi telah memberikan kemudahan bagi manusia dalam hal mendapatkan informasi maupun dalam menyelesaikan pekerjaan. Adanya manfaat yang diberikan oleh suatu teknologi, maka banyak yang mengembangkan perangkat lunak untuk membantu kinerja manusia. Dalam pengembangan perangkat lunak biaya pengembangan proyek merupakan biaya yang mendominasi yaitu mencapai 60%. Proyek perangkat lunak sering mengalami penundaan, over budget, dan tidak selesai karena kegagalan memperkirakan biaya pengembangan perangkat lunak. Maka perlu dilakukan penentuan estimasi biaya yang baik menggunakan metode *Function Point*. *Function Point* merupakan metode pengukuran fungsionalitas perangkat lunak berdasarkan tipe fungsi pengguna yaitu *External Input*, *External Output*, *External Inquire*, *Internal Logic File*, dan *External Interface File* serta perhitungan teknis dari pengembangan perangkat lunak. Hasil dari penelitian pengembangan proyek pada aplikasi E-pres Saja membutuhkan biaya sekitar Rp 49.025.482

**Kata kunci:** Estimasi Biaya, Function Point, Presensi Pegawai

## Abstract

*Technological advances have made it easier for humans to obtain information and complete work. Due to the benefits provided by technology, many people develop software to help human performance. In software development, project development costs are the dominant costs, reaching 60%. Software projects often experience delays, go over budget, and are not completed due to failure to estimate software development costs. So it is necessary to determine a good cost estimate using the Function Point method. Function Point is a method of measuring software functionality based on the type of user function, namely External Input, External Output, External Inquire, Internal Logic File, and External Interface File as well as technical calculations from software development. The results of project development research on the E-pres application alone cost around IDR 49,025,482*

**Key word:** Cost Estimates, Function Points, Employee Presence

## 1. Pendahuluan

Kemajuan teknologi telah memberikan kemudahan bagi manusia dalam hal mendapatkan informasi maupun dalam menyelesaikan pekerjaan (Widyastuti, Susi, Ariandi, Wahyu, & Sulistiono 2019). Adanya manfaat yang diberikan oleh suatu teknologi, maka banyak yang mengembangkan perangkat lunak untuk membantu kinerja manusia.

Dalam pengembangan perangkat lunak tahap perencanaan merupakan tahap awal yang akan menentukan keberhasilan suatu proyek perangkat lunak. Kegagalan suatu proyek dapat terjadi karena memiliki perencanaan yang buruk. Proyek tidak dapat dikendalikan jika proyek tersebut tidak mempunyai estimasi dan perencanaan. Estimasi dan perencanaan dalam proyek perangkat lunak bisa diawali dengan pengukuran dari perangkat lunak yang akan dibuat atau dikembangkan. Maka dari itu dalam membangun perangkat lunak sebaiknya dimulai dengan perencanaan proyek yang matang (Sevilla, Jiménez, and Plaza 2015). Aktifitas dari perencanaan proyek perangkat lunak yaitu estimasi

sumber daya, biaya serta jadwal proyek (Sandhea, Maulidiyah, and Sari 2020).

Estimasi ukuran perangkat lunak, adalah proses pengembangan untuk menentukan arah dan tujuan serta menjaga proses pengembangan tetap dalam kendali. Untuk mendapatkan hasil estimasi yang akurat, diperlukan metode yang memiliki akurasi tinggi. Estimasi biaya dan waktu dapat diperoleh dengan beberapa metode antaranya adalah: Constructive Cost, Model (COCOMO) II, Expert Judgment, Function Point (FP), Analogi, Neural Network, Fuzzy, dan lain-lain. Masing masing metode mempunyai kelebihan dan kekurangan untuk pengembangan perangkat lunak (Grandys Frieska Prassida, Achmad Holil Noor Ali 2018). Function Point Analysis (FPA) merupakan metode estimasi ukuran perangkat lunak yang memiliki akurasi cukup baik. FPA memperkirakan ukuran perangkat lunak dalam Function Point (FP) yang dapat digunakan untuk *menghitung Line of Code (LOC)* (Saptono and Utama 2016). Keunggulan dari Function Point yaitu mampu untuk mengukur perkiraan volume proyek dalam

bentuk sumber daya pengembangan yang dibutuhkan sebagai dasar penting untuk memperkirakan harga jual perangkat lunak yang dibuatnya (Permatasari 2016)

Dalam mengembangkan perangkat lunak Aplikasi E-PresSaja dibutuhkan manajemen proyek yang baik untuk mengetahui estimasi biaya maupun waktu dari perangkat lunak tersebut untuk menghindari adanya *over estimates* dan *under estimates*. *Over estimates* akan menimbulkan penambahan alokasi dana maupun sumberdaya yang dibutuhkan sedangkan *under estimates* secara tidak langsung akan mengurangi kualitas perangkat lunak yang dibangun atau dikembangkan karena untuk menekan biaya maka perangkat lunak dibuat biasa saja dan tidak sesuai dengan standar yang diinginkan. Oleh karenanya proses perancangan ini sangat penting dilakukan untuk mengetahui estimasi biaya maupun waktu yang diperlukan dalam pembangunan perangkat lunak tersebut.

Aplikasi E-pres saja merupakan aplikasi mobile berbasis android yang dikembangkan untuk mempermudah penyapu jalan non pegawai melakukan presensi kehadiran dengan menscan wajah pegawai dan memanfaatkan GPS untuk menangkap titik lokasi pegawai berada dan mengirimkannya ke server sehingga pegawai tidak perlu datang ke kantor dinas lingkungan hidup untuk melakukan presensi yang terkadang letaknya jauh dari titik lokasi tempat yang akan dibersihkan. Dengan adanya sistem ini lebih efektif dan efisien dalam proses presensi, perekapan presensi dan pemantauan presensi secara realtime.

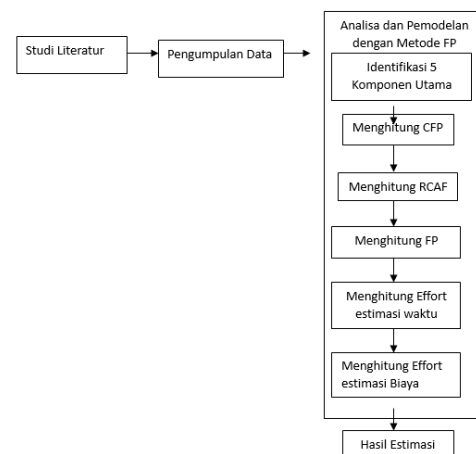
Suatu sistem dikatakan baik ketika memulai perencanaan yang dapat memperkirakan kebutuhan biaya, waktu dan jumlah sumber daya yang diperlukan sebagai dasar perencanaan dalam pengembangan suatu sistem, sebab Pengukuran perangkat lunak menyangkut daur hidup sistem yang dapat digunakan untuk memutuskan bentuk perangkat lunak versi selanjutnya, dalam bentuk pengurangan ataupun penambahan fitur produk. Selain itu Pengukuran Perangkat lunak juga dapat digunakan untuk mengukur faktor kebutuhan dan budget dalam sebuah proyek perangkat lunak, seperti sumber daya manusia dan biaya (Sandhea, Maulidiyah, and Sari 2020). Estimasi atau perkiraan ukuran perangkat lunak dalam proses pengembangan perangkat lunak memegang peranan penting untuk menjaga proses pengembangan tetap dalam kendali pengembang sehingga estimasi biaya dapat menjadi elemen dalam suatu penilaian karena salah satu penentu keberhasilan pengerjaan sebuah proyek adalah adanya estimasi biaya dan estimasi effort suatu proyek

Maka dalam penelitian ini akan dilakukan perhitungan estimasi biaya proyek pada Aplikasi E-PresSaja menggunakan Metode Function Point

dengan tujuan sebagai pedoman awal dalam perhitungan rencana untuk mendapatkan perkiraan biaya yang dikeluarkan pada sistem yang akan dikembangkan. Dengan demikian akurasi estimasi biaya suatu proyek merupakan salah satu penentu keberhasilan suatu proyek (Maddeppungeng and Suryani 2016). Function Point sendiri adalah suatu pengukuran standar dari perangkat lunak untuk kuantifikasi fungsionalitas yang ditawarkan program ke pengguna. Metode perhitungan Function Point berdasarkan pada identifikasi fungsi yang dilakukan oleh sistem dan memberikan tingkat kompleksitas pada setiap fungsi dan dianggap lebih cepat karena tidak memerlukan hasil analisis sistem dalam bentuk skema pelaporan penggunaan naratif dan konsep basis data.

## 2. Metode Penelitian

Penelitian ini bertujuan untuk mengetahui estimasi biaya dari pengembangan Aplikasi E-PresSaja (Elektronik Presensi Sapu Jalan). Secara garis besar terdapat metode yang perlu dilakukan dalam penelitian ini yang dijelaskan pada gambar 1.



Gambar 1. Metode Penelitian

### 2.1. Studi Literatur

Metode penelitian diawali dari studi literatur yang dilakukan dengan mempelajari materi melalui studi literatur yang dibutuhkan dalam melakukan estimasi proyek melalui buku maupun paper paper yang akan dicari melalui internet.

### 2.2. Pengumpulan Data

Pengumpulan data dilakukan dengan menggunakan wawancara untuk mendapatkan data kualitatif berupa informasi terkait kebutuhan fungsionalitas pada Sistem Elektronik Presensi Sapu Jalan.

### 2.3. Analisa dan Pemodelan dengan Metode FP

Metode Function Point adalah sebuah metode yang melakukan pendekatan berorientasi pada fungsionalitas dan kompleksitas perangkat lunak dalam mengestimasi size perangkat lunak

yang selanjutnya digunakan untuk mengestimasi effort dan estimasi biaya yang diperlukan untuk pengembangan perangkat lunak (Saptono and Anggrainingsih 2016). Gunanya (1) untuk memprediksi atau memperkirakan biaya yang dibutuhkan dalam sistem perangkat lunak, (2) untuk memprediksi jumlah error yang mungkin akan terjadi, dan (3) memperkirakan jumlah komponen pada sistem yang diimplementasikan (Fikri and Sekarsari 2015).

Hasil dari pengumpulan data yang diperoleh dilakukan analisa untuk membuat model estimasi menggunakan metode function point untuk mengetahui perhitungan dalam memperkirakan waktu dan biaya yang dibutuhkan sehingga bisa menjadi acuan dalam mengembangkan aplikasi E-Pres Saja. Dalam melakukan Analisa dan pemodelan dengan metode function point ada beberapa tahapan yang dilakukan diantaranya:

### 2.3.1. Identifikasi 5 Komponen Utama

Terdapat 5 komponen dalam Function Point yang dapat digunakan secara efektif sebagai alat untuk memprediksi ukuran sebuah perangkat lunak/sistem yang akan dihasilkan. Komponen dan tipe fungsi pengguna berdasarkan pada karakteristik kompleksitas yang dimiliki meliputi sebagai berikut:

1. External Input (EI), adalah proses dasar yang memproses data dan informasi kontrol yang datang dari luar Batasan aplikasi, baik dari user maupun sistem lainnya yang selanjutnya digunakan untuk mengupdate Internal Logical Files.
2. External Output (EO), adalah sebuah proses dasar dimana hasil data dilewatkan dari dalam ke keluar dari batasan aplikasi untuk menyediakan Informasi kepada user, baik dalam bentuk laporan, tampilan di layar, pesan error, dst.
3. Internal Logical File (ILF), adalah kelompok data atau kelompok informasi kontrol yang digunakan dalam aplikasi, merupakan data yang dikelompokkan secara logis, disimpan secara internal dan didapat dari input eksternal.
4. External Interface File (EIF), adalah kelompok data berelasi atau informasi kontrol yang dirujuk oleh aplikasi, tapi dipelihara oleh aplikasi lain, yang dikelompokkan secara logis namun berada diluar aplikasi yang menyediakan Informasi yang dibutuhkan aplikasi.
5. External Inquiry (EQ), fungsi utamanya adalah menyediakan informasi ke user melalui pengambilan/pemrosesan data atau informasi kontrol dari ILF/EIF, sebagai input online yang memicu respon dari software untuk menghasilkan output online.

### 2.3.2. Menghitung CFP (Crude Function Points)

Pada tahap ini Crude Function Point (CFP) digunakan untuk menghitung bobot nilai dari komponen-komponen function points yang dikaitkan dengan suatu sistem yang akan dikembangkan. Setiap komponen diklasifikasikan berdasarkan tingkat kompleksitas antara lain mudah, sedang, dan tinggi. Nilai dari masing masing komponen kompleksitas dikalikan dengan bobot kompleksitas yang telah ditentukan. Perhitungan CFP melibatkan 5 tipe komponen sistem software yaitu: *External Input*, *External Output*, *External Inquire*, *Internal Logic File*, dan *External Interface File*. Nilai pengelompokan dalam bobot kompleksitas *Low*, *Average*, dan *High* untuk masing-masing komponen terdapat pada tabel 1.

Tabel 1 Pengelompokan Bobot kompleksitas Komponen

| Komponen                             | Level Kompleksitas |         |        |
|--------------------------------------|--------------------|---------|--------|
|                                      | Low                | Average | Height |
| <i>External Inputs (EI)</i>          | 3                  | 4       | 6      |
| <i>Internal Outputs (EO)</i>         | 4                  | 5       | 7      |
| <i>External Inquiries (EQ)</i>       | 3                  | 4       | 6      |
| <i>Internal Logical Files (ILF)</i>  | 7                  | 10      | 15     |
| <i>External Interface File (EIF)</i> | 5                  | 7       | 10     |

Panduan untuk mempermudah dalam menentukan nilai-nilai yang akan diberikan pada komponen perangkat lunak yang akan dinilai nanti terdapat pada tabel 2.

Tabel 2 Panduan Definisi Penilaian Komponen pada proses CFP

| Tipe   | Kompleksitas | Definisi                                                                                                                                 |
|--------|--------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Input  | Low          | Input sederhana apabila data yang diinput pada suatu fitur hanya dicocokkan dengan data yang telah ada dan tidak melibatkan banyak data. |
|        | Average      | Input menengah apabila fitur memungkinkan untuk memasukkan banyak data sekaligus ke dalam basis data.                                    |
|        | Height       | Input kompleks apabila fitur memungkinkan untuk memperbarui basis data seperti menambah / menghapus / mengubah data.                     |
| Output | Low          | Output tergolong sederhana apabila fitur hanya                                                                                           |

| Tipe                    | Kompleksitas | Definisi                                                                                                                                                |
|-------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         |              | menampilkan data statis.                                                                                                                                |
|                         | Average      | Output tergolong menengah apabila fitur menampilkan banyak data dari basis data atau menampilkan data berupa gambar, video, atau tipe data yang serupa. |
|                         | Height       | Output tergolong kompleks apabila data yang ditampilkan adalah data hasil perhitungan dari data yang lain dan melibatkan banyak data pada basis data.   |
| Query / Search / View   | Low          | Data yang tersimpan terlihat simple disimpan tidak dalam database                                                                                       |
|                         | Average      | Data yang tersimpan menunjukkan proses                                                                                                                  |
|                         | Height       | Data yang tersimpan memiliki data yang lebih dari 1/kompleks dan disimpan didalam database.                                                             |
| File / Tabel / Database | Low          | Data berisi tentang file atau database yang jumlahnya sedikit, sekitar lima baris data.                                                                 |
|                         | Average      | Data berisi tentang file atau database yang jumlahnya cukup banyak dan beberapa berhubungan dengan data lainnya                                         |
|                         | Height       | Data berisi tentang file atau database yang jumlahnya sangat banyak dan semua data tersebut terhubung dengan data lainnya                               |
| Interface Eksternal     | Low          | Tergolong sederhana apabila interface external hanya dapat mengambil satu jenis data.                                                                   |
|                         | Average      | Tergolong menengah apabila interface external hanya dapat mengambil dua atau tiga jenis data.                                                           |
|                         | Height       | Tergolong kompleks apabila interface external dapat mengambil lebih dari tiga jenis data.                                                               |

### 2.3.3. Menghitung RCAF (Relative Complexity Adjusment Factor)

RCAF berfungsi untuk menghitung kesimpulan kompleksitas dari sistem software dari beberapa subyek karakteristik. Penilaian berskala 0-5 diberikan pada tiap subyek yang paling berpengaruh terhadap usaha pengembangan yang dibutuhkan.

Tabel 3 Penilaian RCAF

| No | Karakteristik                                 | Deskripsi                                                                                         | Skor |
|----|-----------------------------------------------|---------------------------------------------------------------------------------------------------|------|
| 1  | Tingkat kompleksitas Komunikasi Data          | Tingkat kebutuhan komunikasi langsung antara aplikasi dan processor                               | 0-5  |
| 2  | Tingkat kompleksitas Pemrosesan Terdistribusi | Tingkat kebutuhan transfer data antara komponen-komponen aplikasi                                 | 0-5  |
| 3  | Tingkat kompleksitas Performance              | Tingkat response time dan throughput yang perlu dipertimbangkan dalam pengembangan aplikasi.      | 0-5  |
| 4  | Tingkat kompleksitas Konfigurasi              | Tingkat kebutuhan dimana setting konfigurasi komputer berpengaruh terhadap pengembangan aplikasi. | 0-5  |
| 5  | Tingkat Frekuensi Penggunaan Software         | Tingkat kecepatan transaksi bisnis yang berpengaruh terhadap pengembangan aplikasi.               | 0-5  |
| 6  | Tingkat Frekuensi Input Data                  | Tingkat kebutuhan peng-input-an data secara interaktif                                            | 0-5  |
| 7  | Tingkat Kemudahan Penggunaan Bagi User        | Tingkat kemudahan penggunaan aplikasi                                                             | 0-5  |
| 8  | Tingkat Frekuensi Update Data                 | Tingkat kebutuhan ILF diupdate secara online.                                                     | 0-5  |
| 9  | Tingkat Kompleksitas Prosesing Data           | Tingkat kesulitan logika proses yang mempengaruhi proses development                              | 0-5  |
| 10 | Tingkat Kemungkinan Penggunaan                | Tingkat kebutuhan aplikasi dan kode                                                               | 0-5  |

| No | Karakteristik                                                    | Deskripsi                                                                                               | Skor |
|----|------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|------|
|    | Kembali/Reusable Kode Program                                    | program aplikasi dirancang dan dikembangkan untuk bisa digunakan pada aplikasi lain                     |      |
| 11 | Tingkat Kemudahan Dalam Instalasi                                | Tingkat kemudahan konversi ke sistem baru yang berpengaruh pada proses development.                     | 0-5  |
| 12 | Tingkat Kemudahan operasional software (backup, recovery, dsb)   | Tingkat kemudahan aplikasi dalam aspek-aspek operasional, seperti startup, backup, dan proses recovery. | 0-5  |
| 13 | Tingkat Software dibuat untuk multi organisasi/perusahaan/client | Tingkat kebutuhan aplikasi dapat dioperasikan pada lingkungan hardware dan software yang berbedabeda.   | 0-5  |
| 14 | Tingkat kompleksitas dalam mengikuti perubahan/fleksibel         | Tingkat kemudahan aplikasi untuk modifikasi logika proses maupun struktur data.                         | 0-5  |

Keterangan Skor:

0 = Tidak berpengaruh

1 = Insidental

2 = Moderat

3 = Rata-rata

4 = Signifikan

5 = Essential

#### 2.3.4. Perhitungan Function Point

Nilai FP dapat digunakan untuk mengukur LOC perangkat lunak, cost, effort, resource yang dibutuhkan dan lama pengerjaan perangkat lunak (Balaji et al. 2013). Nilai Function Point (FP) dari perangkat lunak dihitung dengan menggunakan formula berikut.

$$FP = CFP \times (0.65 + 0.01 \times RCAF) \dots \dots \dots (1)$$

Sedangkan untuk memperoleh estimasi ukuran perangkat lunak dalam satuan Lines of Code (LOC), nilai FP dikalikan dengan Productivity Factor berdasarkan bahasa pemrograman yang akan digunakan. Productivity Factor adalah jumlah kode logis per function point dan nilainya bervariasi

untuk setiap bahasa pemrograman, seperti yang dibuat oleh Capers Jones pada Tabel 4.

Tabel 4 Productivity Factor Programming

| Programming Language | Productivity Factor |
|----------------------|---------------------|
| SQL                  | 37                  |
| PHP                  | 53                  |
| HTML/Javascript      | 58                  |

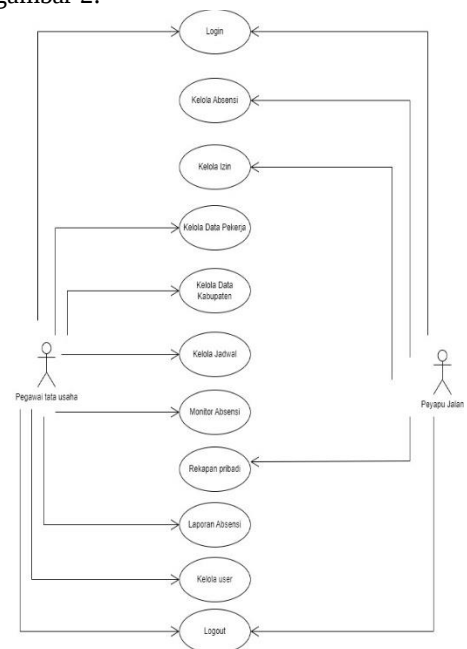
### 3. Hasil Penelitian dan Pembahasan

#### 3.1. Studi Literatur

Metode penelitian diawali dari studi literatur yang dilakukan dengan mempelajari materi melalui studi literatur jurnal penelitian terkait dengan metode Function Point yang dibutuhkan dalam pengembangan aplikasi yang sebagian besar diperoleh dari buku dan paper yang dicari melalui internet.

#### 3.2. Pengumpulan Data

Dalam pengumpulan data yang dilakukan melalui wawancara dan observasi diperoleh informasi terkait kebutuhan fungsionalitas yang diharapkan oleh pengguna pada aplikasi E-PresSaja pada gambar 2.



Gambar 2 Rancangan Kebutuhan fungsional sistem E-Pres Saja

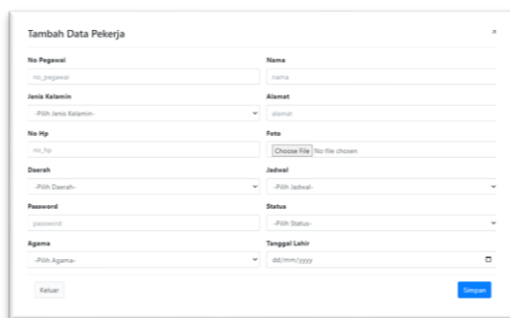
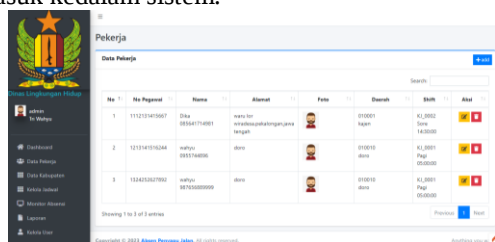
Berdasarkan rancangan kebutuhan fungsional sistem E-Pres Saja yang dibuat diketahui bahwa dalam sistem tersebut terdapat dua level pengguna yaitu bagian pegawai tata usaha yang bertugas untuk melakukan pengelolaan data presensi sapu jalan dan bagian penyapu jalan yang akan melakukan presensi kehadiran kerja. Hasil rancangan kebutuhan fungsional tersebut menjadi

dasar dalam pembuatan rancangan tampilan dari sistem E-Pres Saja.



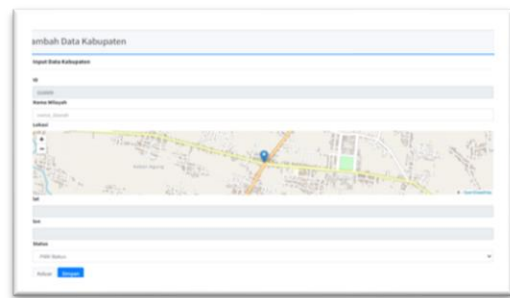
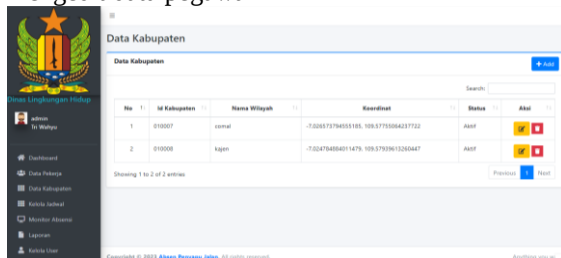
Gambar 1 Tampilan Login Pengguna

Gambar 3 merupakan rancangan tampilan login yang akan digunakan oleh pengguna untuk masuk kedalam sistem.



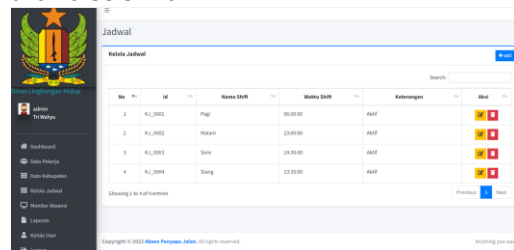
Gambar 2 Tampilan Mengelola Data Pekerja

Gambar 4 merupakan tampilan data pekerja yang digunakan oleh pegawai tata usaha untuk melakukan pengelolaan data penyapu jalan seperti melakukan pencarian, penambahan maupun mengedit data pegawai



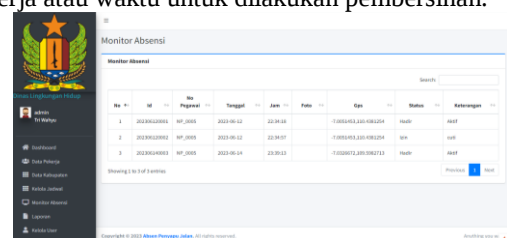
Gambar 3 Tampilan mengelola data Kabupaten

Gambar 5 merupakan rancangan tampilan mengelola data kabupaten yang digunakan oleh pegawai tata usaha untuk menambahkan, mengedit maupun mencari data kabupaten. Data kabupaten ini berfungsi untuk menentukan titik Lokasi yang akan dibersihkan



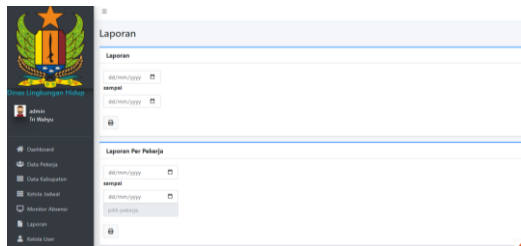
Gambar 4 Tampilan mengelola jadwal

Gambar 6 merupakan tampilan mengelola jadwal yang digunakan oleh pegawai tata usaha untuk menambahkan maupun mengedit jadwal kerja atau waktu untuk dilakukan pembersihan.



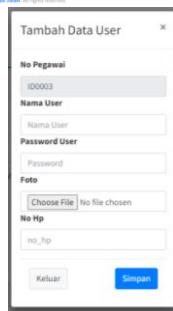
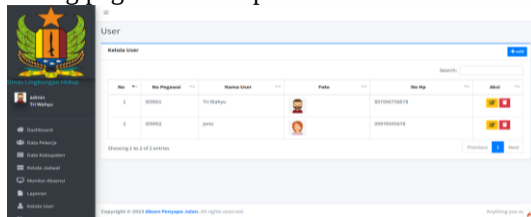
Gambar 5 Tampilan Monitor Absensi

Gambar 7 merupakan tampilan monitor Absensi yang digunakan oleh petugas tata usaha untuk memonitor absensi pegawai sapu jalan.



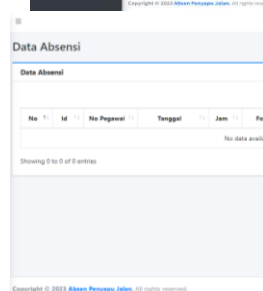
Gambar 6 Tampilan Laporan

Gambar 8 merupakan tampilan laporan yang digunakan oleh tata usaha untuk mencetak rekapan laporan kehadiran seluruh pegawai berdasarkan periode tertentu dan rekapan kehadiran dari masing masing pegawai dalam periode tertentu.



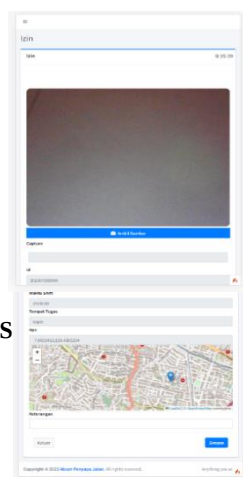
Gambar 7 Tampilan Kelola user

Gambar 9 merupakan tampilan Kelola user yang digunakan oleh petugas tata usaha untuk mengelola pembuatan akun pengguna agar dapat mengakses sistem E-pres saja.

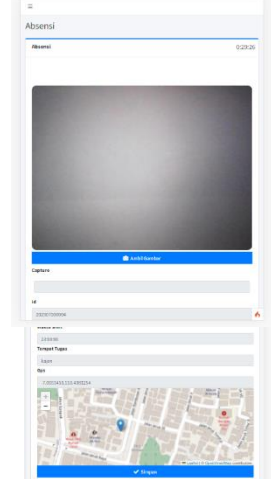


Gambar 8  
Tampilan Beranda  
Penyapu Jalan

Gambar 10 merupakan tampilan beranda penyapu jalan



yang digunakan oleh pegawai penyapu jalan untuk melihat jumlah kehadiran maupun izin.



Gambar 9 Tampilan melakukan presensi pegawai

Gambar 11 merupakan rancangan tampilan presensi pegawai yang digunakan oleh pegawai sapu jalan melakukan presensi kehadiran maupun izin kehadiran dengan cara mengambil foto dan mengirimkan titik Lokasi dan menambahkan keterangan jika melakukan izin.

### 3.3. Analisa dan Pemodelan dengan Metode FP

Pada dasarnya pengukuran merupakan kegiatan penentuan angka bagi suatu objek yang bertujuan untuk melakukan perkiraan dan untuk menyatakan kualitas produk. Pada pengembangan perangkat lunak, perlu adanya estimasi atau perkiraan ukuran suatu perangkat lunak yang akan dikembangkan untuk menentukan dan mengetahui berapa besar effort atau usaha yang akan dikeluarkan untuk mengembangkan perangkat lunak. Dalam penelitian ini Metode yang digunakan dalam melakukan estimasi biaya pengembangan perangkat lunak adalah metode function point. Metode Function Point (FP) merupakan salah satu metode yang digunakan untuk mendapatkan ukuran fungsionalitas. Tujuan dari pengukuran menggunakan FP adalah untuk mendapatkan ukuran biaya, durasi, dan jumlah sumber daya yang diperlukan oleh sebuah proyek perangkat lunak dan dapat dilakukan pada setiap tahap pengembangan perangkat lunak dalam hal ini yaitu pengembangan aplikasi E-Pres Saja

Berikut merupakan data – data yang digunakan dalam perancangan sistem estimasi biaya pengembangan perangkat lunak

#### 3.3.1. Identifikasi 5 Komponen Utama

Tabel 4 merupakan identifikasi dari tipe fungsi pengguna atau kebutuhan fungsional yang dirancang pada pengembangan aplikasi E-Pres saja yang akan digunakan sebagai komponen utama dalam menentukan nilai CFP dalam perhitungan

estimasi proyek menggunakan metode function point.

Tabel 1 Tipe Fungsi Pengguna

|                          |                                                                                                                                                                                                                                                                                                                         |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tipe input               | <ul style="list-style-type: none"> <li>- Login</li> <li>- Mengelola data user</li> <li>- Mengelola data pekerja</li> <li>- Mengelola data kabupaten</li> <li>- Mengelola jadwal</li> <li>- Mengelola pembagian jadwal</li> <li>- Penginputan presensi kehadiran</li> <li>- Penginputan izin kehadiran</li> </ul>        |
| Tipe output              | <ul style="list-style-type: none"> <li>- Menampilkan data pekerja</li> <li>- Menampilkan data kabupaten</li> <li>- Menampilkan pembagian jadwal</li> <li>- Menampilkan data monitor presensi</li> <li>- Menampilkan laporan presensi per periode</li> <li>- Menampilkan laporan presensi berdasarkan pekerja</li> </ul> |
| Tipe Query/Search/View   | <ul style="list-style-type: none"> <li>- Melakukan pencarian data pekerja</li> <li>- Menghitung rekap presensi kehadiran dalam periode bulan</li> <li>- Menghitung rekap presensi izin dalam periode bulan</li> <li>- Menampilkan monitoring presensi berdasarkan karyawan</li> </ul>                                   |
| Tipe File/Tabel/Database | <ul style="list-style-type: none"> <li>- Data user</li> <li>- Data pekerja</li> <li>- Data kabupaten</li> <li>- Data jadwal</li> <li>- Data pembagian jadwal</li> <li>- Data presensi</li> </ul>                                                                                                                        |
| Tipe Interface Eksternal | <ul style="list-style-type: none"> <li>- Mengambil data Lokasi saat melakukan presensi kehadiran maupun izin</li> </ul>                                                                                                                                                                                                 |

### 3.3.2. Menghitung CFP (Crude Function Points)

Setelah mengidentifikasi tiap komponen utama dari kebutuhan fungsional sistem selanjutnya yaitu mengevaluasi kuantitas bobot kerumitan dari tiap komponen tersebut dan mengelompokan tiap komponen berdasarkan bobot kompleksitasnya. Kemudian menghitung jumlah dari tipe fungsi pengguna berdasarkan bobot kompleksitasnya

dikalikan dengan bobot kompleksitas. Setelah dilakukan pembobotan yang berdasarkan kompleksitasnya, maka dapat dilakukan perhitungan Crude Function Point (CFP) yaitu Hasil dari perkalian pembobotan tersebut kemudian dijumlahkan sehingga hasil dari proses tersebut menjadi angka CFP. Hasil dari tahapan perhitungan CFP dapat dilihat pada tabel perhitungan CFP dibawah ini.

Tabel 2. Perhitungan CFP

| Tipe<br>Komponen                      | Jml | Level Kompleksitas |       |       |          |       |       |          |       |       | Total |
|---------------------------------------|-----|--------------------|-------|-------|----------|-------|-------|----------|-------|-------|-------|
|                                       |     | Sederhana          |       |       | Menengah |       |       | Kompleks |       |       |       |
|                                       |     | Jml                | Bobot | Point | Jml      | Bobot | Point | Jml      | Bobot | Point |       |
| External<br>Input (EI)                | 8   | 1                  | 3     | 3     | 3        | 4     | 12    | 4        | 6     | 24    | 39    |
| Internal<br>Outputs<br>(EO)           | 6   | 2                  | 4     | 8     | 2        | 5     | 10    | 2        | 7     | 14    | 32    |
| External<br>Inquiries<br>(EQ)         | 4   | 1                  | 3     | 3     | 0        | 4     | 0     | 3        | 6     | 18    | 21    |
| Internal<br>Logical<br>Files<br>(ILF) | 1   | 0                  | 7     | 0     | 1        | 10    | 10    | 0        | 15    | 0     | 10    |
| External<br>Interface<br>File (EIF)   | 6   | 3                  | 5     | 15    | 1        | 7     | 7     | 2        | 10    | 20    | 42    |
| Total<br>(CFP=UFP)                    |     |                    |       |       |          |       |       |          |       |       | 44    |

Berdasarkan pada Tabel 5. maka dapat diketahui nilai total dari Crude Function Point (CFP) sebesar 144

### 3.3.3. Menghitung RCAF (Relative Complexity Adjustment Factor)

Pada tahapan RCAF untuk pengguna dalam melakukan estimasi dengan tujuan untuk menghitung nilai RCAF dari aplikasi yang akan diestimasi pada tabel dibawah ini

Tabel 7 Perhitungan RCAF

| No         | Karakteristik                                                    | Skor |
|------------|------------------------------------------------------------------|------|
| 1          | Tingkat kompleksitas Komunikasi Data                             | 5    |
| 2          | Tingkat kompleksitas Pemrosesan Terdistribusi                    | 5    |
| 3          | Tingkat kompleksitas Performance                                 | 4    |
| 4          | Tingkat kompleksitas Konfigurasi                                 | 3    |
| 5          | Tingkat Frekuensi Penggunaan Software                            | 3    |
| 6          | Tingkat Frekuensi Input Data                                     | 4    |
| 7          | Tingkat Kemudahan Penggunaan Bagi User                           | 4    |
| 8          | Tingkat Frekuensi Update Data                                    | 4    |
| 9          | Tingkat Kompleksitas Prosesing Data                              | 4    |
| 10         | Tingkat Kemungkinan Penggunaan Kembali/Reusable Kode Program     | 2    |
| 11         | Tingkat Kemudahan Dalam Instalasi                                | 0    |
| 12         | Tingkat Kemudahan operasional software (backup, recovery, dsb)   | 1    |
| 13         | Tingkat Software dibuat untuk multi organisasi/perusahaan/client | 0    |
| 14         | Tingkat kompleksitas dalam mengikuti perubahan/fleksibel         | 3    |
| Total RCAF |                                                                  | 42   |

Berdasarkan pada Tabel 7. maka dapat diketahui nilai total RCAF sebesar 42, yang didapat dari total skala dalam 14 karakteristik tersebut.

### 3.3.4. Perhitungan Function Point

Setelah mengetahui nilai Crude Function Point (CFP) pada Tabel 6. dan nilai Relative Complexity Adjustment Factor (RCAF) pada Tabel 7. maka langkah selanjutnya menentukan nilai Function Point (FP). Nilai kompleksitas akan berbeda dari satu perangkat lunak ke perangkat lunak lainnya. Nilai Function Point ini akan merupakan petunjuk terhadap besarnya ukuran perangkat lunak. Semakin besar nilai Function Point akan semakin besar pula ukuran perangkat lunaknya.

Function Point (FP) yaitu proses melakukan perhitungan untuk mendapatkan nilai Function Point dari suatu sistem yang akan dikembangkan. Untuk menghitung Function Point (FP) maka dapat menggunakan rumus  $FP = CFP \times (0.65 + 0.01 \times RCAF)$ . Angka 0,65 dan 0,01 merupakan angka yang konstanta atau ketetapan yang dibuat oleh International Function Point Users Group (IFPUG). Berikut perhitungan dalam menentukan nilai FP:

$$\begin{aligned} FP &= CFP \times (0,65 + (0,01 \times RCAF)) \\ &= 144 \times (0,65 + (0,01 \times 42)) \\ &= 144 \times (0,65 + 0,42) \\ &= 154,08 \end{aligned}$$

Jadi, nilai FP pada Aplikasi E-Pres saja adalah 154,08.

Dari nilai FP tersebut digunakan untuk menentukan estimasi diantaranya:

#### 1. Menentukan Estimasi Ukuran Proyek

Untuk mendapatkan nilai estimasi ukuran dari sistem dalam Lines of Codes (LOC), maka nilai FP dikalikan dengan 53 yang merupakan productivity factor dari bahasa pemrograman PHP, sehingga LOC didapatkan seperti pada formula  $LOC = 154,08 \times 53 = 8166,24$

Ukuran proyek telah diestimasi dengan nilai estimasi 8166,24 baris kode dengan menggunakan metode FPA

#### 2. Menentukan Estimasi Effort

Untuk menghitung estimasi effort yang dibutuhkan dalam pengembangan sistem maka nilai LOC yang diperoleh dibagi 1000 baris code kemudian dikalikan dengan 1.4 yang merupakan angka yang konstanta atau ketetapan yang dibuat oleh International Function Point Users Group (IFPUG).

$$\begin{aligned} \text{Effort} &= 1.4 \times 8.16624 \\ &= 11,43 \text{ (in Person- Months)} \end{aligned}$$

Artinya sistem E-Pres saja dapat diselesaikan dengan estimasi effort sebesar 11,43 man/month

#### 3. Perhitungan Estimasi Waktu

Untuk menghitung estimasi waktu yang dibutuhkan dalam pengembangan sistem dihitung dengan formula berikut:

$$\begin{aligned} \text{Time} &= 3.0 \times \text{Effort}^{1/3} \text{ (in Months)} \dots\dots\dots (2) \\ \text{Time} &= 3.0 \times 11,43^{1/3} \\ &= 6,7 \text{ (in Months)} \end{aligned}$$

Artinya estimasi waktu yang dibutuhkan dalam pengembangan sistem E-pres saja membutuhkan waktu sekitar 6,7 bulan.

#### 4. Menentukan Estimasi Biaya

Berdasarkan hasil perhitungan function point, dapat ditentukan estimasi biaya terhadap software E-Pres saja. Untuk harga 1 poin, penulis menetapkan harga senilai Rp 318.182 Estimasi biaya didasarkan pada website dari <https://idstar.co.id/> yang menyebutkan bahwa Gaji rata-rata web developer di Indonesia berkisar Rp 7.000.000. Oleh karena itu, harga software E-Press saja ditetapkan sebagai berikut:

##### a. Gaji/hari Web Developer

Jumlah hari kerja selama satu bulan adalah 22 hari dengan 5 hari kerja dan 2 hari libur. Maka ditetapkan gaji/hari sebagai berikut:  
 $Rp\ 7.000.000 : 22 = Rp\ 318.182$

##### b. Gaji/hari Web Developer

Jumlah jam kerja selama satu hari adalah 8 jam. Maka ditetapkan gaji/jam sebagai berikut:  
 $Rp\ 318.182 : 8 = Rp\ 39.772,75$

##### c. Harga/poin pada hasil perhitungan function point

Dari perhitungan gaji/jam web developer, maka dapat diketahui harga dari website pendataan ekstrakurikuler siswa sebagai berikut:

$$Rp\ 318.182 \times 154,08 = Rp\ 49.025.482$$

Berdasarkan hasil perhitungan poin-poin penilaian dengan harga yang diberikan, dihasilkan estimasi biaya pengembangan perangkat lunak E Pres saja sebesar harga Rp 49.025.482

## 4. Kesimpulan

Berdasarkan dari penelitian yang telah dilakukan yaitu menerapkan metode function point dalam perencanaan pengembangan dapat diketahui estimasi ukuran proyek yang akan dibangun, estimasi waktu, dan biaya yang dibutuhkan dalam pengembangan sistem E – Pressaja. Dari hasil penelitian diketahui hasilnya yaitu bahwa ukuran proyek yang akan dikembangkan sebesar 8166,24 LOC dengan waktu yang dibutuhkan dalam pengembangan sistem ini yaitu 6,7 bulan dan estimasi biaya yang dibutuhkan sebesar 49.025.482. diharapkan penelitian kedepan dapat dilakukan estimasi proyek dengan metode lain seperti use case point atau metode yang lainnya untuk membandingkan hasil dari metode tersebut sehingga dapat diketahui tingkat akurasi dari masing-masing metode.

## 5. Referensi

Balaji, By N, N Shivakumar, V Vignaraj Ananth, N Balaji α, N Shivakumar σ, and V Vignaraj Ananth ρ. 2013. “Software Cost Estimation Using Function Point with Non Algorithmic

- Approach Software Cost Estimation Using Function Point with Non Algorithmic Approach Software Cost Estimation Using Function Point with Non Algorithmic Approach.” *Type: Double Blind Peer Reviewed International Research Journal Publisher: Global Journals Inc* 13 (8).
- Fikri, Rijal Muhammad, and Jane Sekarsari. 2015. “Analisis Estimasi Biaya Proyek Peningkatan Jalan Beton Di Kabupaten Tangerang Dengan Metode Cost Significant Model MK-337 MK-338.” *Seminar Nasional Teknik Sipil V Tahun 2015 – UMS* 5 (2010): 1–12.
- Grandys Frieska Prassida, Achmad Holil Noor Ali, Sholiq. 2018. “Estimasi Biaya Pembuatan Modul Enterprise Resource Planning (ERP) Untuk Unit Bisnis Pabrik Gula Di PT. Perkebunan XYZ Dengan Metode Use Case Point Grandys,” no. September 2012.
- Maddeppungeng, Andi, and Irma Suryani. 2016. “ESTIMASI BIAYA PADA PROYEK PERUMAHAN (Studi Kasus Proyek Pembangunan Citra Serang Residence).” *Fondasi : Jurnal Teknik Sipil* 5 (1): 79–86. <https://doi.org/10.36055/jft.v5i1.1249>.
- Permatasari, Rochma. 2016. “Lunak Menggunakan Metode Function Point Dengan Masukan Dokumen Use Case , Data Flow Diagram , Dan Natural Language Project Effort Estimation Using Function Point Method With Use Case Document , Data Flow Diagram , and Natural Language.” *Institut Teknologi Sepuluh Nopember* 2 (3): 1–462.
- Sandhea, Annisa Azzar, Sri Maulidiyah, and Renny Dewi Sari. 2020. “Estimasi Biaya Perangkat Lunak Pada Aplikasi SIBIMA Universitas XYZ Dengan Menggunakan Metode Function Point.” *JURIKOM (Jurnal Riset Komputer)* 7 (1): 41. <https://doi.org/10.30865/jurikom.v7i1.1893>.
- Saptono, Ristu, and Rini Anggrainingsih. 2016. “Development of Software Size Estimation Application Using Function Point Analysis (FPA) Approach with Rapid Application Development (RAD).” *Jurnal Ilmiah Teknologi Dan Informasi* 5 (2): 96–102.
- Saptono, Ristu, and Galih Dian Hutama. 2016. “Peningkatan Akurasi Estimasi Ukuran Perangkat Lunak Dengan Menerapkan Logika Samar Metode Mamdani.” *Scientific Journal of Informatics* 2 (1): 41–51. <https://doi.org/10.15294/sji.v2i1.4527>.
- Sevilla, Jorge, Luis Ignacio Jiménez, and Antonio Plaza. 2015. “Sparse Unmixing-Based Content Retrieval of Hyperspectral Images on Graphics Processing Units.” *IEEE Geoscience and Remote Sensing Letters* 12 (12): 2443–47. <https://doi.org/10.1109/LGRS.2015.2483679>.
- Widyastuti, Susi, Ariandi, Wahyu, & Sulistiono, Vergamana. 2019. “View of Implementasi Kriptografi AES Dalam Pengamanan Data Seleksi Peserta JAMKESMAS.Pdf.”.