

# RANCANG BANGUN APLIKASI PENDATAAN PENDUDUK BERBASIS ANDROID DENGAN PENERAPAN SISTEM *MOBILE* MENGGUNAKAN METODE *RAPID APPLICATION DEVELOPMENT* (RAD)

Fauzan Ambadar<sup>1)</sup>, Arief Hertadi Rustam<sup>2)</sup>

<sup>1), 2)</sup> Jurusan Teknik Informatika, STMIK “AMIKBANDUNG”, Jl. Jakarta No.28, Kebonwaru, Kota Bandung;  
Telp. 0895 1380 8029  
Email: fauzanambadar12@gmail.com

## Abstrak

Pendataan penduduk di tingkat wilayah masih banyak dilakukan secara konvensional melalui pencatatan manual atau spreadsheet, yang berpotensi menimbulkan keterlambatan pembaruan, kesalahan pencatatan, serta ketidaksinkronan data antarwilayah. Permasalahan tersebut berdampak pada rendahnya akurasi dan efisiensi pengelolaan data kependudukan. Penelitian ini bertujuan merancang dan membangun aplikasi pendataan penduduk berbasis Android untuk mempermudah pengurus RT dalam melakukan input dan pembaruan data, serta memungkinkan RW mengakses data seluruh RT secara real-time. Pengembangan aplikasi menggunakan metode Rapid Application Development (RAD) yang meliputi tahap analisis kebutuhan, perancangan, implementasi, dan evaluasi secara iteratif. Pengujian sistem dilakukan dengan Black Box Testing untuk menguji fungsionalitas, serta User Acceptance Test (UAT) untuk mengukur kepuasan pengguna. Hasil pengujian menunjukkan bahwa aplikasi berjalan sesuai kebutuhan pengguna dengan rata-rata nilai 86,6% pada aspek Functionality, Reliability, Usability, dan Efficiency, yang termasuk kategori “Sangat Setuju”. Aplikasi ini terbukti mampu meningkatkan efisiensi, akurasi, dan kecepatan administrasi kependudukan di tingkat RT dan RW, serta mendukung digitalisasi pelayanan publik secara terintegrasi dan real-time.

**Kata kunci:** Aplikasi Android, RAD, *Black Box Testing*, Pendataan Penduduk.

## Abstract

Population data collection at the neighborhood level is still largely conducted using conventional methods such as manual records or spreadsheets, which often lead to delayed updates, data entry errors, and inconsistencies between administrative units. These issues negatively affect the accuracy and efficiency of population data management. This study aims to design and develop an Android-based population data collection application to assist neighborhood administrators (RT) in inputting and updating data, while enabling community administrators (RW) to access integrated data from all RTs in real time. The application was developed using the Rapid Application Development (RAD) method, which consists of requirements analysis, system design, implementation, and iterative evaluation. System testing was carried out using Black Box Testing to verify functional performance and User Acceptance Testing (UAT) to measure user satisfaction. The test results indicate that the application meets user requirements, achieving an average score of 86.6% across the Functionality, Reliability, Usability, and Efficiency aspects, categorized as “Strongly Agree.” The application is proven to improve efficiency, accuracy, and speed in population administration at the RT and RW levels, support real-time data synchronization, and enhance the quality of public services through digital transformation.

**Keywords:** Android Application, RAD, *Black Box Testing*, Population Data.

## 1. PENDAHULUAN

### 1.1. Latar Belakang

Setiap lingkungan tempat tinggal, baik di daerah perkotaan maupun pedesaan, memiliki sistem administrasi kependudukan yang bertujuan untuk mengelola data warganya secara terstruktur. Sistem administrasi kependudukan ini umumnya terdiri dari dua tingkatan pengurus, yaitu Rukun Tetangga (RT) dan Rukun Warga (RW). RT merupakan unit terkecil dalam suatu lingkungan yang terdiri dari beberapa kepala keluarga yang dipimpin oleh satu orang ketua. Sementara itu,

RW merupakan gabungan dari beberapa RT dalam suatu wilayah dan dipimpin oleh satu ketua.

Namun, pada kenyataannya, proses pencatatan data penduduk di tingkat RT dan RW umumnya masih belum terkelola dengan digitalisasi, saat ini menggunakan *spreadsheet* atau bahkan di buku. Metode konvensional ini mencakup pencatatan data secara tertulis, pengumpulan formulir kertas dari warga, serta penginputan data secara manual ke dalam file *spreadsheet* tanpa sistem otomatisasi. Hal ini juga terjadi pada kepengurusan RW 01 Cikudapateuh Dalam yang masih mengandalkan

cara tradisional dalam mengelola dataarganya. Proses ini memerlukan waktu dan tenaga yang besar, terutama karena jumlah penduduk yang terus bertambah serta kebutuhan untuk memperbarui data secara berkala dengan akurasi tinggi.

Dalam pelaksanaannya, pendataan penduduk di tingkat RT dan RW diawali dari pengumpulan data oleh ketua RT yang meminta informasi langsung dari setiap warga di lingkungannya, baik melalui pendataan dari rumah ke rumah maupun permintaan pengisian formulir. Data tersebut kemudian diolah untuk menghasilkan berbagai kategori kependudukan seperti agama, jenis kelamin, status pernikahan, tingkat pendidikan, pekerjaan, usia, dan lain-lain. Selain itu, ketua RT juga bertanggung jawab untuk memperbarui data ketika terdapat perubahan, seperti adanya warga yang pindah tempat tinggal atau meninggal dunia. Setelah data tersebut diperbarui, hasil pendataan dari setiap RT kemudian dilaporkan kepada ketua RW untuk disesuaikan dan digabungkan dalam sistem administrasi kependudukan di tingkat RW. Namun, proses ini sering kali menimbulkan kendala berupa ketidaksesuaian data antara RT dan RW, terutama apabila pembaruan data tidak disampaikan dengan cepat dan akurat.

Untuk itu, penulis melakukan penelitian di RW 01 RT 07 yang berlokasi di Jalan Cikudapateuh Dalam, Kelurahan Kacapiring, Kecamatan Batununggal, dengan tujuan untuk mengetahui kebutuhan pengurus serta mengidentifikasi kendala dalam proses pendataan warga. Solusi untuk mengatasi permasalahan tersebut, dibutuhkan sebuah aplikasi berbasis android yang dapat memfasilitasi proses pendataan penduduk secara efisien dan terintegrasi. Aplikasi ini dirancang agar setiap ketua RT dapat melakukan input, pembaruan, dan pengelolaan data penduduk secara digital sesuai dengan wilayahnya. Selain itu, aplikasi ini memungkinkan ketua RW untuk mengakses seluruh data dari masing-masing RT secara *real-time*, sehingga meminimalkan risiko ketidaksesuaian data. Dengan sistem yang terstruktur dan mudah diakses, proses pendataan dapat dilakukan dengan lebih cepat, akurat, dan efisien, sekaligus mempermudah pengelolaan informasi kependudukan di RW 01 Cikudapateuh Dalam.

Berdasarkan latar belakang yang telah diuraikan di atas, penulis merencanakan untuk merancang aplikasi pendataan penduduk berbasis android yang dapat mengintegrasikan proses pengelolaan data antara RT dan RW. Aplikasi ini bertujuan untuk membantu pengurus RT dalam mengumpulkan, memperbarui, dan mengelola data penduduk secara digital, serta memudahkan pengurus RW dalam menyamakan dan mengakses data dari seluruh RT secara *real-time*. Dengan mengurangi ketergantungan pada pencatatan manual dan mengoptimalkan proses pembaruan informasi, aplikasi ini diharapkan dapat meningkatkan

efisiensi kerja, akurasi data, serta mempercepat pelayanan administrasi bagi masyarakat RW 01 Cikudapateuh

## 2. METODOLOGI PENELITIAN

### 2.1. Teknik Pengumpulan Data

Dalam proses pengumpulan data, penulis menggunakan beberapa cara atau metode untuk mendapatkan data yang relevan. Teknik pengumpulan data yang dilakukan oleh penulis adalah sebagai berikut:

#### a. Wawancara

Wawancara merupakan metode pengumpulan data yang dilakukan dengan cara tanya jawab dengan pengurus Rukun Warga (RW) mengenai sistem pendataan penduduk di RW 01 Cikudapateuh Dalam untuk mendapatkan data dukungan dan informasi yang berkaitan dengan sistem yang akan dibuat.

### 2.2. Metodologi *Rapid Application Development* (RAD)

*Rapid Application Development* (RAD) yang merupakan salah satu model dari *System Development Life Cycle* (SDLC). *Rapid Application Development* (RAD) merupakan model proses pengembangan perangkat lunak yang menekankan pada siklus pengembangan yang sangat singkat. RAD memiliki siklus yang fleksibel, setiap tahapannya dapat dilakukan secara berulang hingga menghasilkan aplikasi yang sesuai dengan kebutuhan pengguna. Beberapa tahapan umum dalam metode RAD diantaranya yaitu:

#### 1. *Requirement Planning* (Perencanaan Kebutuhan)

Pada tahap ini, dilakukan proses identifikasi kebutuhan dari para pengguna sistem yang terdiri dari Ketua RT, Ketua RW, dan Admin. Proses ini melibatkan kegiatan wawancara dan observasi terhadap Ketua RT dan Ketua RW di RW 01 Cikudapateuh Dalam untuk memahami permasalahan yang dihadapi dalam proses pendataan penduduk. Penulis juga membuat Spesifikasi Kebutuhan Pengguna atau biasa disebut *Request User* untuk mencatat kebutuhan pengguna dalam aplikasi pendataan penduduk.

#### 2. *System Design* (Desain Sistem)

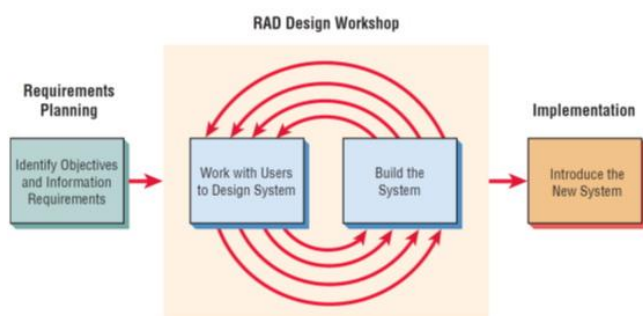
Setelah semua kebutuhan pengguna telah diperoleh dari tahap *Requirement Planning*, langkah selanjutnya adalah perancangan *System Design* atau perangkat lunak. Perancangan ini didasarkan pada analisis dan desain berorientasi objek, yaitu dengan menggunakan pemodelan UML (*Unified Modelling Language*). Proses perancangan dimulai dari menggambarkan alur kerja sistem dalam bentuk *use case diagram*. Kemudian menjelaskan struktur data seperti tabel apa saja, *field* apa saja dan relasi antar data dimodelkan dalam bentuk *class diagram* dan proses cara kerja fitur dalam bentuk *sequence diagram* dan perancangan *user interface*. Perancangan dilakukan secara bertahap dan iteratif, dimulai dari rancangan awal yang difokuskan pada fitur-fitur dasar seperti autentikasi dan *dashboard*.

### 3. Development (Pengembangan)

Tahap pengembangan dilakukan dengan mengimplementasikan seluruh desain yang telah dibuat pada tahap sebelumnya. Pengembangan dilakukan secara **iteratif** dan berulang. Setelah pengembangan fitur dasar pada iterasi pertama, dilakukan pengujian dan demonstrasi kepada pengguna lapangan. Masukan dari pengguna kemudian digunakan untuk perbaikan dan pengembangan pada fitur-fitur. Pada tahap awal, penulis menyiapkan Laravel 11 sebagai *framework backend* dan SQLite untuk pengelolaan basis data. Pengkodean dilakukan menggunakan Visual Studio Code. *Backend* Laravel dikembangkan dengan pendekatan REST API untuk menangani komunikasi antara aplikasi Android dan *server*, termasuk fitur autentikasi, manajemen data penduduk, dan *log* aktivitas pengguna. Di sisi *frontend*, aplikasi Android dibangun menggunakan *framework* Flutter dengan bahasa pemrograman Dart. Flutter terhubung ke *backend* Laravel melalui HTTP *request* dengan format data JSON. Fitur-fitur utama yang dikembangkan meliputi tambah data penduduk, *filter* data, pencarian, serta fitur lainnya sesuai kebutuhan pengguna. Setelah selesai, *testing* awal dilakukan oleh penulis untuk memastikan fungsionalitas dasar berjalan dengan baik.

### 4. Implementation (Implementasi)

Tahap terakhir ini mencakup penerapan aplikasi di lingkungan pengguna sesungguhnya. Penulis mendemonstrasikan hasil akhir aplikasi kepada Ketua RT dan Ketua RW beserta jajarannya. Selanjutnya, dilakukan evaluasi dan perbaikan berdasarkan masukan dari pengguna akhir, termasuk perbaikan *bug* dan penyesuaian fungsionalitas. Selain itu, dilakukan pelatihan kepada Ketua RT dan Ketua RW dalam menggunakan aplikasi ini agar mereka dapat mengoperasikan sistem dengan lancar. Setelah itu, aplikasi akan diperbaiki dan disesuaikan berdasarkan masukan dari pengguna, yang meliputi perbaikan *bug* dan penambahan fitur apabila diperlukan. Dengan aplikasi ini, diharapkan pengelolaan data penduduk menjadi lebih efektif dan efisien.



Gambar 1. Tahapan Model *Rapid Application Development*

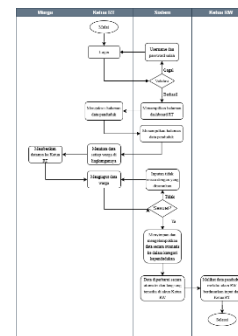
## 3. ANALISIS DAN PERANCANGAN

### 3.1. Analisis Sistem Sedang Berjalan (*Existing Flow*)

Sistem pendataan penduduk yang berjalan di lingkungan RW 01 Cikudapateuh Dalam saat ini masih dilakukan secara manual oleh para ketua RT dan RW.

Proses pendataan warga, seperti pencatatan nama, tempat dan tanggal lahir, alamat, agama, pendidikan terakhir, pekerjaan, dan status pernikahan, dilakukan di *spreadsheet* atau bahkan secara tertulis melalui buku catatan. Metode ini seringkali menyebabkan keterlambatan dalam pelaporan data, inkonsistensi informasi, dan kesulitan saat dilakukan rekapitulasi untuk keperluan administratif maupun laporan ke kelurahan.

Selain itu, proses koordinasi antara Ketua RT dan RW juga menghadapi tantangan, terutama dalam hal komunikasi data yang cepat dan akurat. Ketika data penduduk perlu diperbaharui atau dilaporkan, Ketua RT harus menyerahkan laporan fisik kepada Ketua RW, yang kemudian Ketua RW akan menyusunnya kembali secara manual. Hal ini menimbulkan risiko data ganda, kehilangan informasi, serta keterlambatan dalam pengambilan keputusan yang berbasis data warga. Oleh karena itu, diperlukan sebuah sistem digital yang mampu memfasilitasi pendataan secara *real-time* dan terpusat.



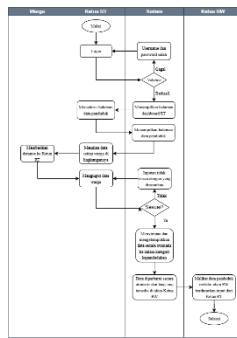
Gambar 2. Proses Pendataan Penduduk Ketua RT Ke Ketua RW Melalui Aplikasi Wargaku Di RW 01 Cikudapateuh Dalam

### 3.2. Analisis Sistem Diusulkan (*Proposed Flow*)

Untuk mengatasi permasalahan pendataan penduduk yang masih dilakukan secara manual, penulis mengusulkan sebuah sistem aplikasi pendataan penduduk berbasis android. Aplikasi ini dirancang khusus untuk membantu tugas RT dan RW dalam mendata, mengelola, dan melaporkan informasi kependudukan di wilayahnya. Melalui aplikasi ini, Ketua RT dapat langsung menginput data warga melalui perangkat seluler, dan data tersebut akan secara otomatis tersimpan dalam database pusat yang juga dapat diakses oleh Ketua RW. Sistem ini juga dilengkapi dengan fitur *filter* data berdasarkan jenis kelamin, usia, pendidikan, pekerjaan, agama, dan status pernikahan, sehingga mempermudah penyusunan laporan sesuai kebutuhan wilayah.

Dengan diterapkannya aplikasi pendataan penduduk ini, diharapkan proses pendataan warga di RW 01 Cikudapateuh Dalam dapat dilakukan dengan lebih cepat, akurat, dan efisien. Penggunaan sistem ini juga mampu mengurangi ketergantungan pada proses manual yang rawan kesalahan, serta meningkatkan transparansi dalam pelaporan dan pengambilan keputusan

di tingkat RT dan RW. Selain itu, aplikasi ini dapat menjadi langkah awal dalam mendukung transformasi digital pelayanan masyarakat di tingkat lingkungan.



Gambar 3. Proses Pendataan Penduduk Ketua RT Ke Ketua RW Melalui Aplikasi Wargaku Di RW 01 Cikudapateuh Dalam

### 3.3. Analisis Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berisi mengenai proses yang akan dilakukan oleh sistem. Berdasarkan informasi yang didapatkan maka kebutuhan fungsional yang dapat dilihat dari tabel dibawah ini:

Tabel 1. Kebutuhan Fungsional

| Nomor KF                | Deskripsi                            |
|-------------------------|--------------------------------------|
| <b>Aktor 1 Admin</b>    |                                      |
| KF-01                   | Menu <i>Login</i>                    |
| KF-02                   | Menu <i>Dashboard</i>                |
| KF-03                   | Menu Data <i>User</i>                |
| KF-04                   | Menu Data Penduduk                   |
| KF-05                   | <i>Filter</i> Wilayah Daerah, RW, RT |
| KF-06                   | Tambah Data Penduduk                 |
| KF-07                   | Cari Data Penduduk                   |
| KF-08                   | <i>Edit</i> Data Penduduk            |
| KF-09                   | Hapus Data Penduduk                  |
| KF-10                   | <i>Filter</i> Kriteria Data Penduduk |
| KF-11                   | Menu <i>Logout</i>                   |
| <b>Aktor 2 Ketua RW</b> |                                      |
| KF-12                   | Menu <i>Login</i>                    |
| KF-13                   | Menu <i>Register</i>                 |
| KF-14                   | Menu <i>Dashboard</i>                |
| KF-15                   | Menu Data Penduduk                   |
| KF-16                   | <i>Filter</i> Wilayah RT             |
| KF-17                   | Tambah Data Penduduk                 |
| KF-18                   | Cari Data Penduduk                   |
| KF-19                   | <i>Edit</i> Data Penduduk            |
| KF-20                   | Hapus Data Penduduk                  |
| KF-21                   | <i>Filter</i> Kriteria Data Penduduk |
| KF-22                   | Menu <i>Logout</i>                   |
| <b>Aktor 3 Ketua RT</b> |                                      |
| KF-23                   | Menu <i>Login</i>                    |
| KF-24                   | Menu <i>Register</i>                 |
| KF-25                   | Menu <i>Dashboard</i>                |
| KF-26                   | Menu Data Penduduk                   |
| KF-27                   | Tambah Data Penduduk                 |
| KF-28                   | Cari Data Penduduk                   |
| KF-29                   | <i>Edit</i> Data Penduduk            |
| KF-30                   | Hapus Data Penduduk                  |
| KF-31                   | <i>Filter</i> Kriteria Data Penduduk |
| KF-32                   | Menu <i>Logout</i>                   |

### 3.4. Analisis Kebutuhan Non Fungsional

Kebutuhan non-fungsional merupakan aspek-aspek yang menggambarkan kualitas dan performa dari suatu sistem,

yang tidak berhubungan langsung dengan fungsi utama yang dijalankan oleh sistem. Elemen ini memiliki peran penting karena dapat memengaruhi kenyamanan pengguna dalam menggunakan sistem serta memastikan sistem tetap stabil dan dapat digunakan dalam jangka waktu yang panjang. Adapun beberapa kebutuhan non-fungsional sebagai berikut:

Tabel 2. Kebutuhan Non Fungsional

| Nomor KNF         | Deskripsi                                      |
|-------------------|--|
| <b>Hardware</b>   |  |
| KNF-01            | LG 19M38A                                      |
| KNF-02            | Intel(R) Core(TM) i7-3770                      |
| KNF-03            | CPU @ 3.40GHz (8 CPUs)                         |
| KNF-04            | RAM 16384 MB                                   |
| KNF-05            | Keyboard dan Mouse                             |
| KNF-06            | Monitor 18.5                                   |
| <b>Software</b>   |  |
| KNF-07            | <i>Windows</i> 10                              |
| KNF-08            | <i>Laragon</i> v6.0                            |
| KNF-09            | <i>Visual Studio Code</i> v1.101.1             |
| KNF-10            | <i>Framework Flutter</i> v3.22.1               |
| KNF-11            | <i>Dart</i> v3.2.1                             |
| KNF-12            | <i>Framework Laravel</i> v12.1.1               |
| KNF-13            | <i>PHP</i> v8.3.13                             |
| KNF-14            | <i>DB Browser for SQLite</i> v3.13.1           |
| KNF-15            | <i>SQLite</i> v3.46.1                          |
| <b>Fungsional</b> |  |
| KNF-16            | Sistem aplikasi berbasis Android               |
| KNF-17            | Akses aplikasi menggunakan jaringan Internet   |
| KNF-18            | Sistem dijalankan menggunakan <i>handphone</i> |

### 3.5. Definisi Aktor

Aktor dalam sistem pendataan penduduk di RW 01 Cikudapateuh Dalam merupakan individu atau kelompok yang berperan langsung dalam penggunaan dan pengelolaan sistem. Dalam konteks aplikasi yang dirancang, aktor yang terlibat terdiri dari Admin, Ketua RW, dan Ketua RT yang ada di wilayah RW 01 Cikudapateuh Dalam. Ketiga aktor ini memiliki peran penting dalam mendukung kelancaran proses pendataan serta pengelolaan data warga secara digital.

Tabel 3. Analisis Pengguna

| No | Aktor | Keterangan  |
|----|-------|---|
| 1. | Admin | Admin adalah aktor yang memiliki peran untuk mengelola dan memantau seluruh aktivitas pengguna dalam aplikasi, termasuk pengguna RW dan RT. Admin memiliki hak akses penuh terhadap sistem, seperti melihat data penduduk dari seluruh wilayah, melihat data pengguna RW dan RT memantau <i>log</i> aktivitas <i>user</i> , serta mengakses diagram statistik seperti perbandingan jumlah pengguna RW dan RT serta wilayah dengan jumlah penduduk terbanyak. Dengan peran tersebut, admin berfungsi sebagai pengawas utama sistem yang memastikan semua data terorganisir dengan baik dan dapat digunakan untuk keperluan evaluasi atau pengambilan keputusan tingkat atas. |



## 2. Ketua RW

RW adalah aktor yang bertugas mengawasi dan mengelola data dari seluruh RT yang ada di bawah wilayahnya. Dalam aplikasi, RW bisa melihat data penduduk dari semua RT di wilayahnya, memantau aktivitas yang dilakukan oleh RT, dan melihat grafik jumlah penduduk RW per bulan serta data berdasarkan jenis kelamin, agama, pendidikan, usia dan status pernikahan. Dengan peran ini, RW membantu memastikan semua data dari RT tercatat dengan baik dan bisa dipantau oleh admin.

RT adalah aktor yang bertugas menginput dan mengelola data penduduk di wilayahnya secara langsung. Dalam aplikasi, RT dapat menambahkan, mengedit, dan menghapus data warga, serta melihat informasi data warga yang diinput oleh RT tersebut. RT juga bisa melihat grafik jumlah penduduk per bulan dan data berdasarkan kategori tertentu seperti jenis kelamin, agama, pendidikan, status pernikahan dan usia. Dengan peran ini, RT menjadi pengguna utama yang bertanggung jawab atas pendataan warga di lingkungan terkecil.

## 3. Ketua RT

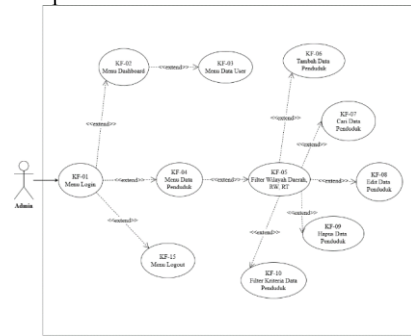
### 3.6. Use Case Diagram

*Use case diagram* merupakan salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan hubungan atau interaksi antara aktor (pengguna) dengan sistem berdasarkan fungsi-fungsi utama yang tersedia. Diagram ini memberikan gambaran umum mengenai perilaku sistem dari sudut pandang pengguna, sehingga memudahkan dalam memahami kebutuhan fungsional serta batasan sistem yang akan dibangun. Tujuan utama dari *use case diagram* adalah untuk mengidentifikasi siapa saja yang akan menggunakan sistem (aktor), serta layanan atau fitur apa saja yang dibutuhkan oleh masing-masing aktor. Diagram ini juga memvisualisasikan interaksi antara aktor dengan berbagai proses dalam sistem, serta menunjukkan hubungan antar fitur melalui dependensi seperti `<<include>>` dan `<<extend>>`. Dengan demikian, *use case diagram* berperan penting dalam merancang sistem karena membantu semua pihak memahami ruang lingkup dan alur kebutuhan fungsional secara menyeluruh.

#### 1. Use Case Diagram Admin

*Use case diagram* untuk aktor Admin digunakan untuk menggambarkan interaksi antara Admin dengan sistem secara keseluruhan. Diagram ini menunjukkan fungsi-fungsi utama yang dapat dilakukan oleh Admin, seperti mengelola data pengguna, memantau aktivitas RT dan RW, serta mengakses laporan pendataan warga. Melalui *use case diagram* ini, dapat terlihat bagaimana Admin menjadi pusat kontrol dalam sistem, dengan hak akses penuh

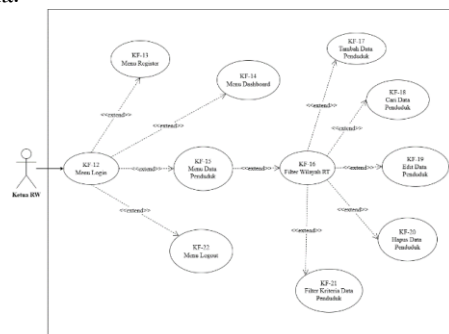
terhadap fitur-fitur penting yang mendukung kelancaran operasional aplikasi.



Gambar 4. Use case diagram Admin

#### 2. Use Case Diagram Ketua RW

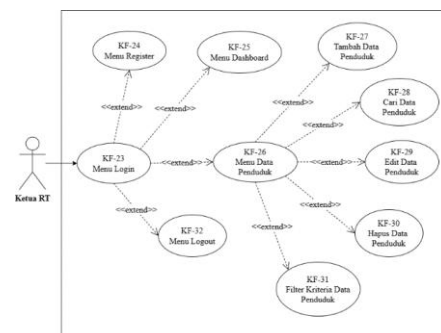
*Use case diagram* untuk aktor RW digunakan untuk menggambarkan interaksi RW dengan sistem dalam menjalankan tugasnya sebagai pengawas dan pengelola data dari seluruh RT di wilayahnya. Diagram ini menampilkan fitur-fitur yang dapat digunakan oleh RW seperti mengakses *dashboard*, melihat data penduduk dari semua RT, menambah data penduduk, mengedit dan menghapus data yang sudah ada, serta menggunakan fitur *filter* berdasarkan wilayah maupun kriteria tertentu seperti jenis kelamin, agama, pendidikan, status pernikahan, dan usia.



Gambar 5. Use case diagram Ketua RW

#### 3. Use Case Diagram Ketua RT

*Use case diagram* untuk aktor RT digunakan untuk menggambarkan interaksi RT dengan sistem dalam menjalankan tugasnya sebagai pencatat dan pengelola data warga di wilayah RT masing-masing. Diagram ini mencakup fitur-fitur seperti mengakses *dashboard*, menambahkan data penduduk, mengedit dan menghapus data, serta melihat data penduduk yang berada di lingkungannya sendiri.



Gambar 5. *Use case diagram* Ketua RT

### 3.7. Use Case Scenario

*Use case scenario* adalah deskripsi langkah-langkah atau alur yang menjelaskan secara singkat bagaimana suatu fungsi dalam sistem dijalankan oleh aktor dari awal hingga akhir. Skenario ini menggambarkan interaksi antara aktor dan sistem dalam suatu *use case*, baik pada kondisi normal (alur utama) maupun kondisi khusus (alur alternatif). Berikut *use case scenario* untuk Admin, Ketua RW dan Ketua RT.

Tabel 4. *Use case scenario* Menu *Login*

| Identifikasi  |  |
|---|--|
| Kode  | KF-01, KF-12 dan KF-23   |
| Nama  | Menu <i>Login</i>  |
| Aktor   | Admin, Ketua RW dan Ketua RT   |
| Pre-kondisi   | Aktor belum melakukan <i>login</i>   |
| Kondisi Akhir yang diharapkan                           | Aktor dapat masuk ke sistem  |
| Skenario utama  | Berhasil <i>login</i>  |
| Aksi Aktor  | Reaksi Sistem  |
| 1. Membuka sistem                                       | 2. Menampilkan <i>splash screen</i>  |
|   | 3. Menampilkan menu <i>login</i>   |
| 4. Mengisi <i>email</i> dan <i>password</i> yang sesuai |  |
| 5. Klik <i>button</i> “ <i>Login</i> ”                  | 6. Verifikasi <i>email</i> dan <i>password</i>                                 |
|   | 7. Menerima dan menyimpan token  |
|   | 8. Menampilkan menu <i>dashboard</i> berdasarkan hak akses <i>role</i>         |
| Skenario Alternatif                                     | Gagal <i>Login</i>   |
| 1. Membuka sistem                                       | 2. Menampilkan <i>splash screen</i>  |
|   | 3. Menampilkan Menu <i>Login</i>   |
| 4. Mengisi <i>email</i> dan <i>password</i> yang sesuai |  |
| 5. Klik <i>button</i> “ <i>Login</i> ”                  | 6. Verifikasi <i>email</i> dan <i>password</i>                                 |
|   | 7. Muncul <i>pop up</i> dengan pesan “ <i>email</i> dan <i>password</i> salah” |

Tabel 5. *Use case scenario* Menu *Filter Wilayah Data Penduduk*

| Identifikasi   |   |
|--|---|
| Kode   | KF-05 dan KF-16   |
| Nama   | <i>Filter Wilayah</i>   |
| Aktor  | Admin dan Ketua RW  |
| Pre-kondisi  | Aktor berada di menu data penduduk                                      |
| Kondisi Akhir yang diharapkan                                    | Aktor dapat melihat data penduduk sesuai wilayah yang dipilih pengguna  |
| Skenario utama   | Berhasil menampilkan data penduduk sesuai wilayah yang dipilih pengguna |
| Aksi Aktor   | Reaksi Sistem   |
| 1. Memilih <i>form</i> wilayah berdasarkan hak akses <i>role</i> | 2. Memproses permintaan <i>filter</i> wilayah                           |
|  | 3. Menampilkan data penduduk berdasarkan wilayah yang telah dipilih     |

#### Skenario Alternatif

- Data penduduk belum ada di wilayah yang dipilih
- Memilih *form* wilayah berdasarkan hak akses *role*
  - Memproses permintaan *filter* wilayah
  - Menampilkan menu data penduduk dengan pesan “Belum ada data penduduk”

Tabel 6. *Use case scenario* Menu *Tambah Data Penduduk*

| Identifikasi   |   |
|--|---|
| Kode   | KF-06, KF-17 dan KF-27  |
| Nama   | Tambah Data Penduduk  |
| Aktor  | Admin, Ketua RW dan Ketua RT  |
| Pre-kondisi  | Aktor berada di menu data penduduk  |
| Kondisi Akhir yang diharapkan  | Aktor dapat menambahkan data penduduk   |
| Skenario utama   | Berhasil menambahkan data penduduk  |
| Aksi Aktor   | Reaksi Sistem   |
| 1. Klik <i>button</i> “ <i>Tambah</i> ”  | 2. Muncul <i>pop up form</i> tambah data penduduk                                   |
| 3. Mengisi no nik, nama, jenis kelamin, tempat lahir, tanggal lahir, status, agama, pendidikan, pekerjaan, kependudukan di kelurahan, alamat dan tanggal mulai di RT |   |
| 4. Klik <i>button</i> “ <i>Simpan</i> ”  | 5. Validasi <i>input</i> data tambah penduduk                                       |
|  | 6. Menampilkan menu data penduduk dengan pesan “Data penduduk berhasil ditambahkan” |
| Skenario Alternatif  | Gagal menambahkan data penduduk   |
| 1. Klik <i>button</i> “ <i>Tambah</i> ”  | 2. Muncul <i>pop up form</i> tambah data penduduk                                   |
| 3. Mengisi no nik, nama, jenis kelamin, tempat lahir, tanggal lahir, status, agama, pendidikan, pekerjaan, kependudukan di kelurahan, alamat dan tanggal mulai di RT |   |
| 4. Klik <i>button</i> “ <i>Simpan</i> ”  | 5. <i>Input</i> data <i>form</i> tambah data penduduk tidak valid                   |
|  | 6. Muncul <i>alert</i> di <i>field</i> yang salah                                   |

Tabel 7. *Use case scenario* Menu *Cari Data Penduduk*

| Identifikasi |                                    |
|--------------|------------------------------------|
| Kode         | KF-07, KF-18 dan KF-28             |
| Nama         | Cari Data Penduduk                 |
| Aktor        | Admin, Ketua RW dan Ketua RT       |
| Pre-kondisi  | Aktor berada di menu data penduduk |

| Kondisi Akhir yang diharapkan                                     | Aktor dapat mencari data penduduk berdasarkan nama penduduk<br>Berhasil menampilkan data penduduk berdasarkan <i>input</i> nama penduduk |
|---|--|
| Skenario utama  |  |
| Aksi Aktor  | Reaksi Sistem  |
| 1. Menuliskan nama penduduk yang di cari di <i>form</i> pencarian | 2. Memproses permintaan pencarian data penduduk<br>3. Menampilkan data penduduk berdasarkan <i>input</i> nama penduduk                   |
| Skenario Alternatif   | Data penduduk yang dicari tidak ada  |
| 1. Menuliskan nama penduduk yang di cari di <i>form</i> pencarian | Memproses permintaan pencarian data penduduk<br>2. Menampilkan menu data penduduk dengan pesan "Data penduduk yang dicari tidak ada"     |

Tabel 8. *Use case scenario* Menu Edit Data Penduduk

| Identifikasi                                  |  |
|---|--|
| Kode  | KF-08, KF-19 dan KF-29   |
| Nama  | Edit Data Penduduk   |
| Aktor   | Admin, Ketua RW dan Ketua RT                                     |
| Pre-kondisi                                   | Aktor berada di menu data penduduk                               |
| Kondisi Akhir yang diharapkan                 | Aktor dapat mengedit data penduduk                               |
| Skenario utama                                | Berhasil mengedit data penduduk                                  |
| Aksi Aktor                                    | Reaksi Sistem  |
| 1. Klik <i>button</i> "edit" di data penduduk | 2. Muncul <i>pop up form</i> edit data penduduk                  |
| 3. Mengedit data penduduk yang mau diubah     |  |
| 4. Klik <i>button</i> "Update"                | 5. Menampilkan menu data penduduk dengan pesan "Berhasil update" |
| Skenario Alternatif                           | Gagal mengedit data penduduk                                     |
| 1. Klik <i>button</i> "edit" di data penduduk | 2. Muncul <i>pop up form</i> edit data penduduk                  |
| 3. Mengedit data penduduk yang mau diubah     |  |
| 4. Klik <i>button</i> "Update"                | 5. Gagal mengedit data penduduk                                  |

Tabel 9. *Use case scenario* Menu Hapus Data Penduduk

| Identifikasi |                        |
|--------------|------------------------|
| Kode         | KF-09, KF-20 dan KF-30 |
| Nama         | Hapus Data Penduduk    |

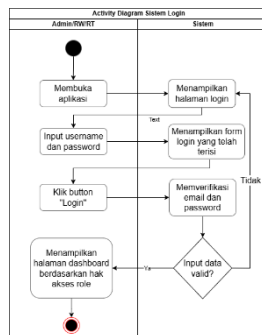
| Aktor  | Admin, Ketua RW dan Ketua RT  |
|--|---|
| Pre-kondisi                                    | Aktor berada di menu data penduduk  |
| Kondisi Akhir yang diharapkan                  | Aktor dapat menghapus data penduduk   |
| Skenario utama                                 | Berhasil menghapus data penduduk  |
| Aksi Aktor                                     | Reaksi Sistem   |
| 1. Klik <i>button</i> "hapus" di data penduduk | 2. Muncul <i>pop up</i> konfirmasi hapus data penduduk                          |
| 3. Klik <i>button</i> "hapus"                  | 4. Menampilkan menu data penduduk dengan pesan "Data penduduk berhasil dihapus" |
| Skenario Alternatif                            | Gagal menghapus data penduduk   |
| 1. Klik <i>button</i> "hapus" di data penduduk | 2. Muncul <i>pop up</i> konfirmasi hapus data penduduk                          |
| 3. Klik <i>button</i> "hapus"                  | 4. Gagal menghapus data penduduk  |

Tabel 9. *Use case scenario* Filter Kriteria Data Penduduk

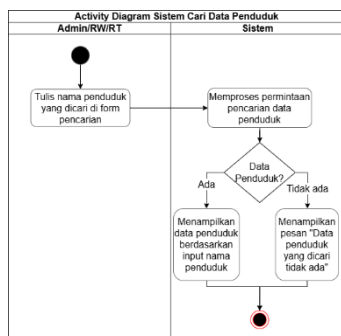
| Identifikasi   |  |
|--|--|
| Kode   | KF-10, KF-21 dan KF-31   |
| Nama   | Filter Kriteria Data Penduduk  |
| Aktor  | Admin, Ketua RW dan Ketua RT   |
| Pre-kondisi  | Aktor berada di menu data penduduk   |
| Kondisi Akhir yang diharapkan                          | Aktor dapat memfilter data penduduk  |
| Skenario utama   | Berhasil memfilter data penduduk   |
| Aksi Aktor   | Reaksi Sistem  |
| 1. Klik <i>button</i> "Filter"                         | 2. Muncul <i>pop up form</i> filter data penduduk                                    |
| 3. Pilih kriteria data penduduk yang ingin ditampilkan |  |
| 4. Klik <i>button</i> "Terapkan Filter"                | 5. Memproses permintaan <i>filter</i> data penduduk                                  |
| Skenario Alternatif                                    | 6. Menampilkan data penduduk berdasarkan <i>filter</i> data penduduk                 |
| 1. Klik <i>button</i> "Filter"                         | Data penduduk yang dicari tidak ada  |
| 3. Pilih kriteria data penduduk yang ingin ditampilkan | 2. Muncul <i>pop up form</i> filter data penduduk                                    |
| 4. Klik <i>button</i> "Terapkan Filter"                | 5. Memproses permintaan <i>filter</i> data penduduk                                  |
|  | 6. Menampilkan menu data penduduk dengan pesan "Data penduduk yang dicari tidak ada" |

### 3.8. Activity Diagram

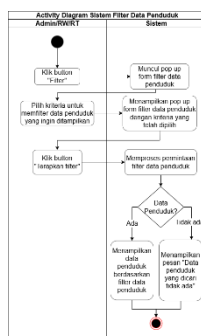
*Activity diagram* adalah salah satu jenis diagram dalam UML (*Unified Modeling Language*) yang digunakan untuk menggambarkan alur aktivitas atau proses dalam sebuah sistem secara rinci dan terstruktur. Diagram ini menunjukkan urutan aktivitas yang dilakukan sistem maupun pengguna saat menjalankan suatu fungsi, termasuk keputusan, percabangan, dan kondisi tertentu dari awal hingga akhir. *Activity diagram* membantu memvisualisasikan kebutuhan fungsional dalam sistem, sehingga memudahkan pemahaman terhadap logika dan alur kerja dari fitur atau proses dalam aplikasi. Berikut beberapa *Activity Diagram* pada aplikasi pendataan penduduk.



Gambar 6. Activity Diagram Login



Gambar 7. Activity Diagram Cari Data Penduduk

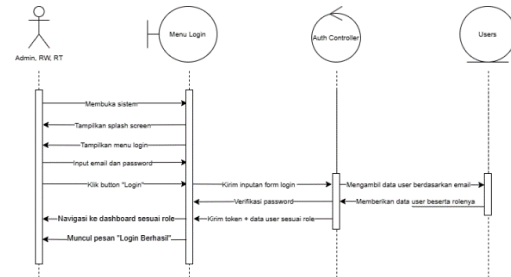


Gambar 8. Activity Diagram Filter Data Penduduk

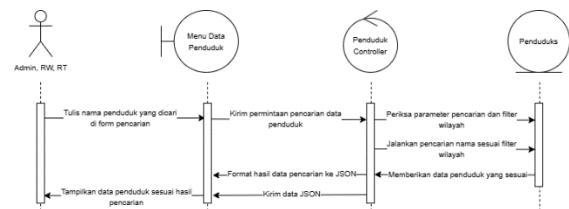
### 3.9. Sequence Diagram

*Sequence diagram* adalah salah satu jenis diagram dalam UML yang digunakan untuk menggambarkan urutan interaksi antara aktor dan sistem dalam menjalankan suatu proses. Diagram ini menunjukkan bagaimana pesan atau perintah dikirim dari satu objek ke objek

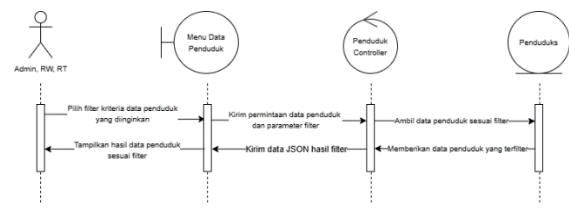
lainnya secara berurutan berdasarkan garis waktu. *Sequence diagram* digunakan untuk memvisualisasikan kebutuhan fungsional dan logika proses dalam sistem, sehingga membantu perancangan alur kerja yang efisien dan terstruktur. Dengan begitu, diagram ini bermanfaat untuk memahami komunikasi antar komponen secara detail dan memastikan setiap langkah proses berjalan sesuai fungsinya. Berikut beberapa *sequence diagram* pada aplikasi pendataan penduduk.



Gambar 9. Sequence Diagram Login



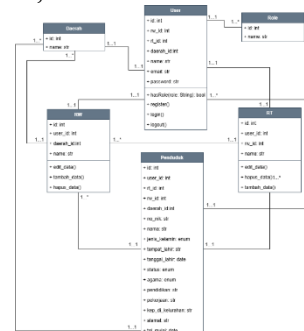
Gambar 10. Sequence Diagram Cari Data Penduduk



Gambar 11. Sequence Diagram Filter Kriteria Data Penduduk

### 3.10. Class Diagram

*Class Diagram* adalah jenis diagram dalam UML yang digunakan untuk menggambarkan struktur kelas dalam sistem, termasuk atribut, fungsi (*method*), dan relasi antar kelas. Diagram ini membantu menggambarkan kebutuhan fungsional sistem secara terstruktur, serta menunjukkan bagaimana setiap bagian saling terhubung dan saling berinteraksi. Berikut class diagram untuk Admin, RW, dan RT.



Gambar 11. Class Diagram Aplikasi Pendataan Penduduk



### 3.11. Perancangan Database

Perancangan *database* dilakukan untuk menyusun dan mengatur data agar dapat diakses, dikelola, dan disimpan secara efisien dalam aplikasi android pendataan penduduk. *Database* ini dirancang untuk mendukung kebutuhan fungsional aplikasi, seperti menyimpan data pengguna, data wilayah RT/RW, serta data kependudukan secara terstruktur. Dengan perancangan yang tepat, sistem dapat berjalan lebih optimal, akurat, dan memudahkan proses pencarian maupun pengelolaan data oleh masing-masing peran pengguna. Berikut beberapa tabel pada *database* dalam pembuatan aplikasi pendataan penduduk

Tabel 10. Perancangan Database Tabel Users

| No  | Field                    | Type            | Size | Keterangan                         |
|-----|--------------------------|-----------------|------|------------------------------------|
| 1.  | <i>Id*</i>               | <i>int</i>      | -    | <i>Primary key, auto increment</i> |
| 2.  | <i>rw_id</i>             | <i>int</i>      | -    | <i>Foreign key ke rw_id</i>        |
| 3.  | <i>rt_id</i>             | <i>int</i>      | -    | <i>Foreign key ke rt_id</i>        |
| 4.  | <i>daerah_id</i>         | <i>int</i>      | -    | <i>Foreign key ke daerah_id</i>    |
| 5.  | <i>name</i>              | <i>varchar</i>  | 255  | Nama pengguna                      |
| 6.  | <i>email</i>             | <i>varchar</i>  | 255  | Email unik pengguna                |
| 7.  | <i>email_verified_at</i> | <i>datetime</i> | -    | Waktu verifikasi email             |
| 8.  | <i>password</i>          | <i>varchar</i>  | 255  | Kata sandi terenkripsi             |
| 9.  | <i>remember_token</i>    | <i>varchar</i>  | 255  | Token sesi login                   |
| 10. | <i>created_at</i>        | <i>datetime</i> | -    | Waktu data dibuat                  |
| 11. | <i>updated_at</i>        | <i>datetime</i> | -    | Waktu data diperbarui              |

Tabel 11. Perancangan Database Tabel Penduduks

| No  | Field                   | Type           | Size | Keterangan                         |
|-----|-------------------------|----------------|------|------------------------------------|
| 1.  | <i>Id*</i>              | <i>int</i>     | -    | <i>Primary key, auto increment</i> |
| 2.  | <i>user_id</i>          | <i>int</i>     | -    | <i>Foreign key ke user_id</i>      |
| 3.  | <i>rw_id</i>            | <i>int</i>     | -    | <i>Foreign key ke rw_id</i>        |
| 4.  | <i>rt_id</i>            | <i>int</i>     | -    | <i>Foreign key ke rt_id</i>        |
| 5.  | <i>daerah_id</i>        | <i>int</i>     | -    | <i>Foreign key ke daerah_id</i>    |
| 6.  | <i>no_nik</i>           | <i>varchar</i> | 255  | NIK penduduk                       |
| 7.  | <i>nama</i>             | <i>varchar</i> | 255  | Nama penduduk                      |
| 8.  | <i>jenis_kelamin</i>    | <i>varchar</i> | 255  | Jenis kelamin penduduk             |
| 9.  | <i>tempat_lahir</i>     | <i>varchar</i> | 255  | Tempat lahir penduduk              |
| 10. | <i>tanggal_lahir</i>    | <i>date</i>    | -    | Tanggal lahir penduduk             |
| 11. | <i>status</i>           | <i>varchar</i> | 255  | Status pernikahan penduduk         |
| 12. | <i>agama</i>            | <i>varchar</i> | 255  | Agama penduduk                     |
| 13. | <i>pendidikan</i>       | <i>varchar</i> | 255  | Pendidikan terakhir penduduk       |
| 14. | <i>pekerjaan</i>        | <i>varchar</i> | 255  | Pekerjaan penduduk                 |
| 15. | <i>kep_di_kelurahan</i> | <i>varchar</i> | 255  | Kependudukan di kelurahan penduduk |

|     |                   |                 |     |                              |
|-----|-------------------|-----------------|-----|------------------------------|
| 16. | <i>Alamat</i>     | <i>varchar</i>  | 255 | Alamat rumah penduduk        |
| 17. | <i>tgl_mulai</i>  | <i>date</i>     | -   | Tanggal penduduk mulai di RT |
| 18. | <i>created_at</i> | <i>datetime</i> | -   | Waktu data dibuat            |
| 19. | <i>updated_at</i> | <i>datetime</i> | -   | Waktu data diperbarui        |

### 3.12. Perancangan User Interface

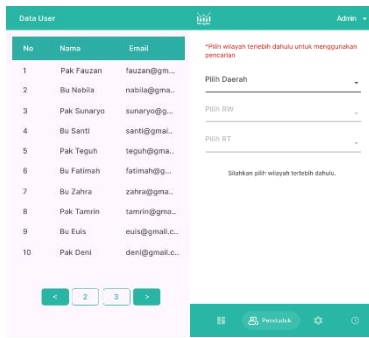
Perancangan *user interface* (antarmuka pengguna) bertujuan untuk menciptakan tampilan aplikasi yang mudah digunakan, jelas, dan sesuai dengan kebutuhan pengguna. Antarmuka ini menjadi jembatan antara pengguna dengan sistem, di mana setiap elemen visual dirancang untuk mendukung pelaksanaan kebutuhan fungsional seperti mengakses, mengelola, dan menampilkan data penduduk. Desain yang baik akan memudahkan Admin, RW, dan RT dalam memahami informasi, melakukan navigasi, serta menyelesaikan tugas dengan cepat dan efisien. Dengan begitu, perancangan *user interface* tidak hanya meningkatkan kejelasan informasi, tetapi juga memberikan pengalaman pengguna yang nyaman dan mendukung efektivitas dalam pengelolaan data kependudukan.



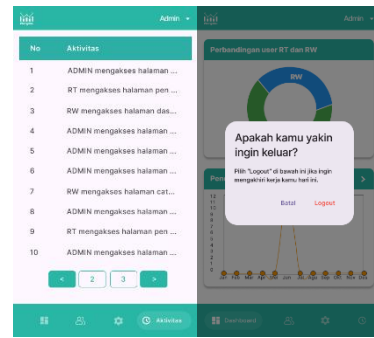
Gambar 12. Perancangan Interface Splash Screen dan Menu Login



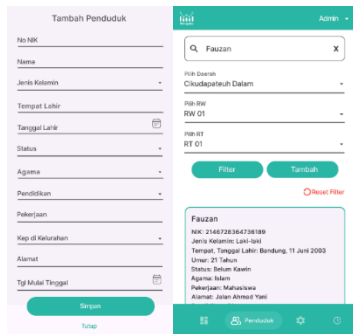
Gambar 13. Perancangan Interface Menu Register dan Menu Dashboard



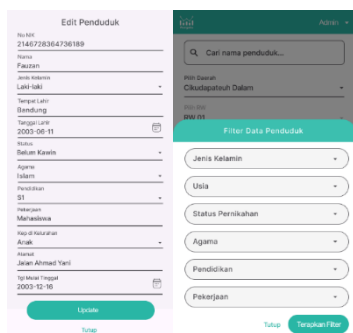
Gambar 14. Perancangan *Interface* Menu Data User dan Menu Data Penduduk



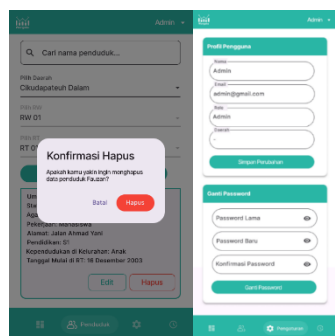
Gambar 18. Perancangan *Interface* Menu Catatan Aktivitas dan Menu Logout



Gambar 15. Perancangan *Interface* Tambah Data Penduduk dan Cari Data Penduduk



Gambar 16. Perancangan *Interface* Edit Data Penduduk dan Filter Kriteria Data Penduduk



Gambar 17. Perancangan *Interface* Hapus Data Penduduk dan Menu Pengaturan

### 3.13. Algoritma *Sequential Search*

*Sequential Search* adalah algoritma pencarian yang bekerja dengan cara membandingkan elemen satu per satu secara berurutan dari awal hingga akhir, hingga ditemukan data yang sesuai dengan *input* pencarian atau seluruh data telah diperiksa.

Penulis mengetahui bahwa terdapat algoritma pencarian lain yang lebih efisien, yaitu *Binary Search*. Namun, penggunaan *Sequential Search* dinilai lebih tepat untuk aplikasi ini. *Sequential Search* bekerja dengan cara memeriksa data satu per satu secara berurutan, dapat digunakan pada data yang tidak terurut dan cukup cepat untuk data dalam jumlah kecil. Sedangkan *Binary Search* bekerja dengan cara membagi data menjadi dua bagian di setiap langkah pencarian, wajib digunakan pada data yang sudah terurut dan sangat cepat, terutama untuk data dalam jumlah besar.

Berdasarkan perbandingan yang telah disebutkan, penulis memilih menggunakan algoritma *Sequential Search* untuk proses pencarian data penduduk berdasarkan nama warga dalam satu wilayah RT. Karena pencarian hanya dilakukan setelah pengguna memilih wilayah RT, RW, dan daerah tertentu, maka jumlah data yang dicari relatif kecil. Selain itu, untuk menggunakan *Binary Search*, data harus diurutkan terlebih dahulu, yang akan menambah beban komputasi yang tidak perlu. Oleh karena itu, *Sequential Search* dinilai lebih efisien, sederhana, dan cukup optimal untuk kebutuhan aplikasi ini.

```
// Search hanya dijalankan jika semua wilayah di pilih
if (
  $this->request->filled(key: 'search') &&
  $this->request->filled(key: 'daerah_id') &&
  $this->request->filled(key: 'rw_id') &&
  $this->request->filled(key: 'rt_id')
) {
  $query->where(column: 'nama', operator: 'like', value: '%' . $this->request->search . '%');
}
```

Gambar 19. Penerapan Algoritma *Sequential Search* Aplikasi Pendataan Penduduk

## 4. IMPLEMENTASI DAN PENGUJIAN

### 4.1. Implementasi *Rapid Application Development* (RAD)

Pada tahap ini, proses pembangunan sistem dilakukan menggunakan metode *Rapid Application Development* (RAD), yang menekankan pada pembangunan aplikasi secara cepat dan fleksibel berdasarkan kebutuhan

pengguna. Implementasi dimulai dari tahap perencanaan kebutuhan, yang dilakukan melalui observasi dan wawancara bersama Ketua RT dan RW di RW 01 Cikudapateuh Dalam. Setelah kebutuhan pengguna dihimpun, penulis menyusun perancangan sistem terlebih dahulu menggunakan *Unified Modeling Language* (UML) dan desain antarmuka pengguna (*user interface*) untuk menggambarkan alur kerja dan tampilan aplikasi. Selanjutnya, sistem dikembangkan menggunakan Laravel sebagai *backend* dan Flutter untuk *frontend* Android. Tahapan ini mencakup proses iteratif mulai dari perencanaan, perancangan, pembuatan prototipe awal, hingga finalisasi sistem yang siap diuji dan digunakan oleh pengguna.

## 4.2. Black Box Testing

*Black Box Testing* digunakan karena dalam *Black Box Testing* tidak memerlukan sumber daya atau tim berlatar belakang teknis, yang terpenting adalah penguji mampu memahami dan melihat dari perspektif pengguna. Metode ini dipilih karena sesuai dengan tujuan pengujian, yaitu memastikan bahwa *input* yang diberikan oleh pengguna menghasilkan *output* yang benar, serta memastikan sistem memberikan respons yang sesuai terhadap setiap tindakan pengguna. Dengan pendekatan ini, pengujian dilakukan dari sudut pandang pengguna, sehingga lebih menekankan pada apakah fungsi-fungsi aplikasi sudah bekerja dengan baik dan mudah digunakan oleh Admin, Ketua RW, dan Ketua RT. Berikut adalah *testcase* pengujian menggunakan *blackbox*:

Tabel 11. Black Box Testing

| No | ID Kebutuhan Fungsional | Test Case                    | Skenario Pengujian  | Hasil yang Di-harapkan   | Status Validasi |
|----|-------------------------|------------------------------|---|--|-----------------|
| 1  | KF-01, KF-12, KF-23     | Menu Login                   | Pengguna mengisi <i>form login</i> dengan <i>email</i> dan <i>password</i> .                      | Jika data valid, pengguna berhasil masuk sesuai <i>role</i> -nya. Jika tidak, muncul pesan <i>error</i> .  | Valid           |
| 2  | KF-13, KF-24            | Menu Register                | <i>Form register</i> diisi dengan data yang sesuai ketentuan <i>form</i> yang diminta.            | Jika data diisi dengan benar, pengguna diarahkan ke halaman <i>login</i> dengan pesan registrasi berhasil. Jika ada kesalahan, ditampilkan pesan <i>error</i> pada <i>field</i> yang sesuai. | Valid           |
| 3  | KF-02, KF-14, KF-25     | Menu Dashboard               | Pengguna berhasil <i>login</i> ke dalam sistem.   | Sistem menampilkan <i>dashboard</i> sesuai   | Valid           |
| 4  | KF-03                   | Menu Data User               | Admin mengakses Pendaftaran User per Periode 2025 di <i>dashboard</i> .                           |  | Valid           |
| 5  | KF-04, KF-15, KF-26     | Menu Data Penduduk           | Pengguna mengakses halaman data penduduk  |  | Valid           |
| 6  | KF-05, KF-16,           | Filter Wilayah Data Penduduk | Pengguna memilih wilayah pada <i>form filter</i> .  |  | Valid           |
| 7  | KF-06, KF-17, KF-27     | Tambah Data Penduduk         | Pengguna mengisi <i>form</i> tambah data penduduk dan menekan tombol simpan.                      |  | Valid           |
| 8  | KF-07, KF-18, KF-28     | Cari Data Penduduk           | Pengguna memasukkan nama penduduk yang dicari pada kolom pencarian data penduduk.                 |  | Valid           |
| 9  | KF-08, KF-19, KF-29     | Edit Data Penduduk           | Pengguna membuka <i>form edit</i> dan mengubah data penduduk, lalu menekan tombol <i>update</i> . |  | Valid           |

|   |                     |                               |   |   |       |
|---|---------------------|-------------------------------|---|---|-------|
| 1 | KF-09, KF-20, KF-30 | Hapus Data Penduduk           | Pengguna menekan tombol hapus pada salah satu data penduduk.  | Jika ada kesalahan, ditampilkan pesan <i>error</i> pada <i>field</i> yang bermasalah. Data penduduk terhapus dari daftar dan tidak lagi ditampilkan di halaman. | Valid |
| 1 | KF-10, KF-21, KF-31 | Filter Kriteria Data Penduduk | Pengguna memilih kriteria seperti usia, jenis kelamin, agama, pendidikan, pekerjaan, atau status pernikahan pada <i>form filter</i> . | Sistem menampilkan data penduduk sesuai dengan kriteria yang dipilih. Jika tidak ada data yang cocok, ditampilkan pesan "Data penduduk yang dicari tidak ada".  | Valid |
| 1 | KF-11, KF-22, KF-32 | Menu Logout                   | Pengguna menekan tombol <i>logout</i> di dalam sistem.  | Pengguna keluar dari sistem dan diarahkan ke halaman <i>login</i> .   | Valid |

### 4.3. User Acceptance Testing

*User Acceptance Testing* (UAT) dilakukan untuk mengevaluasi kualitas perangkat lunak dan memastikan bahwa sistem telah memenuhi kebutuhan pengguna akhir. Pengujian ini mengacu pada dimensi ISO 9126 dan menggunakan skala Likert sebagai dasar penilaian yang bersifat kuantitatif dan terstruktur. Proses pengujian dilakukan pada tahap akhir setelah sistem selesai dikembangkan, sebagai langkah verifikasi sebelum implementasi. Evaluasi dilakukan berdasarkan enam aspek utama dari ISO 9126, yaitu *functionality*, *reability*, *usability*, dan *efficiency*. Berikut adalah daftar pernyataan untuk *user testing*:

Tabel 12. UAT Aspek *Functionality* (Fungsionalitas)

| No | Pertanyaan  |
|----|---|
| 1. | Aplikasi Wargaku menyediakan fitur yang sesuai dengan kebutuhan saya sebagai pengguna.              |
| 2. | Semua menu dan fungsi dalam aplikasi dapat diakses dan berjalan sesuai dengan tujuan penggunaannya. |

Tabel 13. UAT Aspek *Reability* (Reabilitas)

| No | Pertanyaan   |
|----|--|
| 1. | Aplikasi Wargaku jarang mengalami gangguan atau <i>error</i> saat digunakan.                         |
| 2. | Data yang saya input dan simpan di aplikasi selalu tersimpan dengan baik tanpa kehilangan informasi. |

Tabel 14. UAT Aspek *Usability* (Kebergunaan)

| No | Pertanyaan   |
|----|--|
| 1. | Antarmuka aplikasi Wargaku mudah dipahami dan digunakan, bahkan oleh pengguna baru.              |
| 2. | Instruksi dan ikon dalam aplikasi sudah jelas dan memudahkan saya dalam mengoperasikan aplikasi. |

Tabel 15. UAT Aspek *Efficiency* (Efisiensi)

| No | Pertanyaan  |
|----|---|
| 1. | Proses input, pencarian, dan pengolahan data dalam aplikasi berlangsung dengan cepat dan efisien. |
| 2. | Aplikasi tidak membutuhkan waktu lama untuk memuat halaman atau berpindah antar menu.             |

Berikut adalah hasil perhitungan uji kualitas berdasarkan keempat aspek kualitas perangkat lunak.

Tabel 16. Tingkat Kualitas Perangkat Lunak

| Aspek                | Skor Aktual                      | Skor Ideal  | %Skor Aktual | Kriteria             |
|----------------------|----------------------------------|-------------|--------------|----------------------|
| <i>Functionality</i> | 568                              | 650         | 87,4%        | Sangat Setuju        |
| <i>Reability</i>     | 555                              | 650         | 85,4%        | Sangat Setuju        |
| <i>Usability</i>     | 565                              | 650         | 86,9%        | Sangat Setuju        |
| <i>Efficiency</i>    | 565                              | 650         | 86,9%        | Sangat Setuju        |
| <b>Total</b>         | <b>2253</b>                      | <b>2600</b> |              |                      |
| <b>Perhitungan</b>   | $\frac{2253}{2600} \times 100\%$ |             | <b>86,6%</b> | <b>Sangat Setuju</b> |

Dari perhitungan Tabel 16, berdasarkan hasil pengujian kualitas perangkat lunak pada aspek-aspek yang diuji, dapat disimpulkan sebagai berikut:

- Functionality*, *Reability*, *Usability* dan *Efficiency* berada dalam kategori Sangat Setuju dengan persentase di atas 84%. Hal ini menunjukkan bahwa aplikasi yang dibangun telah memenuhi standar kualitas yang sangat baik pada setiap aspek pengujian.
- Secara keseluruhan, aplikasi memperoleh rata-rata persentase sebesar 86,6%, yang termasuk dalam kategori Sangat Setuju. Dengan demikian, aplikasi dinilai telah layak digunakan dan mampu memenuhi kebutuhan pengguna. Meskipun begitu, pengembangan dan evaluasi berkelanjutan tetap dibutuhkan untuk mempertahankan serta meningkatkan kualitas aplikasi di masa mendatang.

## 5. HASIL DAN PEMBAHASAN

### 5.1 Hasil Implementasi Sistem

Implementasi sistem dilakukan menggunakan metode Rapid Application Development (RAD) yang memungkinkan proses pembangunan aplikasi dilakukan secara cepat, bertahap, dan fleksibel. Berdasarkan hasil observasi dan wawancara bersama Ketua RT dan RW di RW 01 Cikudapateuh Dalam, kebutuhan sistem berhasil dihimpun secara menyeluruh, sehingga pengembangan dapat dilakukan secara tepat sasaran.

Perancangan sistem menggunakan Unified Modeling Language (UML) serta desain antarmuka pengguna yang berfungsi untuk memvisualisasikan alur kerja aplikasi. Sistem kemudian dikembangkan menggunakan Laravel sebagai backend dan Flutter untuk frontend Android, sehingga aplikasi dapat berjalan dengan baik serta mudah digunakan oleh perangkat RW, RT, maupun admin.

Selama proses implementasi, model RAD memberikan keuntungan berupa feedback cepat dari pengguna setiap kali prototipe diuji. Perubahan dapat dilakukan secara iteratif, sehingga pada tahap finalisasi aplikasi telah sesuai dengan kebutuhan operasional di lingkungan RW 01. Hasil implementasi menunjukkan bahwa seluruh fitur utama, yakni manajemen penduduk, pendaftaran user, pengelolaan wilayah, filter data, hingga autentikasi pengguna, berhasil dijalankan dengan baik sesuai rancangan.

### 5.2 Hasil Pengujian Black Box Testing

Pengujian menggunakan metode Black Box Testing dilakukan untuk memastikan fungsi aplikasi bekerja sesuai dengan yang diharapkan dari sudut pandang pengguna tanpa melihat kode program yang digunakan. Berdasarkan 12 skenario pengujian yang dilakukan, seluruh fungsi yang diuji menunjukkan hasil "Valid", yang berarti sistem telah berjalan dengan benar. Adapun aspek-aspek yang diuji meliputi: Proses login dan registrasi Akses dashboard sesuai role (Admin, Ketua RT, Ketua RW) Manajemen data penduduk (tambah, edit, cari, hapus) Filter data berdasarkan wilayah maupun kategori penduduk Proses logout Hasil ini menunjukkan bahwa aplikasi mampu memberikan output yang sesuai terhadap input pengguna, menampilkan pesan kesalahan yang relevan ketika data tidak valid, serta menjaga konsistensi alur penggunaan. Dengan demikian, pada tahap fungsional dasar, sistem telah terbukti berjalan stabil dan memenuhi kebutuhan pengguna operasional.

### 5.3 Hasil User Acceptance Testing (UAT)

Pengujian User Acceptance Testing dilakukan kepada pengguna akhir dengan mengacu pada empat aspek kualitas perangkat lunak berdasarkan ISO 9126, yaitu Functionality, Reliability, Usability, dan Efficiency. Penilaian dilakukan menggunakan skala Likert untuk memperoleh gambaran objektif terkait penerimaan pengguna terhadap sistem.

Hasil perhitungan menunjukkan:

| Aspek         | Persentase | Kriteria      |
|---------------|------------|---------------|
| Functionality | 87,4%      | Sangat Setuju |
| Reliability   | 85,4%      | Sangat Setuju |
| Usability     | 86,9%      | Sangat Setuju |

|                 |       |               |
|-----------------|-------|---------------|
| Efficiency      | 86,9% | Sangat Setuju |
| Total Rata-rata | 86,6% | Sangat Setuju |

## 6. KESIMPULAN

1. Penelitian ini menghasilkan aplikasi pendataan penduduk berbasis Android yang dibangun menggunakan metode Rapid Application Development (RAD) dengan frontend Flutter dan backend Laravel yang terintegrasi melalui REST API. Aplikasi ini berhasil menggantikan proses pendataan manual yang sebelumnya mengandalkan spreadsheet, sehingga meningkatkan kecepatan, ketepatan, dan konsistensi data pada tingkat RT dan RW. Pengujian sistem melalui Black Box Testing menunjukkan bahwa seluruh fungsi berjalan sesuai kebutuhan, sedangkan User Acceptance Testing memberikan nilai rata-rata 86,6% yang termasuk kategori "Sangat Setuju". Hasil tersebut menunjukkan bahwa aplikasi mampu memenuhi kebutuhan pengguna dalam hal fungsionalitas, keandalan, kemudahan penggunaan, dan efisiensi. Dengan adanya sistem digital ini, proses pendataan dan sinkronisasi informasi antar pengurus dapat dilakukan secara real-time, sehingga mendukung pengambilan keputusan yang lebih cepat dan akurat serta meningkatkan kualitas layanan administrasi kependudukan. Penelitian ini juga menunjukkan bahwa pendekatan RAD efektif untuk pengembangan aplikasi layanan publik yang membutuhkan respons cepat terhadap perubahan kebutuhan pengguna.

## 7. DAFTAR PUSTAKA

- Deni Murdiani, Heri Hermawan, June 2022. Perbandingan Metode *Waterfall* Dan RAD (*Rapid Application Development*) Pada Pengembangan Sistem Informasi. *Jurnal Teknologi Informasi*, 6, (1).[Online]. Available: <https://core.ac.uk/download/pdf/524936989.pdf>.
- Tata Sutabri, April 2012. Analisis Sistem Informasi, Jakarta: CV ANDI OFFSET, [Online]. Available: <https://books.google.co.id/books?id=uI5eDwAAQBAJ&printsec=frontcover&hl=id#v=onepage&q&f=false>.
- Erwan Effendy, Elsa Adelia Siregar, Putri Chairina Fitri, Ibnu Alif Syahbana Damanik, 2023. Mengenal Sistem Informasi Manajemen Dakwah (Pengertian Sistem, Karakteristik Sistem), *Jurnal Pendidikan dan Konseling*, 35(2). [Online]. Available: <https://journal.universitaspahlawan.ac.id/index.php/jpdk/article/view/14061/10847>.
- Fitriantini Suaib, Hadrayanti, January 2018. Perancangan Aplikasi Sistem Helpdesk Berbasis Website Pada Direktorat Teknologi Informasi Universitas Hasanuddin Makassar, [Online].



Available: <https://digilibamin.unismuh.ac.id/upload/798-FullText.pdf>.

- Hasan Abdurahman, Asep Ririh Riswaya, December 2014. Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha Bhakti, *Jurnal Computech & Bisnis*, 8(2), [Online]. Available: <https://www.academia.edu/download/100315988/291480199.pdf>.
- Afista Galih Pradana, Sekreningsih Nita, 2019. Rancang Bangun Game Edukasi "AMUDRA" Alat Musik Daerah Berbasis Android, *Seminar Nasional Teknologi Informasi dan Komunikasi (SENATIK)*, 2(1), [Online]. Available: <https://prosiding.unipma.ac.id/index.php/SENATIK/article/view/1062/925>.
- Richy R. Saragih, 24 December 2018. Pemrograman Dan Bahasa Pemrograman, [Online]. Available: [https://www.researchgate.net/profile/RichySaragih/publication/329885312\\_PEMROGRAMAN\\_DAN\\_BAHASA\\_PEMROGRAMAN/link/s/5c209488458515a4c7f4b123/PEMROGRAMAN-DAN-BAHASA-PEMROGRAMAN.pdf](https://www.researchgate.net/profile/RichySaragih/publication/329885312_PEMROGRAMAN_DAN_BAHASA_PEMROGRAMAN/link/s/5c209488458515a4c7f4b123/PEMROGRAMAN-DAN-BAHASA-PEMROGRAMAN.pdf).
- Nelly Sofi, Riza Dharmawan, June 2022. Perancangan Aplikasi Bengkel Csm Berbasis Android Menggunakan Framework Flutter (Bahasa Dart), *Jurnal Teknik Dan Science*, 1(2), [Online]. Available: <https://journal.admi.or.id/index.php/JTS/article/download/125/137>.
- Abdul Mubarak, April 2019. Rancang Bangun Aplikasi Web Sekolah Menggunakan UML (Unified Modeling Language) Dan Bahasa Pemrograman PHP (PHP Hypertext Preprocessor) Berorientasi Objek, *JIKO (Jurnal Informatika dan Komputer) Ternate*, 02(1), [Online]. Available: <https://ejournal.unkhair.ac.id/index.php/jiko/article/viewFile/1052/1065>.
- Andry Andaru, 25 December 2018. Pengertian Database Secara Umum, *Fakultas Komputer*, [Online]. Available: <https://osf.io/preprints/osf/43b5jv1>.
- Ronal Micha Mansawan, Dani Sasmoko, Aristo Ari Kuncoro. Desember 2024. Aplikasi Penjualan Mieayam Berbasis Android Pada Kedai Mie Pangsito Sragen, *Jurnal Perkivi*, 1(2), [Online]. Available: <https://journal.perkivi.or.id/index.php/perkivi/article/view/27/26>.
- Ndaru Adi Pratama, Catur Hermawan, S.Kom, 2016. Aplikasi Pembelajaran Tes Potensi Akademik Berbasis Android, *JPDF: Jurnal Penelitian Dosen FIKOM (UNDA)*, 6(1), [Online]. Available: <http://www.jurnal.unda.ac.id/index.php/Jpdf/article/view/11/13>.
- Davit Irawan, Zerian Novianto, 2020. Perancangan E-Learning Pada Sman 1 Kota Lubuklinggau Menggunakan Framework Codeigniter (CI), *Jurnal Digital Teknologi Informasi* 3(2), [Online]. Available: <https://scholar.archive.org/work/iproe7aqlbey3o6uw7hkznrsxq/access/wayback/https://jurnal.um-palembang.ac.id/digital/article/download/2690/pdf>.
- Andy Dharmalau, Nur Sucahyo, Indra Mukti, July 2023. Perancangan Aplikasi Point Of Sales (Pos) Berbasis Android Menggunakan Framework Flutter Di Kafe Elangsta, *Jurnal Rekayasa Informasi Swadharma*, 03(02), [Online]. Available: <https://ejurnal.swadharma.ac.id/index.php/jris/article/view/326/261>.
- Yudho Yudhanto, Helmi Adi Prasetyo, October 2018. Panduan Mudah Belajar Framework Laravel, *Jakarta: Elex Media Komputindo, Inc*, [Online]. Available: <https://books.google.co.id/books?id=TpV1DwAAQBAJ&printsec=frontcover&hl=id#v=onepage&q&f=false>.
- Taylor Otwell, 2011. Pencipta Laravel. Laravel Documentation.
- Yosef Koko Kurniawan, Yetli Oslan, Harianto Kristanto, November 2013. Implementasi Rest - Api Untuk Portal Akademik Ukdwberbasis Android, *Jurnal EKSIS*, 06(02), [Online]. Available: <https://vlabti.ukdw.ac.id/ojs/index.php/eksis/article/view/368/136>.
- M. Ilyas Tri Khaqiqi, Nisa Hanum Harani, Bandung Barat 2023. Penerapan Metode Gamifikasi Pada Rest API Spring Boot, *Penerbit Buku Pedia*, Bandung Barat. [Online]. Available: [https://books.google.co.id/books?hl=id&lr=&id=t76-EAAAQBAJ&oi=fnd&pg=PA1&dq=jurnal+tetang+pengertian+rest+api&ots=VR7T1lsCnI&sig=G82actlmGRL96uUK5YC5R\\_5Dnw8&redir\\_esc=y#v=onepage&q&f=false](https://books.google.co.id/books?hl=id&lr=&id=t76-EAAAQBAJ&oi=fnd&pg=PA1&dq=jurnal+tetang+pengertian+rest+api&ots=VR7T1lsCnI&sig=G82actlmGRL96uUK5YC5R_5Dnw8&redir_esc=y#v=onepage&q&f=false).
- Muhammad Romzi, Budi Kurniawan, December 2020. "Implementasi Pemrograman Python Menggunakan Visual Studio Code, *Jurnal Informatika dan Komputer*, 11(2), Available: <https://journal.unmaha.ac.id/index.php/jik/article/view/198/176>.
- Sudjiran, Mohamad Saefudin, Safrido Ahmad Perdana, June 2023. Manajemen Rancangan UI/UX Sistem Digital Pengajuan Peminjaman Permodalan Biaya Pertanian Menggunakan Perangkat Lunak Figma, *JISICOM (Journal of Information System, Informatics and Computing)*, 7(1), [Online]. Available: <https://journal.stmikjayakarta.ac.id/index.php/jisicom/article/view/1090/715>.
- Eko Suprpto, October 2021. User Acceptance Testing (UAT) Refreshment PBX Outlet Site BNI Kanwil Padang, *Jurnal Civronlit Unbari*, 6(2), [Online]. Available: <http://jt.unbari.ac.id/index.php/CIVRONLIT/article/view/85/72>.

- Destiarini, Abdul Rahman, Ketut Sumartayasa, 2023. Analisa Kualitas Website BPJS Kesehatan Dengan Metode WebQual 4.0 Dan User Acceptance Testing Di Wilayah Kabupaten Ogan Komering Ulu, *Jurnal Media Infotama*, 19(2), [Online]. Available: <https://jurnal.unived.ac.id/index.php/jmi/article/view/3911/3616>.
- M. Leo Adi Saputra, 2022. Analisis Sistem Keuangan Desa (SISKEUDES) Di Kecamatan Muara Sugihan Menggunakan Metode *Black Box Testing*, *Skripsi Kementerian Pendidikan, Kebudayaan, Riset, Dan Teknologi, Institut Teknologi Dan Bisnis Palcomtech, Palembang* [Online]. Available: [http://repo.palcomtech.ac.id/id/eprint/1658/1/SKRIPSI\\_IF\\_2022\\_M.%20LEO%20ADI%20SAPUTRA.pdf](http://repo.palcomtech.ac.id/id/eprint/1658/1/SKRIPSI_IF_2022_M.%20LEO%20ADI%20SAPUTRA.pdf).
- Tri Sandhika Jaya, January 2018. Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis, *JPIT (Jurnal Pengembangan IT)*, 03(02), [Online]. Available: <https://ejournal.poltekharber.ac.id/index.php/informatika/article/view/647/640>.
- Nurman Hidayat, Kusuma Hati, February 2021. Penerapan Metode *Rapid Application Development* (RAD) dalam Rancang Bangun Sistem Informasi Rapor Online (SIRALINE), *Jurnal Sistem Informasi Stmik Antar Bangsa*, X(1), [Online]. Available: <https://ejournal.antarbangsa.ac.id/jsi/article/view/352>.
- Dian Gustina, Yudi Irawan Chandra, November 2015. Aplikasi Sistem Pakar Untuk Mendiagnosa Penyakit Paru Pada Anak Menggunakan Metode *Rapid Application Development* (RAD), *Jurnal.FTUMJ*, [Online]. Available: <https://core.ac.uk/download/pdf/290154571.pdf>.
- Fiftin Noviyanto, Tedy Setiadi, Iis Wahyuningsih, January 2014. Implementasi SIKADES (Sistem Informasi Kependudukan Desa) Untuk Kemudahan Layanan Administrasi Desa Berbasis Web Mobile, *Jurnal Informatika*, 8(1), [Online]. Available: <https://media.neliti.com/media/publications/101999-ID-implementasi-sikades-sistem-informasi-ke.pdf>.
- Erga Trivaika, Mamok Andri Senubekti, 2022. Perancangan Aplikasi Pengelola Keuangan Pribadi Berbasis Android, *Jurnal.Uniku*, [Online]. Available: <https://journal.uniku.ac.id/index.php/ilkom/article/view/4670/2896>