

Rancang Bangun Sistem Manajemen Keuangan Pribadi Menggunakan Model *Budget Jar* Berbasis Android

Hajna Agung Milkham ^{1*}, Petrus Sokibi ², Amroni ³

Program Studi Teknik Informatika, Universitas Catur Insan Cendekia, Cirebon, Indonesia
¹hajna.milkham.ti.18@cic.ac.id; ²petrus.sokibi@cic.ac.id; ³amroni@cic.ac.id
* Penulis Korespondensi

ABSTRAK

Mengelola anggaran pendapatan dan pengeluaran merupakan langkah dalam mencegah terjadi permasalahan finansial di masa depan. Dalam mengelola keuangan, seringkali sulit mempersiapkan anggaran kebutuhan dan anggaran keinginan sesuai pada porsinya. Tingkat keinginan yang terlalu tinggi dan minim kontrol dalam membatasi anggaran sering membuat pengeluaran biaya jauh di luar anggaran yang sudah ditetapkan. Penelitian ini dilakukan untuk memperkenalkan suatu model pengelola keuangan bernama *budget jar* dengan mengintegrasikan ke dalam sistem manajemen yang terkomputerisasi berbasis android. Model ini akan membagi anggaran ke dalam enam pos pengeluaran dengan tujuan tertentu. Sistem yang dihasilkan dari penelitian ini dapat membantu pengguna dalam mengatur anggaran dalam periode tertentu dan dapat digunakan dalam mencatat pengeluaran yang memiliki urgensi terkait upaya manajemen keuangan.

Riwayat Artikel

Diterima 29 Agustus 2023
Diperbaiki 25 Maret 2024
Diterbitkan 27 Maret 2024

Kata Kunci

Budget Jar
Finansial
Informasi
Manajemen Keuangan
Sistem



This is an open-access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license

1. Pendahuluan

Mengelola anggaran pendapatan dan mengatur pengeluaran merupakan salah satu langkah dalam mencegah terjadinya permasalahan finansial di masa depan. Dalam mengelola keuangan, seringkali sulit bagi kita untuk mempersiapkan anggaran kebutuhan dan anggaran keinginan sesuai pada porsinya. Tingkat keinginan yang terlalu tinggi dan minimnya kontrol dalam membatasi anggaran sering membuat kita mengeluarkan biaya jauh diluar anggaran yang sudah ditetapkan.

Berdasarkan studi komparasi yang dilakukan sebelumnya, diketahui bahwa terdapat beberapa aplikasi yang digunakan untuk membantu penggunaannya dalam melakukan pencatatan keuangan dengan fungsinya yang memiliki keunggulan dan kekurangan masing-masing [1]. Pembuatan aplikasi manajemen keuangan pribadi sebelumnya pernah dilakukan oleh Silvia Ratna, dimana aplikasi keuangan yang berbasis *desktop* dapat digunakan dalam melakukan pencatatan pemasukan dan pengeluaran sehari-hari dan membuat laporan keuangan dengan lebih mudah [2].

Analisis permasalahan berdasarkan penelitian sebelumnya adalah belum adanya fitur pada aplikasi yang dapat membantu pengguna dalam menetapkan anggaran dari pendapatan yang ada. Penetapan anggaran ini penting dilakukan agar pengguna dapat lebih mudah dalam mengatur setiap pos pengeluaran dan dapat membedakan pengeluaran yang harus diprioritaskan [3].

Berbeda dari penelitian sebelumnya, sistem yang akan dibangun pada penelitian ini berfokus pada pengalokasian dana pendapatan dengan menggunakan model *budget jar*, dimana pendapatan akan dibagi ke dalam enam pos anggaran yaitu 50% anggaran Kebutuhan Pokok, 10% anggaran Tabungan Jangka Panjang, 10% anggaran investasi, 10% anggaran Pendidikan, 10% anggaran Hiburan, dan 10% anggaran untuk Amal [4]. Sistem akan dibangun dengan menggunakan model pengembangan sistem *Extreme Programming*. Model *Extreme Programming* digunakan untuk efisiensi dalam hal waktu pengembangan, prosesnya sederhana, dan *feedback* yang konstan sehingga mempermudah perbaikan terhadap kendala pada sistem.

Pengembangan sistem akan dilakukan dengan menggunakan bahasa pemrograman Java yang tersedia pada *Android Studio*. *Android Studio* merupakan sebuah *tools* yang secara resmi diciptakan dan direkomendasikan *Google.inc* untuk mengembangkan aplikasi berbasis *Android* [5]. Bahasa

pemrograman yang tersedia pada *Android Studio* secara umum diketahui yaitu bahasa pemrograman Java dan bahasa pemrograman Kotlin. Peneliti memilih untuk menggunakan bahasa pemrograman Java karena dinilai memiliki kelebihan yaitu karakteristik yang sederhana dan *portable* sehingga membuat *source code* dan juga pemrograman dapat dengan mudah ditransfer ke *platform* yang berbeda-beda. Pemrograman Java sendiri didefinisikan sebagai nama dari sekumpulan teknologi untuk membangun dan menjalankan perangkat lunak dalam lingkungan komputer atau jaringan yang berdiri sendiri [6].

Menilai dari segi kemudahan dalam perancangan *database* dan kemudahan dalam implementasi, penulis memilih SQLite sebagai *database* yang akan digunakan dalam penelitian ini. SQLite memiliki kelebihan dimana *database* ini mengizinkan kita dalam meakses beberapa *files database* secara bersamaan, sehingga memungkinkan untuk menggabungkan tabel-tabel pada *database* berbeda atau menyalin data antar *database* dalam hal perintah [7].

Tujuan akhir dari penelitian ini adalah terciptanya sebuah sistem yang dapat membantu pengunanya dalam mengelola anggaran dan pencatatan transaksi sehingga pengguna dapat lebih mudah dalam menentukan pos pengeluaran yang perlu diprioritaskan. Sistem manajemen keuangan pribadi ini juga diharapkan dapat memberikan edukasi terhadap penggunanya tentang pentingnya mengatur anggaran dari pendapatan agar tidak mengalami permasalahan finansial di masa depan.

2. Metode

Penelitian ini tergolong sebagai penelitian kualitatif. Metode penelitian kualitatif adalah metode penelitian yang bertujuan untuk mendapatkan pemahaman tentang kenyataan melalui proses berfikir induktif dimana peneliti terlibat dalam situasi dan *setting* fenomena yang diteliti [8]. Data yang digunakan pada penelitian ini bersumber dari respon angket, wawancara, studi pustaka, dan observasi. Pengembangan perangkat lunak akan dilakukan dengan menggunakan model *Extreme Programming*.

Tahapan dari model pengembangan *Extreme Programming* yang disampaikan oleh [9] adalah sebagai berikut:

1. *Planning*

Tahap perencanaan dimulai dengan pemahaman konteks bisnis dari aplikasi, mendefinisikan *output*, fitur yang ada pada aplikasi, fungsi dari aplikasi yang dibuat, serta alur pengembangan aplikasi.

2. *Design*

Tahap ini berfokus pada *design* aplikasi secara sederhana. Alat untuk mendesain pada tahap ini dapat menggunakan CRC (*Class Responsibility Collaborator*). CRC memetakan kelas-kelas yang akan dibangun dalam *use case diagram*, *class diagram*, dan *activity diagram*.

3. *Coding*

Coding atau pengkodean merupakan penerjemahan dari perancangan dalam Bahasa pemrograman yang dikenali oleh komputer.

4. *Testing*

Sebelum dirilis, sistem akan diuji terlebih dahulu apakah sudah sesuai dengan kebutuhan yang ditentukan pada tahap perencanaan dan untuk menghindari *error* pada saat dipublikasikan.

Model uji coba yang digunakan dalam penelitian ini adalah *Blackbox Testing*. *Blackbox Testing* merupakan tahapan uji coba sistem yang berfokus pada spesifikasi fungsional dari perangkat lunak. Penguji akan mencoba sistem kemudian akan memeriksa apakah dengan masukan yang diberikan akan muncul keluaran seperti yang diharapkan [10].

3. Hasil dan Pembahasan

3.1. Hasil Penelitian

Fitur yang ada pada sistem informasi ini dibuat berdasarkan *user stories* yang didapat dari respon pada angket yang disebarkan melalui media sosial. Interpretasi respon pada angket akan ditarik kesimpulan menjadi beberapa *user stories* seperti yang tersaji pada Tabel 1.

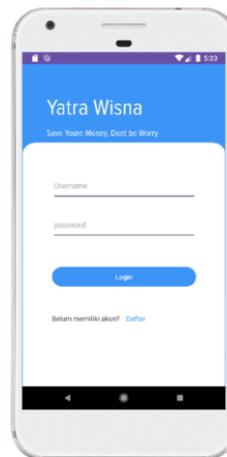
Tabel 1. User Stories Responden

Nomor	User Stories	Value
US-01	Sebagai <i>user</i> , saya ingin sistem ini dapat mengingatkan saya terkait hutang-piutang yang saya punya.	2
US-02	Sebagai <i>user</i> , saya ingin sistem ini dapat mengingatkan saya dalam mencatat pengeluaran dan mengingatkan saya dalam pengeluaran yang melebihi anggaran.	4
US-03	Sebagai <i>user</i> , saya ingin sistem ini bisa membantu dalam mencatat keuangan dan menyimpannya agar data tidak hilang.	2
US-04	Sebagai <i>user</i> , saya ingin sistem ini bisa menunjukkan berapa anggaran yang saya miliki.	4
US-05	Sebagai <i>user</i> , saya ingin sistem ini bisa memantau alur transaksi pengeluaran saya.	3

Berdasarkan *user stories* pada Tabel 1, ditetapkan beberapa fitur yang dapat digunakan oleh *user* dalam mengelola keuangan. Fitur-fitur tersebut diantaranya:

1. Kelola pendapatan,
2. Kelola transaksi,
3. Catatan hutang,
4. Laporan keuangan,
5. Pengingat untuk transaksi yang melebihi batas anggaran.

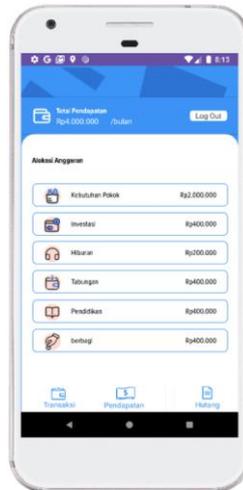
Lima fitur di atas dapat diakses setelah melakukan *login* pada sistem. halaman *login user* dapat dilihat pada Gambar 1.

**Gambar 1. Halaman Login**

Saat membuka aplikasi untuk pertama kali, *user* akan masuk ke halaman *login* terlebih dahulu. *User* tidak dapat mengakses halman menu sebelum melakukan *login*. Apabila *user* belum memiliki akun, *user* dapat pindah ke halaman *Sign Up* dengan menekan “Daftar” yang ada di bawah *button Login*.

3.1.1. Halaman Menu

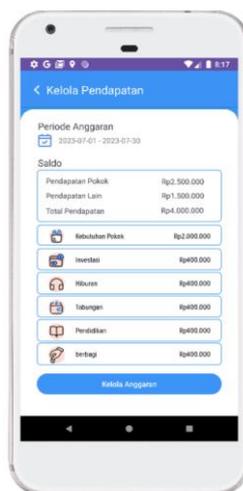
Halaman Menu adalah halaman yang akan ditampilkan setelah *user* melakukan *login*. Pada halaman ini, terdapat informasi terkait besar pendapatan *user*, anggaran yang dimiliki *user*, serta beberapa *button* yang menghubungkan menu utama ke menu lainnya seperti menu kelola pendapatan, kelola transaksi dan kelola hutang. Informasi terkait anggaran dan pendapatan *user* akan ditampilkan setelah *user* mengisi data pendapatan pada halaman Kelola Pendapatan seperti yang tersaji pada Gambar 2.



Gambar 2. Halaman Menu

3.1.2. Halaman Kelola Pendapatan

Proses yang dapat dilakukan oleh *user* pada halaman ini adalah proses Kelola Anggaran, dimana *user* dapat memasukkan data pendapatan yang akan diproses oleh sistem menjadi pos anggaran yang disesuaikan berdasarkan pembagian anggaran pada model *Budget Jar*. Setelah proses Kelola Anggaran ini dilakukan, pembagian anggaran dikeluarkan sebagai *output* pada halaman Kelola Pendapatan seperti tersaji pada Gambar 3.

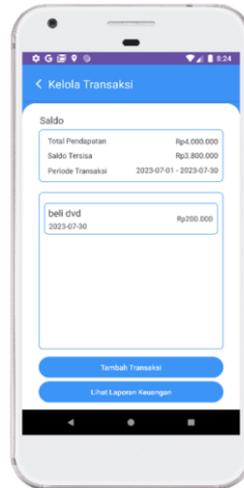


Gambar 3. Halaman Kelola Pendapatan

3.1.3. Halaman Kelola Transaksi

User dapat melakukan beberapa proses pada halaman Kelola Transaksi, sebagai berikut:

1. Proses Pencatatan Transaksi, dimana *user* dapat mencatat transaksi dengan menekan *button* Tambah Transaksi seperti yang ditampilkan pada Gambar 4. *Output* dari proses ini adalah detail transaksi yang dicatat oleh *user* melalui sistem.
2. Membuat laporan keuangan. Laporan keuangan bisa dibuat dengan menekan *button* Lihat Laporan Keuangan. Laporan keuangan mencakup data pendapatan, sisa anggaran, dan transaksi yang dicatat oleh *user* ke dalam sistem. Laporan keuangan dapat disimpan kedalam bentuk pdf.



Gambar 4. Halaman Kelola Transaksi

3.1.4. Halaman Hutang

Halaman hutang dapat diakses *user* melalui halaman menu. *User* dapat melakukan proses pencatatan hutang pada halaman ini. Proses pencatatan hutang dimulai dengan menekan *button* Tambah Hutang seperti yang tersaji pada Gambar 5. *User* akan dipindahkan ke halaman Tambah Hutang setelah menekan *button*. *User* akan diminta mengisi *form* untuk mencatat hutang baru.



Gambar 5 Halaman Hutang

3.2. Pembahasan

Berdasarkan hasil uji *blackbox*, terdapat beberapa hal yang perlu dibahas sebagai berikut:

3.2.1. Pengujian Halaman Login

Pada pengujian halaman *login* terdiri dari beberapa tahapan, diawali dengan memasukkan data secara lengkap pada *form* untuk melakukan *login* sampai dengan memasukkan berbagai variabel masukan yang berbeda agar diketahui pesan *error* dapat muncul apabila data tidak sesuai.

3.2.2. Pengujian Halaman Kelola Pendapatan

Pengujian pada halaman ini dimulai dengan pengujian *button* navigasi menuju halaman Kelola Anggaran. Pengujian ini menunjukkan bahwa *button* navigasi berfungsi dengan baik. Pengujian berlanjut dengan mengisi *form* pengisian data pada halaman Kelola Anggaran. Hasil pengujian pada *form* ini dapat dilihat melalui hasil pengujian *blackbox* seperti yang tersaji pada Tabel 2.

Tabel 2. Hasil Uji Blackbox Halaman Kelola Pendapatan

No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Masuk ke Halaman Kelola Pendapatan sebelum melakukan proses Kelola Anggaran	Menampilkan data kosong pada halaman Kelola Pendapatan	Menampilkan data kosong pada halaman Kelola Pendapatan	<i>Valid</i>
2	Memasukan variabel masukan kosong tapi menekan <i>button</i> Atur Anggaran	Muncul pesan “Masih ada yang kosong”	Muncul pesan “Masih ada yang kosong”	<i>Valid</i>
3	Memasukan secara bergantian variabel kosong pada kolom isian tapi menekan <i>button</i> Atur Anggaran	Muncul pesan “Masih ada yang kosong”	Muncul pesan “Masih ada yang kosong”	<i>Valid</i>
4	Memasukan variabel sesuai dengan ketentuan kolom isian lalu menekan <i>button</i> Atur Anggaran	Muncul pesan “anggaran berhasil di- <i>input</i> ”, pindah ke halaman Kelola pendapatan	Muncul pesan “anggaran berhasil di- <i>input</i> ”, pindah ke halaman Kelola Pendapatan	<i>Valid</i>
5	Masuk ke Halaman Kelola Pendapatan setelah melakukan proses Kelola Anggaran	Menampilkan data dari tabel anggaran	Menampilkan data dari tabel anggaran	<i>Valid</i>
6	Menekan <i>button</i> Kelola Anggaran saat masih ada periode yang sedang berlangsung	Menampilkan pesan “Periode anggaran belum berakhir. Yakin untuk mengganti periode anggaran? Ya/Tidak”	Menampilkan pesan “Periode anggaran belum berakhir. Yakin untuk mengganti periode anggaran? Ya/Tidak”	<i>Valid</i>

3.2.3. Pengujian Halaman Kelola Transaksi

Pengujian pada halaman ini dimulai dengan pengujian *button* navigasi menuju halaman Tambah Transaksi dan Laporan Keuangan. Pengujian ini menunjukkan bahwa kedua *button* navigasi berfungsi dengan baik. Pengujian berlanjut dengan mengisi *form* pengisian data pada halaman Tambah Transaksi. Hasil pengujian pada *form* ini dapat dilihat melalui hasil pengujian *blackbox* seperti tersaji pada Tabel 3.

Tabel 3. Hasil Uji Blackbox Halaman Kelola Transaksi

No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Memasukan variabel masukan kosong tapi menekan <i>button</i> Tambah Transaksi	Muncul pesan “Masih ada yang kosong”	Muncul pesan “Masih ada yang kosong”	<i>Valid</i>
2	Memasukan secara bergantian data kosong pada <i>form</i> tapi menekan <i>button</i> Tambah Transaksi	Muncul pesan “Masih ada yang kosong”	Muncul pesan “Masih ada yang kosong”	<i>Valid</i>
3	Memasukan variabel jumlah transaksi melebihi sisa anggaran tertentu	Muncul pesan “Anggaran yang tersisa pada @kategori sebesar @sisa_anggaran. Anggaran tidak mencukupi.”	Muncul pesan “Anggaran yang tersisa pada @kategori sebesar @sisa_anggaran. Anggaran tidak mencukupi.”	<i>Valid</i>
4	Memasukan variabel sesuai dengan ketentuan kolom isian lalu menekan <i>button</i> Tambah Transaksi	Muncul pesan “Transaksi berhasil di- <i>input</i> ”, pindah ke halaman Kelola Transaksi	Muncul pesan “Transaksi berhasil di- <i>input</i> ”, pindah ke halaman Kelola Transaksi	<i>Valid</i>
5	Masuk ke halaman Kelola Transaksi setelah melakukan proses Tambah Transaksi	Memunculkan item transaksi sesuai dengan data transaksi yang baru ditambahkan pada	Memunculkan item transaksi sesuai dengan data transaksi yang baru ditambahkan pada	<i>Valid</i>

No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
		kolom Riwayat Transaksi.	kolom Riwayat Transaksi.	
5	Menekan <i>button</i> Lihat Laporan Keuangan	Pindah ke Halaman Laporan Keuangan, muncul data laporan keuangan	Pindah ke Halaman Laporan Keuangan, muncul data laporan keuangan	Valid
6	Menekan item transaksi pada Riwayat transaksi tertentu	Pindah ke halaman Detail Transaksi dan menampilkan data detail transaksi	Pindah ke halaman Detail Transaksi dan menampilkan data detail transaksi	Valid

3.2.4. Pengujian Halaman Hutang

Pengujian pada halaman ini dimulai dengan pengujian *button* navigasi menuju halaman Tambah Hutang. Pengujian ini menunjukkan bahwa *button* navigasi berfungsi dengan baik. Pengujian berlanjut dengan mengisi *form* pengisian data pada halaman Tambah Hutang. Hasil pengujian pada *form* ini dapat dilihat melalui hasil pengujian *blackbox* seperti tersaji pada Tabel 4.

Tabel 4. Hasil Uji Blackbox Halaman Hutang

No.	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Memasukan variabel masukan kosong tapi menekan <i>button</i> Tambah Hutang	Muncul pesan “Masih ada yang kosong”	Muncul pesan “Masih ada yang kosong”	Valid
2	Memasukan secara bergantian data kosong pada <i>form</i> tapi menekan <i>button</i> Tambah Hutang	Muncul pesan “Masih ada yang kosong”	Muncul pesan “Masih ada yang kosong”	Valid
3	Memasukan variabel sesuai dengan ketentuan kolom isian lalu menekan <i>button</i> Tambah Hutang	Muncul pesan “Transaksi berhasil diinput”, pindah ke halaman Hutang	Muncul pesan “Transaksi berhasil diinput”, pindah ke halaman Hutang	Valid

3.3. Manfaat Sistem informasi Manajemen Keuangan Pribadi

Adapun manfaat dari rancang bangun sistem manajemen keuangan ini, diantaranya adalah:

- Membantu *user* dalam mengalokasikan anggaran dan melakukan pencatatan keuangan
- Sistem dapat membantu penggunaanya dalam mengingatkan sisa anggaran yang ada.
- Sistem dapat mengingatkan pengguna saat bertransaksi yang melebihi sisa anggaran.

4. Kesimpulan

Berdasarkan pembahasan yang telah diuraikan sebelumnya, dapat ditarik kesimpulan sebagai berikut:

- Sistem manajemen sudah dapat digunakan dalam mengalokasikan anggaran sesuai dengan model pengelola keuangan *budget jar*.
- Sistem manajemen sudah dapat menampilkan peringatan saat *user* melakukan transaksi yang melebihi batasan dari anggaran yang sudah ditetapkan.
- Model pengelolaan anggaran *Budget jar* telah berhasil diterapkan dalam *algoritma* yang terdapat pada sistem manajemen keuangan ini.
- Mutasi anggaran telah dapat ditinjau oleh *user* melalui laporan keuangan.
- Sistem sudah dapat digunakan oleh *user* untuk membantu dalam mengelola keuangan di kehidupan sehari-hari.

Saran yang dapat peneliti sampaikan untuk prospek pengembangan sistem yaitu:

- Mengembangkan fitur notifikasi secara berkala untuk mengingatkan *user* agar tidak lupa dalam melakukan pencatatan transaksi.
- Menu hutang dapat dikembangkan dengan menambahkan fitur catatan piutang kepada orang lain.

3. Aplikasi dapat dikembangkan dengan menambahkan model pengelolaan anggaran lain sehingga *user* dapat memilih sendiri model pengaturan anggaran yang paling cocok secara personal.
4. Aplikasi juga dapat dikembangkan dengan menghubungkan sistem manajemen ke pihak lain seperti *e-wallet* atau *e-banking* untuk melakukan transaksi.
5. Sistem Informasi yang telah diselesaikan dapat didesain kembali dengan tampilan yang lebih baik sesuai kebutuhan dan minat pengguna.

Daftar Pustaka

- [1] D. Liana, G. Rohmat, "Studi Komparasi Alat Bantu Pengelola Keuangan Pribadi Berbasis *Mobile*", Jurnal Teknologi dan Terapan Bisnis, No.1, Vol.2, pp. 34-41, 2019.
- [2] R. Silvia, "Aplikasi Pengelolaan Keuangan Pribadi Berbasis Desktop", Jurnal Ilmiah Technologia, No.2, Vol.12, pp. 68-71, 2021.
- [3] S. Safir, "Siapa Bilang Jadi Karyawan Nggak Bisa Kaya?", Alex Media Komputindo, Jakarta, 2017.
- [4] T., Harv, E., "Secret of the Millionaire Mind: Mastering the Inner Game of Wealth", Harper Business, New York, 2005,
- [5] M. P. Juli, A.I. Muhammad, dan S.S. Anggi, "Rancang Bangun Aplikasi Berbasis Android Dengan Penerapan Web Service Pada Sistem Informasi Perpustakaan (Studi Kasus: Perpustakaan Daerah Kalimantan)". Jurnal Sistem dan Teknologi Informasi, No.1, Vol.5, pp. 47-51, 2017.
- [6] A. Ichsanul, "Rancang Bangun Media Pembelajaran Bahasa Pemrograman Java Berbasis WEB Menggunakan Exe-Learning untuk Anak Sekolah Menengah Pertama", *Skripsi*, Pendidikan Teknologi Informasi, Universitas Islam Negeri Aar-Raniry, 2022.
- [7] A.L. Geraldo, S.N. Rizal, dan B.N. Xaverius, "Rancang Bangun Aplikasi Ensiklopedia Hukum Indonesia Berbasis Android. Jurnal Teknik Informatika", No.3, Vol.14, pp. 341-348, 2019.
- [8] N.A. Miza, H.D. Anisya, Y. Sarah, C. Octavia, dan J.M. Sauda, "Metode Penelitian Kualitatif Studi Pustaka", Edumaspul: Jurnal Pendidikan, No.1, Vol.6, pp. 974-980, 2022.
- [9] T. Anggy, "Implementasi *Extreme Programming (XP) Agile Development* pada Pengembangan Sistem Informasi Keluargaku", Jurnal Informatika Universitas Pamulang, No.2, Vol.6, pp. 305-311, 2021.
- [10] C.P. Anisya, S. Alfy, S. Fitriani, F. Umi, A.G. Qhoiril, dan L.H. Muhammad, "Pengujian Black Box dan White Box Sistem Informasi Parkir Berbasis WEB", Jurnal Testing dan Implementasi Sistem Informasi, No.1, Vol.1, 1-16, 2023.