

BAB I

PENDAHULUAN

A. Latar Belakang

Saat ini Pengguna Teknologi sangat di perlukan bagi setiap orang untuk mengelolah informasi dibutuhkan teknologi informasi yang baik, karena informasi yang baik akan semakin berkembang dan bersaing dengan teknologi terbaru lainnya.

Internet adalah salah satu bentuk kemajuan dan teknologi dan informasi pada saat ini. Internet atau yang bisa di kenal sebagai dunia maya bukan sesuatu hal yang baru, cepatnya berkembang di dunia internet membuat segalanya lebih muda, termasuk dalam menyampaikan informasi seolah dunia menjadikan terbatas.

Panti Asuhan Abadi Asiyiah adalah panti asuhan tertua yang terletak di Kota Parepare. Panti asuhan ini dibangun secara bertahap yang peletakan batu pertamanya pada tahun 1960, dan mulai ditempati pada tahun 1962, sehingga pada tahun berdirinya Panti Asuhan Abadi Asiyiah Kota Parepare pada tahun 1963.

Panti asuhan abadi asiyiah oleh Pimpinan Daerah Muhammdiyah Kota Parepare. Panti asuhan ini menampung anak-anak dari berbagai umurr dan berbagai golongan seperti anal yatim piatu, anak yatim dan orang tidak mampu.

Sumbangan yang masuk di panti asuhan terbagi menjadi 2 jenis yaitu berupa sumbangan barang dan uang. Dari sumbangan tersebut bentuk pertanggung jawaban pihak pengelola panti asuhan kepada para penyumbang.

Dalam sistem yang berjalan, data-data yang dikelola dalam Panti Asuhan Abadi Aisyiyah masih menggunakan pencatatan dana manual yaitu buku besar di samping itu juga sebelumnya penelitian sebelum sudah dibuatkan Aplikasi Pengelolah Data Donatur Dan Dana Sumbangan Pada Panti Asuhan Berbasis Client Server dan sekarang ingin di kembangkan dengan Sistem Berbasis Android.

Dalam segala pertimbangan tersebut, maka penulis berinisiatif membuat pengembangan aplikasi dana sosial pada panti asuhan berbasis android dalam pengelolah serta mempermudah proses dalam mengelolah panti asuhan.

Berdasarkan latar belakang yang telah diuraikan di atas, maka penulisan tertarik untuk mengangkat judul “**Pengembangan Aplikasi Dana Sosial Pada Panti Asuhan Berbasis Android**”.

B. Rumusan Masalah

Berdasarkan uraian latar belakang yang telah diuraikan, maka rumusan masalah yang akan dibahas yaitu bagaimana Pengembangan Aplikasi Dana Sosial Pada Panti asuhan Berbasis Android?

A. Tinjauan Teori

1. Aplikasi (*Application*)

Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata kerja to *apply* yang dalam Bahasa Indonesia berarti pengolah. Secara istilah, aplikasi komputer adalah suatu *sub* kelas perangkat lunak komputer yang menggunakan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Menurut Kamus Besar Bahasa Indonesia (KBBI, 2015) Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Sementara menurut Hartono (1999) aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.

Fungsi aplikasi Setiap aplikasi yang diciptakan tentu memiliki tujuan dan fungsi tersendiri. Di bawah ini terdapat beberapa fungsi aplikasi dalam berbagai bidang kehidupan.

1. Pendidikan

Dalam konteks pendidikan, aplikasi dapat digunakan sebagai sarana pembelajaran yang interaktif, seperti halnya aplikasi Microsoft PowerPoint.

Dengan menggunakan aplikasi ini, proses belajar-mengajar dapat menjadi lebih dinamis dan interaktif. Aplikasi MS PowerPoint memungkinkan para

pengajar untuk menciptakan presentasi yang menarik dan beragam, sehingga suasana kelas menjadi lebih hidup dan menarik perhatian para siswa.

2. Bisnis

Dalam bidang bisnis, aplikasi dapat memiliki banyak peran. Microsoft Excel adalah salah satu contoh fungsi aplikasi yang dapat digunakan untuk menghitung keuntungan bisnis dan mengelola keuangan. Aplikasi seperti ini lebih memudahkan perusahaan dalam menghitung secara cepat, sehingga tidak perlu cara manual lagi.

3. Kedokteran

Fungsi aplikasi berikutnya adalah dalam bidang kedokteran, salah satu penerapan aplikasi pada bidang ini yaitu kemudahan konsultasi jarak jauh antara pasien dengan dokter dengan menggunakan aplikasi telemedis. Dengan aplikasi tersebut, pasien tidak harus selalu datang ke rumah sakit untuk berkonsultasi dengan dokter, hal ini tentu akan lebih menghemat waktu dan tenaga bagi pasien.

4. Militer

Aplikasi tidak hanya bermanfaat dalam pendidikan, bisnis, atau kedokteran saja, tetapi juga memiliki peran krusial dalam militer. Contohnya adalah aplikasi yang digunakan untuk mengontrol pesawat. Penggunaan aplikasi semacam ini dapat memberikan beberapa keuntungan seperti efisiensi operasional, peningkatan presisi, hingga analisis data *real-time*.

5. Ilmu Pengetahuan

Aplikasi juga berperan dalam bidang ilmu pengetahuan, salah satunya yaitu adanya aplikasi yang mampu membantu menganalisis data besar dan kompleks sehingga mendapatkan wawasan ilmiah. Selain itu juga terdapat aplikasi yang memungkinkan pengguna dapat mengakses jurnal, artikel, dan referensi ilmiah untuk keperluan penelitian.

Jenis-jenis Aplikasi dibagi menjadi beberapa jenis yang dibedakan berdasarkan penggunaannya. Diantaranya sebagai berikut.

1. Real Time Software

Real-time software adalah jenis aplikasi yang memiliki kegunaan penting dalam melakukan pengamatan, pengendalian, dan analisis langsung terhadap situasi di dunia nyata. Dengan menggunakan *real-time software*, kita dapat mengakses dan merespons peristiwa atau data secara instan tanpa ada jeda waktu yang berarti.

Di beberapa bidang, salah satunya dunia kesehatan, *real-time software* digunakan untuk memantau parameter vital pasien dan memberikan peringatan dini jika ada perubahan yang signifikan.

2. System software

Jenis aplikasi berikutnya yaitu *system software* yang memiliki peran penting dalam mengontrol dan mengelola proses operasional internal pada sistem komputer. Aplikasi ini berfokus pada tugas-tugas yang mendukung operasi dan fungsionalitas *hardware* dan *software* pada komputer.

Contoh *system software* dapat mencakup *driver* perangkat keras, yang berfungsi sebagai perantara antara perangkat keras dan sistem operasi. Driver ini memungkinkan sistem operasi mengenali dan berkomunikasi dengan perangkat keras, sehingga perangkat tersebut dapat bekerja dengan benar.

3. Business software

Business software adalah jenis aplikasi yang dapat membantu memenuhi kebutuhan dan tugas-tugas bisnis seseorang atau perusahaan. Tujuan aplikasi ini adalah untuk meningkatkan efisiensi, produktivitas, dan pengelolaan proses bisnis secara keseluruhan. Contohnya seperti aplikasi pengelola keuangan, aplikasi manajemen SDM, dan masih banyak lagi.

4. Personal computer software

Personal computer software adalah jenis aplikasi yang dirancang untuk digunakan oleh pengguna baik secara resmi maupun pribadi pada PC (*personal computer*). Aplikasi ini mencakup berbagai macam *software* yang dapat diinstal dan dijalankan yang berfungsi untuk memenuhi kebutuhan dan preferensi individu pengguna. Contoh seperti aplikasi produktivitas seperti Microsoft Office.

5. Web-based software

Web-based software adalah jenis aplikasi yang berfungsi sebagai antarmuka penghubung antara pengguna dengan internet. Aplikasi ini dirancang agar bisa diakses dan digunakan melalui *web browser* tanpa perlu instalasi perangkat lunak tambahan. Dengan kata lain, aplikasi web ini berjalan di server jarak jauh dan

memungkinkan pengguna untuk berinteraksi dengan layanan dan data melalui internet.

6. *Engineering and scientific software*

Engineering and scientific software adalah jenis aplikasi yang dirancang dan dikembangkan untuk membantu manusia menyelesaikan berbagai masalah yang khusus, terutama dalam bidang teknik dan ilmu pengetahuan. Aplikasi ini didesain untuk mengatasi masalah yang bersifat khusus dan kompleks, yang seringkali tidak dapat dipecahkan dengan algoritma sederhana. Salah satu contoh *engineering and scientific software* yaitu *software CAD (Computer-Aided Design)* yang digunakan insinyur dan desainer untuk merancang dan mengembangkan produk secara digital dan menghasilkan gambar serta model 3D.

2. *Android*

Android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang Menggunakan *Linux*. Menurut (Hermawan, 2011), *Android* merupakan OS (*Operating System*) *Mobile* yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. Berdasarkan pendapat diatas, maka dapat ditarik kesimpulan bahwa *Android* adalah sistem operasi Menggunakan *Linux* yang sedang berkembang ditengah OS lainnya. (Arifianto, 2011)

Ada beberapa macam *android* dari yang tingkatan pertama hingga terbaru dengan beberapa fitur *software* yang melengkapinya sebagai berikut.

- a) *Android Software Development Kit (SDK)*

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*. *Android* merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang release oleh *Google*. Saat ini di sediakan *Android SDK (Software Development Kit)* sebagai alat bantu dan *API* untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman java (Nazruddin, 2011).

b) *Java SE Development Kit (JDK)*

JDK adalah perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode *java* ke *bytecode* yang dapat dimengerti dan dapat dijalankan oleh JRE (*Java Runtime Envirotment*). JDK wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi Menggunakan *java*, namun tidak wajib terinstall di komputer yang akan menjalankan aplikasi yang dibangun dengan *java*. (Nazruddin, 2011).

c) *Android Development Tools (ADT)*

Android Development Tools (ADT) yang lebih dikenal *plugin Eclipse*. *Plugin* ini yang membuat *Eclipse* dapat membuat *project* yang menggunakan *Android*. ADT adalah *plugins* di *Eclipse* yang harus diinstall sehingga *Android SDK* dapat dihubungkan dengan IDE *Eclipse* yang digunakan sebagai tempat coding aplikasi *Android* nantinya. ADT adalah kepanjangan dari *Android Development Tolls* yang menghubungkan antara IDE *Eclipse* dengan *Android SDK* (Nazruddin, 2011). *Android Virtual Device (AVD)* yang merupakan emulator

untuk menjalankan program aplikasi *Android* yang dibuat. AVD ini nantinya yang kita jadikan sebagai tempat tes dan menjalankan aplikasi *Android* yang dibuat. AVD berjalan di *virtual Machine* (Nazruddin, 2011).

3. *Android Studio*

Android Studio adalah *tools Integrated Development Environment (IDE)* resmi hasil kolaborasi antara Google dan JetBrains yang dibuat khusus untuk pengembangan aplikasi android.

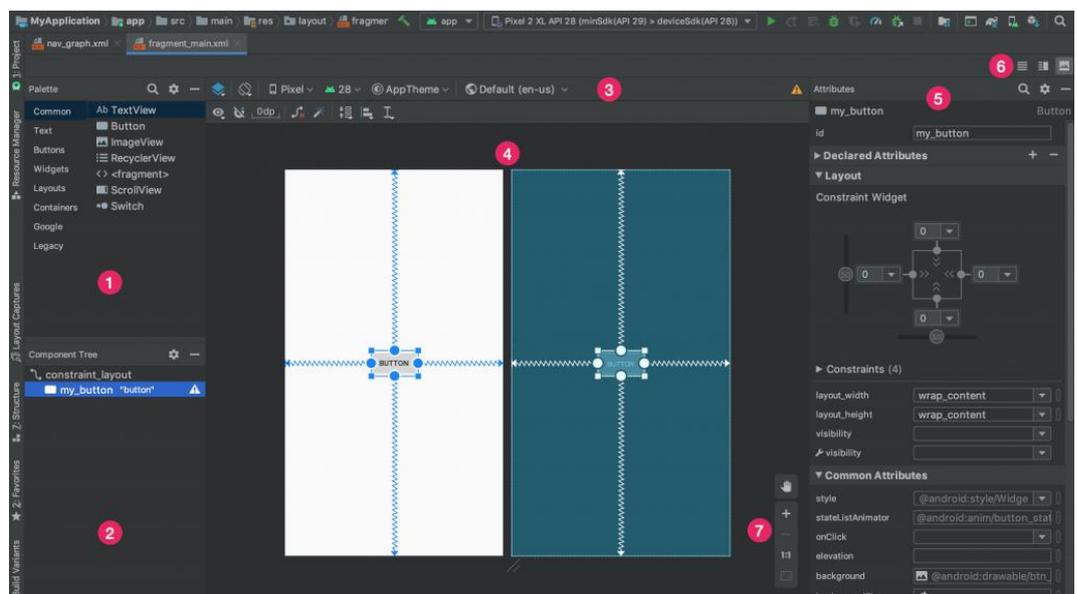
Android studio dikembangkan dari ***IntelliJ IDEA***, sebuah IDE buatan yang juga memproduksi aplikasi seperti PHPStorm dan PyCharm. Sebagai *IDE*, tentu saja *Android Studio* memiliki komponen yang lengkap meliputi *source code editor*, *compiler* dan *debugger*. *Android Studio* diluncurkan pada tanggal 16 Mei 2013 pada acara Google's I/O 2013. Sebelumnya, Google memberikan dukungan terhadap *Eclipse IDE*. *Plugin Eclipse Android Developer Tool (ADT)* dibuat agar Eclipse bisa melakukan kompilasi dan test run project Android.

Seiring berkembangnya aplikasi mobile, Google menghentikan support ADT untuk Eclipse dan beralih ke *Android Studio* karena dianggap memiliki fitur yang lebih lengkap. Sebagai contoh, *Android Studio* mendukung bahasa pemrograman lain seperti *Kotlin* dan memudahkan membuat aplikasi dengan framework Javascript ataupun *Flutter*. Nah, karena *Android Studio* adalah IDE yang memuat semua tools untuk melakukan design, testing dan debug, tool ini cukup menguras RAM pada perangkat komputer.

Oleh karena itu, untuk menggunakan Android Studio, minimal Anda memerlukan spesifikasi sistem sebagai berikut:

- Sistem Operasi : *Windows 8/10 64-bit, MacOS 10.14, distro Linux 64-bit yang mendukung Gnome, KDE atau UnityDE*
- RAM 8 GB
- Ruang Penyimpanan 8 GB (IDE + Android SDK + Android Emulator)
- Resolusi layar 1280 x 800

Nah, itu dia spesifikasi yang Anda butuhkan. Jika masih ragu, intip dulu fitur-fitur *Android Studio* yang bisa membantu Anda mengembangkan aplikasi *Android* terbaik! Layout Editor memiliki tujuh bagian utama yang bisa Anda lihat pada gambar berikut ini:



Gambar 2.1.Android Studio

1. **Palette:** Bagian ini berisi UI elemen yang bisa Anda drag ke layout.
2. **Component Tree :** Bagian ini menunjukkan struktur komponen pada layout Anda.
3. **Toolbar :** Bagian ini berfungsi untuk mengatur tampilan layout editor dan mengubah atribut layout.
4. **Design editor :** Bagian ini merupakan tempat untuk mengedit tampilan desain dan blueprint.
5. **Attributes :** Bagian ini digunakan untuk mengontrol tampilan dari atribut yang dipilih
6. **View mode :** Bagian ini berfungsi untuk mengatur tampilan layout baik tampilan code, design maupun split mode.
7. **Zoom and pan controls:** Bagian ini berguna untuk mengontrol ukuran preview dan posisi dalam editor (Aorinka Anendra, 2023).

4. *Flutter*

Flutter adalah sebuah framework open source yang dapat digunakan untuk mengembangkan aplikasi multiplatform, seperti Android, web, iOS, Windows, MacOS, *Linux*, hingga Fuchsia. Framework ini pertama kali diperkenalkan oleh Google pada Mei tahun 2017. Dengan Flutter, developer dapat membuat aplikasi dengan latar belakang platform yang berbeda dari basis kode yang sama.

Dukungan terhadap berbagai platform membuat Flutter menjadi salah satu framework pengembangan aplikasi terpopuler dan paling banyak digunakan hingga

menduduki peringkat ke-4. Berdasarkan survey *Statista*, penggunaan Flutter terus meningkat dari tahun ke tahun seperti pada grafik berikut. Bahkan setelah 2 tahun dirilis, pada tahun 2019 pengguna Flutter telah meningkat sebesar 30%.

Flutter memiliki dua komponen utama, yaitu Flutter SDK (Software Development Kit) dan framework UI. Flutter SDK (Software Development Kit) – merupakan kumpulan tools dan perangkat lunak yang digunakan untuk mengembangkan aplikasi dengan menggunakan Flutter yang terdiri dari berbagai komponen, termasuk compiler, debugger, emulator, dan berbagai library yang diperlukan untuk mengembangkan aplikasi Flutter.

Framework UI – komponen yang digunakan untuk membangun antarmuka pengguna aplikasi. Diantaranya menyediakan seperti widget, tombol, navigasi, dan banyak lagi yang memungkinkan developer membuat tampilan aplikasi yang menarik dan konsisten di berbagai platform.

dua komponen utama ini, Flutter dapat membantu developer dalam membangun aplikasi cross-platform yang berkinerja tinggi dengan hanya menggunakan satu basis kode.

Cara Kerja Flutter Dalam pengembangannya, Dart adalah bahasa pemrograman yang digunakan untuk membangun aplikasi Flutter. Bahasa ini dikembangkan Google dan memiliki sintaks yang mirip dengan bahasa pemrograman lain seperti JavaScript dan Java.

Hal ini membuat Flutter mudah dipelajari oleh pengembang yang sudah memiliki pengalaman dalam bahasa-bahasa tersebut maupun bagi para pemula. Jika kamu tertarik mempelajarinya, berikut ini terdapat cara kerja framework Flutter.

Developer menggunakan bahasa Dart dan widget bawaan dari Flutter untuk mengembangkan aplikasi. Di Android, kode dikompilasi dengan mesin C++ menggunakan Android NDK (Native Development Kit), sedangkan di iOS, kode dikompilasi dengan LLVM (Low-Level Virtual Machine). Pada platform Android, kode native kemudian dikompilasi kembali dengan Dart compiler.

Setelah proses tersebut, kode akan disesuaikan agar bisa digunakan di lintas platform.

Kelebihan Flutter Kepopuleran Flutter di lingkungan para developer tentu tidak terlepas dari tawaran keunggulan-keunggulannya. Maka dari itu, ini dia beberapa kelebihan Flutter yang menarik minat para pengembang.

a. Lebih mudah dan cepat mengembangkan aplikasi

Fitur yang disediakan oleh Flutter framework membuat pengembangan aplikasi menjadi lebih mudah dan cepat. Hal ini karena adanya user interface widgets yang telah dikembangkan oleh Google dengan kualitas coding yang sangat baik. Penggunaan widgets ini memudahkan pengembang dalam melakukan kustomisasi tanpa perlu membuat coding dari nol. Selain itu, Flutter juga menyajikan fitur Hot Reload yang memungkinkan pengembang untuk melihat hasil perubahan coding secara instan. Fitur ini mempercepat proses pengembangan yang memungkinkan pengembang untuk melakukan eksperimen dalam pembuatan desain tanpa harus menunggu waktu lama untuk melihat perubahan tersebut.

b. Adanya custom user interface yang menarik

Salah satu elemen penting dalam pengembangan aplikasi adalah tampilan dan interaksi pengguna yang menarik. Menurut situs marketplash.com berdasarkan

penelitian dari Forester, user interface yang baik dapat meningkatkan tingkat konversi situs web hingga 200%, sedangkan user experience yang dimaksimalkan dapat meningkatkan tingkat konversi hingga 400%.. Data ini menunjukkan bahwa user interface yang menarik penting pada sebuah aplikasi, baik web maupun yang lain.

Dengan menggunakan Flutter, developer dapat dengan mudah membuat antarmuka pengguna yang menarik dan disesuaikan dengan kebutuhan aplikasi mereka. Flutter menyediakan beragam widget yang dapat digunakan untuk membangun tampilan UI aplikasi yang menarik. Pengembang dapat mengakses koleksi widget yang sudah ada untuk digunakan dalam pembuatan aplikasi, mulai dari layout, animasi, font, navigasi, hingga efek tampilan scrolling. Widget ini dapat dikustomisasi sesuai dengan kebutuhan, memungkinkan pengembang untuk menciptakan tampilan aplikasi yang unik dengan mudah.

c. *Performa yang mirip dengan aplikasi native*

Meskipun Flutter adalah aplikasi multiplatform, namun performanya hampir sama baiknya dengan *aplikasi native*. Hal ini karena rendering engine yang kuat dan kemampuan Flutter untuk menghasilkan kode yang dioptimalkan untuk berbagai platform. Sehingga, meskipun aplikasi dibuat untuk berbagai platform, hasilnya tetap optimal.

d. *Biaya yang dikeluarkan lebih hemat*

Terakhir adalah biaya yang dikeluarkan lebih hemat. Hal ini karena Flutter mendukung multi platform, sehingga pengembang tidak perlu mengembangkan aplikasi dari awal untuk setiap platform yang berbeda. Fitur-fitur seperti UI widget,

Hot Reload, data backend, aksesibilitas, integrasi platform, dan plugin juga membuat pengembangan aplikasi menjadi lebih mudah. Dengan kode yang bisa digunakan kembali untuk berbagai platform, hal tersebut dapat menghemat tenaga, waktu, dan biaya bagi pengembangan.

Kekurangan Flutter Meski menawarkan berbagai keunggulan, framework Flutter juga masih memiliki kekurangan. Berikut ini beberapa kekurangan Flutter yang dapat dijadikan pertimbangan sebelum menggunakannya.

1) *Library pihak ketiga yang belum banyak*

Salah satu kekurangan utama Flutter adalah ketersediaan library pihak ketiga yang belum sebanyak platform lain. Karena masih baru, komunitas pengembangnya juga masih dalam belum begitu banyak. Sehingga ketersediaan resource code atau library untuk Flutter masih terbatas.

Library ini berisi sejumlah fungsi coding yang dapat digunakan kembali untuk mempermudah pembuatan aplikasi. Namun, sayangnya, jumlah library yang tersedia untuk pengembangan aplikasi di Flutter masih terbatas. Oleh karena itu, pengembang perlu melakukan banyak uji coba untuk mengatasi keterbatasan ini.

2) *Aplikasi yang dihasilkan berukuran besar*

Meskipun aplikasi yang dikembangkan dengan Flutter menunjukkan kinerja yang stabil, namun, mereka cenderung memiliki ukuran file yang besar. Sebagai perbandingan, aplikasi sederhana seperti “Hello World!” dalam aplikasi native hanya memiliki ukuran sekitar 500 KB, sedangkan dalam Flutter bisa mencapai lebih dari 4 MB. Kondisi ini dapat menimbulkan masalah bagi pengguna yang memiliki ruang penyimpanan terbatas di perangkat mereka. Selain

itu, ukuran yang besar juga dapat mengurangi performa aplikasi, terutama pada perangkat dengan koneksi internet yang lambat. Dampaknya, pengguna mungkin akan enggan untuk memasang aplikasi yang memiliki ukuran file yang besar.

3) *Perlu untuk belajar bahasa pemrograman yang baru*

Salah satu tantangan dalam menggunakan Flutter adalah perlu belajar bahasa pemrograman baru, yaitu Dart. Meskipun Dart belum begitu populer dan masih sedang berkembang, banyak pengembang yang mulai menggunakannya. Jika kamu sudah mahir dalam bahasa pemrograman Java atau C#, mempelajari Dart mungkin tidak terlalu sulit. Selain itu, Flutter juga menyediakan tutorial video di saluran YouTube-nya untuk membantu pengembang mempelajari Flutter dengan lebih mudah. Meski begitu saat ini juga sudah banyak course tentang Dart dan Flutter.

5. Dart

Bahasa pemrograman multiguna (bahasa pemrograman dengan tujuan umum), Dart, yang dapat digunakan dalam mengembangkan berbagai jenis aplikasi dan program yang digunakan dalam mengembangkan aplikasi web, aplikasi seluler, aplikasi desktop, server, dll. Dart merupakan *programming language* lintas platform atau platform independen yang artinya dapat dijalankan pada sistem operasi yang berbeda seperti Windows, Linux, Unix dan MacOS, dll yang awalnya dikembangkan oleh Google dan kemudian disetujui sebagai standar oleh Ecma, yang saat ini digunakan untuk membangun aplikasi web, server, desktop, dan seluler. Dart awalnya dirancang sebagai *programming language* yang dioptimalkan klien untuk pengembangan cepat aplikasi web dan seluler. Sebagai salah satu dari banyaknya *programming language* yang mendukung multi paradigma, Dart, bersifat imperatif, fungsional, reflektif dan berorientasi objek. Selain itu, Dart juga mengikuti semua konsep pendekatan pemrograman berorientasi objek seperti kelas, pewarisan, abstraksi, enkapsulasi, dan polimorfisme, dll. Dart juga merupakan tipe bahasa pemrograman yang sangat kuat yang menyediakan fitur pengumpul sampah otomatis. Bahasa Dart sendiri bersifat open source yang dilisensikan di bawah BDS. Sintaksnya merupakan sintaksis gaya-C sederhana.

Sejarah Dart dirancang dan dikembangkan oleh Lars Bak dan Kasper Lund di Google yang secara resmi diperkenalkan pada Oktober 2011 di Konferensi GOTO, Denmark dan versi pertama Dart yaitu versi 1.0 resmi dirilis pada Nov 2013. Untuk menstandarisasi *programming language* yang baru dibuat, Ecma International membuat komite teknis yang dikenal sebagai TC52, dan Fakta bahwa kode Dart dapat dikompilasi ke dalam JavaScript telah membuat bahasa ini cukup kompatibel dengan banyak browser web modern. Hal ini menyebabkan standardisasi yang tak terhindarkan pada Juli 2014, ketika Ecma International menyetujui edisi pertama spesifikasi bahasa Dart di Majelis Umum ke-107 badan internasional tersebut. Sekitar enam bulan kemudian, pada bulan Desember 2014, edisi lain dari spesifikasi bahasa Dart telah disetujui oleh Ecma International. Ini adalah edisi kedua dari spesifikasi bahasa Dart. Kompiler pertama yang dapat menghasilkan kode JavaScript dari Dart dikenal sebagai *dartc*, tetapi kemudian dikutuk oleh Google. Itu diikuti oleh *Frog*, kompiler yang ditulis dalam Dart, tetapi tidak dapat mengkompilasi kode sepenuhnya. Kompiler ketiga adalah *dart2js* yang juga ditulis dalam Dart dan masih digunakan. Ada sepuluh fitur Dart yang menjadikannya salah satu bahasa pemrograman populer, yaitu:

- *Cross Platform* (lintas platform)

Dart merupakan *programming language* lintas platform atau platform independen yang berarti dapat dijalankan pada sistem operasi yang berbeda seperti Windows, Linux, Unix dan MacOS, dll.

- *General Purpose* (tujuan umum)

Dart bersifat multiguna (dengan tujuan umum) yang artinya dapat digunakan dalam mengembangkan berbagai jenis aplikasi dan program.

- *Multi-paradigm* (multiparadigma)

Dart mendukung multi paradigma seperti pemrograman imperatif, fungsional, reflektif dan berorientasi objek.

- *Object-Oriented* (berorientasi pada objek)

Dart adalah pemrograman berorientasi objek dan mengikuti semua konsep pendekatan pemrograman berorientasi objek seperti kelas, pewarisan, abstraksi, enkapsulasi, dan polimorfisme, dll.

- *Simple Syntax* (sintaks yang sederhana)

Sintaks Dart adalah sintaksis gaya-C sederhana. *Programmers* pemula dapat dengan mudah mempelajari sintaks Dart.

- *Automatic Garbage Collector* (pengumpul sampah otomatis)

Dart adalah bahasa yang sangat diketik yang menyediakan fitur pengumpul sampah otomatis.

- *Compiled Programming Language* (bahasa pemrograman yang dikompilasi)

Dart merupakan *programming language* yang memenuhi syarat yang berarti bahwa kompiler digunakan untuk mengubah kode yang ditulis dalam bahasa pemrograman Dart menjadi bahasa asli atau JavaScript sebelum dieksekusi.

- *Open Source* (sumber yang terbuka)

Dart adalah bahasa pemrograman open source yang dilisensikan di bawah BDS.

- *Inbuilt Libraries* (perpustakaan bawaan)

Dart menyediakan berbagai macam perpustakaan inbuilt kelas kaya.

6. PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Prosesor*) adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *browser* menjadi kode HTML. (Diar Puji Octavian 2010) Kode PHP mempunyai ciri-ciri khusus, yaitu :

- Hanya dapat dijalankan menggunakan *web* server misalnya : *Apache*.
- Kode PHP dapat diletakan dan dijalankan di *web* server.
- Kode PHP dapat digunakan untuk mengakses database, seperti: *MY SQL*, *PostgreSQL*, *Oracle*, dan lain-lain.
- Merupakan *software* yang bersifat *open source*.
- Gratis untuk di *download* dan digunakan.
- Memiliki sistem *multiplatform*, artinya dapat dijalankan menggunakan sistem operasi apapun, seperti *Linux*, *Unix*, *Windows*, dan lain-lain.

PHP merupakan bahasa skrip sisi *server*, yang berarti instruksi dalam skrip dieksekusi oleh *server*. Kemudian, server menyediakan data sesuai permintaan, meneruskan permintaan tersebut, lalu menata informasinya dalam database.

Saat menerima *skrip*, *web server* akan memproses permintaan di dalamnya dan mengirimkan output ke web browser dalam format HTML. Database web

server kemudian menyimpan informasi tersebut sehingga pengguna lain tidak bisa mengakses data dan kode sumbernya. Menggunakan skrip, pengguna bisa melakukan tindakan seperti menyesuaikan website, membuat perubahan dinamis pada kontennya, dan mengakses database website. Pengguna juga bisa menggunakan bahasa skrip untuk mengembangkan aplikasi web.

Di samping PHP, beberapa contoh lain bahasa skrip sisi server adalah **Python**, **Ruby**, **ColdFusion**, **Java**, dan **C++**. Sementara itu, bahasa skrip sisi klien menggunakan web browser untuk menjalankan skripnya. Bahasa ini ditujukan untuk bagian front-end, yaitu area website yang terlihat oleh pengunjung. Setiap pengunjung bisa melihat skrip sisi klien website dengan menekan **CTRL + U** di **Windows** atau **Option + CMD + U** di **macOS**. Nah, karena PHP dieksekusi di server, Anda bisa menggunakan PHP dengan HTML secara bersamaan. Jadi, PHP akan menjadi kode sumbernya, tapi pengguna lain hanya akan melihat kode HTML.

Sebagai contoh, mari lihat skrip PHP berikut ini:

```
<html>

<head>

<title>PHP Test</title>

</head>

<body>
```

```
<?php echo '<p>Hello World</p>'; ?>

</body>

</html>
```

Output file PHP ini akan terlihat seperti berikut dalam kode HTML:

```
<html>

<head>

<title>PHP                               Test</title>

</head>

<body>

<p>Hello                               World</p>

</body>

</html>
```

Kode ini akan menampilkan pesan ***Hello World*** sebagai output di *web browser*.

Intinya, semua kode PHP akan mengikuti format yang dimulai dengan **<?php** dan diakhiri dengan **?>** setelah instruksinya.

Perlu diketahui bahwa tag PHP peka terhadap huruf besar atau kecil, jadi berhati-hatilah dalam menggunakan huruf besar pada skrip Anda.

7. JavaScript

JavaScript adalah bahasa pemrograman tingkat tinggi dan dinamis. *JavaScript* populer di internet dan dapat bekerja di sebagian besar penjelajah *web* populer seperti *Google Chrome*, *Internet Explorer (IE)*, *Mozilla Firefox*, *Netscape* dan *Opera*. Kode *JavaScript* dapat disisipkan dalam halaman *web* menggunakan tag *SCRIPT*. *JavaScript* merupakan salah satu teknologi inti *World Wide Web* selain *HTML* dan *CSS*. *JavaScript* membantu membuat halaman *web* interaktif dan merupakan bagian aplikasi *web* yang esensial. *Javascript* diperkenalkan pertama kali oleh *Netscape* pada tahun 1995. Pada awalnya bahasa ini dinamakan “*LiveScript*” yang berfungsi sebagai bahasa sederhana untuk browser *Netscape Navigator 2*. Pada masa itu bahasa ini banyak di kritik karena kurang aman, pengembangannya yang terkesan buru buru dan tidak ada pesan kesalahan yang di tampilkan setiap kali kita membuat kesalahan pada saat menyusun suatu program. Kemudian sejalan dengan sedang giatnya kerjasama antara *Netscape* dan *Sun* (pengembang bahasa pemrograman “*Java*”) pada masa itu, maka *Netscape* memberikan nama “*JavaScript*” kepada bahasa tersebut pada tanggal 4 desember 1995. Pada saat yang bersamaan *Microsoft* sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai “*script*” di browser *Internet Explorer*

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen *HTML*, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk *web*. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa *HTML* dengan

mengijinkan pegeksekusian perintah perintah di sisi user, yang artinya di sisi browser bukan di sisi server web. Javascript bergantung kepada browser(navigator) yang memanggil halaman web yang berisi skrip skrip dari Javascript dan tentu saja terselip di dalam dokumen HTML. Javascript juga tidak memerlukan kompilator atau penterjemah khusus untuk menjalankannya (pada kenyataannya kompilator Javascript sendiri sudah termasuk di dalam browser tersebut). Lain halnya dengan bahasa “Java” (dengan mana JavaScript selalu di banding bandingkan) yang memerlukan kompilator khusus untuk menterjemahkannya di sisi user/klien. Tabel daftar navigator dan versi dari Javascript :

Versi Javascript	Browser
1.2	Internet Explorer 4.0
1.1	Netscape Navigator 4.0/4.05
1.0	Netscape Navigator 3.0
1.5	Netscape Navigator 2.0
1.3	Internet Explorer 3.0
1.4	Netscape Navigator 6.0
1.3	Internet Explorer 5.0
1.4	Netscape Navigator 4.06
1.5	Internet Explorer 5.5

1 Kuliah Umum IlmuKomputer.Com Copyright © 2003 Ilmu Komputer.Com

Janganlah anda bingung dengan perbandingan antara Javascript dan Java, karena berdasarkan pengalaman yang saya peroleh baik dari milist milist maupun forum forum banyak yang belum bisa menemukan perbedaan jelas antara keduanya, oleh karena itu saya akan jelaskan berikut ini. Javascript merupakan suatu bahasa yang perkembangannya lambat di bandingkan dengan Java yang berkembang sangat cepat. Di Javascript kita tidak mungkin menyembunyikan kode skrip yang kita tulis, kode langsung di tulis di dalam dokumen HTML dan sangat mudah terlihat, sedangkan di Java, kode sudah berbentuk setengah terkompilasi (dalam bentuk applet) dan tidak mungkin terlihat dari dalam dokumen HTML, satu mesin virtual

di sisi user yang bertanggung jawab untuk menterjemahkan program di dalam applet tersebut setiap kali halaman HTML yang memuat applet tersebut dipanggil oleh browser. Dibandingkan dengan applet java yang cukup lambat dibuka oleh browser, bisa kita katakan bahwa Javascript cukup cepat di panggil(di load) oleh navigator. JavaScript sendiri merupakan bahasa yang mudah dipahami, dalam artian diperlukan skill novice atau dasar untuk mengerti bahasa ini, jika anda sudah terbiasa dan mengenal konsep bahasa pemrograman visual, maupun Java ataupun C, akan sangat mudah untuk memahami konsep Javascript. Berikut ini satu tabel yang berisi beberapa perbandingan mendasar antara Java dan JavaScript :

JavaScript	Java
Bahasa yang diinterpretasikan langsung oleh browser	Bahasa yang setengah terkompilasi dan memerlukan Java Virtual Machine untuk menterjemahkannya
Kode terintegrasi dengan HTML	Kode(applet) terpisah dari dokumen HTML, dipanggil pada saat membuka dokumen HTML
Bahasa dengan karakteristik yang t erbatas	Bahasa dengan karakteristik yang luas (pendeklarasian jenis variabel)
Hubungan dinamis, referensi dari obyek diverifikasi pada saat loading.	Hubungan statis, obyek harus ada pada saat program di loading (di kompilasi)
Kode program bisa di akses	Kode program tersembunyi
JavaScript adalah bahasa yang “case sensitive” artinya membedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh variabel atau fungsi dengan nama TEST berbeda dengan variabel dengan nama test.	
Dan yang terakhir seperti bahasa Java ataupun C, setiap instruksi diakhiri dengan karakter titik koma (;).	

(Alamsyah, 2003)

Awalnya hanya diimplementasi sebagai *client-side* dalam penjelajah *web*, kini *engine JavaScript* disisipkan ke dalam perangkat lunak lain seperti dalam *server-side* dalam server *web* dan basis data, dalam program *non web* seperti perangkat lunak pengolah kata dan pembaca *PDF*, dan sebagai *runtime environment* yang memungkinkan penggunaan *JavaScript* untuk membuat aplikasi *desktop* maupun *mobile*. (Madcoms Madiun, 2012)

8. HTTP (*Hypertext Transfer Protocol*)

HTTP (*Hypertext Transfer Protocol*) adalah *protocol* yang dipergunakan untuk mentransfer dokumen. *Protocol* ini adalah *protocol* ringan, tidak berstatus yang dapat dipergunakan berbagai macam *tipe* dokumen. Pengembangan HTTP dikoordinasikan oleh *Konsorsium World Wide Web* (W3C) dan grup bekerja *Internet Engineering Task Force* (IETF), bekerja dalam publikasi satu seri RFC, yang paling terkenal RFC yang menjelaskan HTTP/1,1, versi HTTP yang digunakan umum sekarang ini. HTTP adalah sebuah *protocol* meminta / menjawab antara *client* dan *server*.

Sebuah *client* HTTP seperti web *browser*, biasanya melalui permintaan dengan membuat hubungan *TCP/IP* ke port tertentu. Sebuah server HTTP yang mendengarkan di *port* tersebut menunggu *client* mengirim kode permintaan di ikuti dengan pesan *MIME* yang memiliki data tertentu.

Fungsi HTTP yaitu mengatur format dan bagaimana data ditransmisikan. HTTP juga berfungsi untuk mengatur bagaimana web server dan browser memproses berbagai macam perintah yang masuk.

Contohnya saja, ketika Anda memasukkan domain (URL) di dalam browser. URL yang Anda masukkan tersebut merupakan sebuah perintah ke dalam web server untuk memberikan data halaman website sesuai dengan alamat yang diakses. Hasil dari perintah ini adalah tampilan halaman website yang muncul melalui web browser.

Fungsi lain dari HTTP adalah mengamankan data dari pencurian dan hacker. Hal ini ditandai dengan munculnya HTTPS (*Hypertext Transfer Protocol Secure*). Secara fungsi sama, yaitu mengatur bagaimana data diproses. Hanya saja, HTTPS adalah protokol versi aman dibandingkan dengan HTTP.

Data yang diproses menggunakan HTTPS akan dienkripsi terlebih dahulu. Cara ini bertujuan menghalangi hacker atau user yang mencoba mencuri atau memata-matai proses pertukaran data yang sedang berlangsung. Data yang tidak dienkripsi akan lebih mudah dibaca dan tentu saja itu akan merugikan. Apalagi data tersebut merupakan data penting seperti username, password, PIN, identitas pribadi, dan lain sebagainya.

Cara Kerja HTTP adalah:

1. HTTP klien mengirimkan permintaan informasi ke web server
2. HTTP server memproses permintaan klien, sedangkan HTTP klien menunggu proses selesai
3. HTTP server memberikan informasi yang diminta. Nantinya, web server tidak akan mengingat apapun koneksi yang sudah terjadi.

Setelah permintaan dan respon sudah selesai dijalankan, web server tidak mengingat apapun koneksi yang sudah terjadi. Jadi, ketika HTTP mengirimkan permintaan ulang, web server adalah memprosesnya seperti permintaan awal, itulah mengapa protokol HTTP disebut sebagai *stateless protocol*.

Sedangkan seluruh proses pengiriman permintaan dari web client sampai dengan respon yang diberikan oleh web server merupakan HTTP Transaction.

Sayangnya, HTTP tak lepas dari kesalahan ketika memproses data. Selengkapnya simak daftar HTTP status code dan cara mengatasinya. Anda juga bisa belajar cara mengatasi HTTP eror 500 dan HTTP eror 503.

HTTP adalah komponen website yang mengatur pertukaran data yang terjadi di dalam internet. Protokol inilah yang membuat web client (browser) dan web server (aplikasi web) dapat terhubung. Protokol HTTP mengatur proses transmisi dan bagaimana format data dikirimkan.

Itulah mengapa protokol HTTP merupakan bagian yang penting. Saking pentingnya HTTP terus dikembangkan ditandai dengan munculnya protokol HTTPS yang menggantikan versi HTTP lama. Dalam praktiknya, HTTPS menggunakan teknologi bernama HSTS. HSTS adalah metode di mana website akan memaksa browser memakai sambungan HTTPS.

Oh iya, kalau Anda ingin punya situs HTTPS, Niagahoster menyediakan SSL murah terbaik, lho. Anda bisa mendapatkannya gratis saat membeli hosting Niagahoster, ataupun membelinya secara terpisah (Deka M Wildan, 2014).

9. *PhpMyAdmin*

PhpMyAdmin adalah suatu perangkat lunak yang ditulis dalam bahasa pemrograman PHP digunakan untuk menangani administrasi *MySQL* dengan menggunakan protokol WWW secara langsung dari *browser*. *PhpMyAdmin* mendukung berbagai operasi *MySQL*, diantaranya mengelola basis data, tabel-tabel, bidang *fields*, relasi *relations*, indeks, pengguna *users*, perijinan *permissions*, dan lain-lain. Dengan menggunakan *PhpMyAdmin* diharapkan akan memudahkan dalam memanajemen database *MySQL*. (Alan Nur Aditya, 2011)

Fungsi phpMyAdmin Sebagai alat administrasi database yang populer, phpMyAdmin memiliki berbagai fungsi esensial untuk administrasi database MySQL, seperti:

1. *Manajemen Basis Data*

phpMyAdmin membantu pengguna dalam mengelola struktur basis data seperti tabel, kolom, dan indeks melalui antarmuka grafis yang user-friendly. phpMyAdmin juga menyediakan antarmuka yang intuitif untuk membuat, mengedit, dan menghapus basis data MySQL.

2. *Kueri SQL*

Salah satu keunggulan utama phpMyAdmin adalah kemampuan untuk mengeksekusi SQL secara langsung melalui antarmuka web. Pengguna dapat menulis, menguji, dan menjalankan perintah SQL tanpa harus bergantung pada perintah terminal maupun aplikasi khusus.

3. Impor dan Ekspor Data

phpMyAdmin memungkinkan pengguna untuk mengimpor data dalam basis data dari berbagai format file, seperti CSV atau SQL. Selain itu, pengguna juga dapat mengekspor isi basis data untuk keperluan backup dan memindahkan data ke server lain dengan mudah.

4. Manajemen Pengguna

Pengguna dapat mengelola pengguna dan izin mereka dengan sangat terperinci melalui phpMyAdmin. Ini termasuk memberikan hak akses spesifik untuk setiap pengguna serta memastikan bahwa data sensitif tetap aman dari akses yang tidak sah.

5. Manajemen Hubungan

phpMyAdmin menyediakan fitur untuk mendefinisikan dan mengelola hubungan antar tabel secara visual dan memudahkan pengguna dalam menentukan kunci asing, menjalin relasi antar tabel, serta memastikan konsistensi data dalam aplikasi mereka.

6. Pemantauan Status Server

phpMyAdmin menawarkan kemampuan pemantauan status dan kinerja server MySQL sehingga pengguna dapat melihat variabel server, status koneksi, serta memeriksa penggunaan daya seperti CPU dan RAM.

7. Fitur Keamanan

Pengaturan keamanan merupakan hal utama dalam phpMyAdmin. Pengguna dapat mengonfigurasi pengaturan keamanan seperti pengaturan kata sandi yang kuat, pengaturan izin akses, serta menerapkan enkripsi SSL untuk koneksi basis data yang aman.

8. Dukungan Multi-bahasa

phpMyAdmin memungkinkan pengguna dari seluruh dunia untuk mengelola basis data dalam bahasa mereka sendiri sehingga mampu meningkatkan aksesibilitas global dan mempermudah pengguna aplikasi di lingkungan multinasional atau multibahasa.

9. Penyesuaian dan Perluasan

Pengguna dapat menyesuaikan antarmuka phpMyAdmin dengan menambahkan plugin atau mengubah tema sesuai kebutuhan mereka. Fitur ini memungkinkan adanya pengembang untuk menyesuaikan pengalaman pengguna sesuai dengan kebutuhan spesifik aplikasi atau lingkungan kerja mereka.

10. Dokumentasi dan Dukungan

phpMyAdmin menyediakan dokumentasi lengkap dan panduan yang mendetail. Selain itu, pengguna juga dapat memanfaatkan forum komunitas untuk berbagi pengalaman, mendapatkan dukungan dari sesama pengguna, dan mencari solusi untuk masalah yang lebih kompleks.

Cara Kerja phpMyAdmin adalah alat administrasi yang menggunakan skrip PHP untuk menghubungkan antarmuka pengguna ke server MySQL. Ketika pengguna mengakses phpMyAdmin melalui browser, aplikasi ini akan mengirimkan permintaan HTTP ke server web tempat phpMyAdmin diinstal.

Selanjutnya server web meneruskan permintaan ini ke server MySQL yang telah dikonfigurasi sebelumnya untuk menampilkan informasi database secara langsung kepada pengguna.

Cara Install phpMyAdmin dapat dilakukan pada berbagai platform. Berikut adalah langkah-langkah untuk menginstal phpMyAdmin di Windows, Linux, dan MacOS:

- *Windows*
 - *Unduh phpMyAdmin dari situs resminya.*
 - *Ekstrak file yang diunduh.*
 - *Konfigurasi file 'config.inc.php'.*
 - *Akses phpMyAdmin melalui browser.*
- *Linux (Ubuntu)*

- Jalankan *'sudo apt update'*.
 - Install *phpMyAdmin* dengan *'sudo apt install phpmyadmin'*.
 - Konfigurasi Apache untuk *phpMyAdmin*.
 - Akses *phpMyAdmin* melalui browser.
-
- MacOS
 - Instal Homebrew jika belum ada.
 - Jalankan *'brew install phpmyadmin'*.
 - Konfigurasi file *'config.inc.php'*.
 - Akses *phpMyAdmin* melalui browser.

Cara Mengembalikan Database yang Terhapus di phpMyAdmin



Gambar 2.2 Database pada *phpmyadmin*

Jika database di phpMyAdmin tidak sengaja terhapus, ada beberapa langkah yang dapat dilakukan untuk memulihkannya, yaitu sebagai berikut:

- 1. Periksa apakah Anda memiliki file cadangan database yang dibuat sebelum perubahan besar. Apabila anda memiliki file cadangan database, buka phpMyAdmin, pilih opsi “import” dari menu utama, lalu unggah file cadangan tersebut.*
- 2. Lakukan pemulihan data dari log transaksi MySQL dengan menggunakan perintah ‘mysqlbinlog’. Langkah ini memerlukan akses ke server dan pemahaman dasar tentang perintah MySQL.*
- 3. Hubungi penyedia hosting Anda dan minta bantuan untuk memulihkan database melalui sistem cadangan mereka.*

Langkah-langkah tersebut memiliki peluang besar untuk memulihkan database yang terhapus dan memastikan integritas data anda tetap terjaga. Selalu pastikan untuk membuat cadangan secara rutin untuk menghindari kehilangan data di masa mendatang.

Selain itu, gunakan fitur dan perangkat lunak untuk memantau kondisi sistem database dan mendeteksi masalah potensial sebelum menjadi masalah yang lebih besar kedepannya.

Mengatasi Error di phpMyAdmin



Gambar 2.2 Ilustrasi error pada phpMyAdmin

Pengguna mungkin akan menghadapi beberapa error umum pada phpMyAdmin, seperti masalah koneksi, izin akses yang tidak tepat, atau bahkan masalah konfigurasi server. Berikut adalah beberapa langkah untuk menangani beberapa error umum tersebut:

1. Kesalahan Koneksi Database

Kesalahan ini terjadi karena phpMyAdmin gagal terhubung ke server MySQL atau MariaDB. Hal tersebut dapat ditangani dengan cara memastikan file konfigurasi 'config.inc.php' telah dikonfigurasi dengan benar, memeriksa kembali nama host, username, password, dan database yang digunakan. Selain itu, pastikan server database berjalan dan dapat diakses dari jaringan Anda.

2. Kesalahan Hak Akses

Kesalahan hak akses dapat diselesaikan dengan cara memastikan pengguna memiliki izin yang cukup (GRANT) untuk melakukan operasi yang diinginkan. Setelah itu, verifikasi hak akses dalam pengaturan MySQL maupun

MariaDB. Jika menggunakan kontrol akses berbasis peran, pastikan pengguna telah diberi peran yang sesuai.

3. Kesalahan versi PHP

phpMyAdmin memerlukan versi PHP yang compatible untuk berjalan dengan baik. Jika terjadi permasalahan pada versi PHP, lakukan pemeriksaan persyaratan versi PHP yang disarankan oleh phpMyAdmin dan pastikan telah diinstal dan dikonfigurasi dengan benar di server anda.

4. Kesalahan SQL

Error yang terjadi pada phpMyAdmin mungkin saja terjadi karena kesalahan perintah SQL melalui phpMyAdmin. Untuk itu, perlu adanya pemeriksaan sintaks SQL yang digunakan untuk memastikan tidak ada kesalahan penulisan atau logika. Gunakan alat bantu SQL error log atau pesan error yang dihasilkan oleh phpMyAdmin untuk memecahkan masalah.

5. Kesalahan Pengaturan Server Web

Masalah terkait konfigurasi server web yang mempengaruhi kinerja phpMyAdmin dapat diselesaikan dengan melakukan pemeriksaan konfigurasi Apache atau Nginx untuk memastikan konfigurasi PHP dan pengaturan akses telah diatur dengan benar. Pastikan modul PHP diaktifkan dan berjalan dengan baik di server web anda.

Dengan mengetahui jenis-jenis error yang umum terjadi pada phpMyAdmin beserta solusinya, Anda dapat mengatasi masalah yang mungkin timbul saat

menggunakan phpMyAdmin. Pastikan untuk selalu memantau log error dan melakukan backup data secara teratur untuk menghindari kehilangan data.

Alternatif phpMyAdmin

Terdapat beberapa alternatif alat administrasi selain phpMyAdmin yang dapat dipertimbangkan untuk administrasi database MySQL dan MariaDB. Berikut adalah beberapa pilihan alternatif yang populer:

- *Adminer*
- *HeidiSQL*
- *DBeaver*
- *Navicat*

Pemilihan alternatif alat administrasi lain tergantung pada apa yang diperlukan pengguna, platform yang digunakan, dan preferensi antarmuka pengguna. Adminer cocok bagi pengguna yang membutuhkan alat ringan dengan fungsionalitas dasar.

HeidiSQL lebih cocok untuk pengguna windows yang memerlukan antarmuka visual yang kuat. Sedangkan DBeaver dan Navicat cocok untuk pengguna yang mencari dukungan multi-platform dan fitur canggih untuk manajemen database dalam skala besar.

10. Dasar Pemrograman

Menurut Ade Hastuty Hasyim (2021). Program adalah kumpulan pernyataan komputer, sedangkan metode dan tahapan sistematis dalam program adalah *algoritma*. Program ditulis dengan menggunakan bahasa pemrograman. Jadi bisa disebut bahwa program adalah suatu implementasi dari bahasa pemrograman.

Wirth (1997) menyatakan dalam bukunya bahwa: ***Algoritma* + Struktur Data = Program**

Bagaimanapun juga struktur data dan *algoritma* berhubungan sangat erat pada sebuah program. *Algoritma* yang baik tanpa pemilihan struktur data yang tepat akan membuat program menjadi kurang baik, Demikian juga sebaliknya. Dalam pembuatan *algoritma* mempunyai banyak keuntungan di antaranya:

- Pembuatan atau penulisan *algoritma* tidak tergantung pada bahasa pemrograman manapun, artinya penulisan *algoritma* independen dari bahasa pemrograman dan komputer yang melaksanakannya.
- Notasi *algoritma* dapat diterjemahkan ke dalam berbagai bahasa pemrograman.
- Apapun bahasa pemrogramannya, output yang akan dikeluarkan sama karena *algoritmanya* sama.

Beberapa hal yang perlu diperhatikan dalam membuat *algoritma*:

- Teks *algoritma* berisi deskripsi langkah-langkah penyelesaian masalah. Deskripsi tersebut dapat ditulis dalam notasi apa- pun asalkan mudah dimengerti dan dipahami.

- Tidak ada notasi yang baku dalam penulisan teks *algoritma* seperti notasi bahasa pemrograman. Notasi yang digunakan dalam menulis *algoritma* disebut notasi *algoritmik*.
- Setiap orang dapat membuat aturan penulisan dan notasi *algoritmik* sendiri. Hal ini dikarenakan teks *algoritma* tidak sama dengan teks program. Namun, supaya notasi *algoritmik* mudah ditranslasikan ke dalam notasi bahasa pemrograman tertentu, maka sebaiknya notasi *algoritmik* tersebut berkorespondensi dengan notasi bahasa pemrograman secara umum.
- Notasi *algoritmik* bukan notasi bahasa pemrograman, karena itu *pseudocode* dalam notasi *algoritmik* tidak dapat dijalankan oleh komputer. Agar dapat dijalankan oleh komputer, *pseudocode* dalam notasi algoritmik harus ditranslasikan atau diterjemahkan ke dalam notasi bahasa pemrograman yang dipilih. Perlu diingat bahwa orang yang menulis program sangat terikat dalam aturan tata bahasanya dan spesifikasi mesin yang menjalannya.
- *Algoritma* sebenarnya digunakan untuk membantu kita dalam mengkonversikan suatu permasalahan ke mejep esueq pemrogram.
- *Algoritma* merupakan hasil pemikiran konseptual, supaya dapat dilaksanakan oleh komputer, *algoritma* harus ditranslasikan ke dalam notasi bahasa pemrograman.

11. Algoritma

Menurut Ade Hastuty Hasyim (2021). "*Algoritma* adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis". Kata logis merupakan kata kunci dalam *algoritma*. Langkah-langkah dalam *algoritma* harus logis dan harus dapat ditentukan bernilai salah atau benar. Dalam beberapa konteks, *algoritma* adalah spesifikasi urutan langkah untuk melakukan pekerjaan tertentu. Pertimbangan dalam pemilihan *algoritma* adalah, pertama, *algoritma* haruslah benar. Artinya *algoritma* akan memberikan keluaran yang dikehendaki dari sejumlah masukan yang diberikan.

Tidak peduli sebegus apapun *algoritma*, kalau mnemberikan keluaran yang salah, pastilah *algoritma* tersebut bukanlah *algoritma* yang baik. Pertimbangan kedua yang harus diperhatikan adalah kita harus mengetahui seberapa baik hasil yang dicapai oleh *algoritma* tersebut. Hal ini penting terutama pada *algoritma* untuk menyelesaikan masalah yang memerlukan *aproksimasi* hasil (hasil yang hanya berupa pendekatan). *Algoritma* yang baik harus mampu memberikan hasil yang sedekat mungkin dengan nilai yang sebenarnya. Ketiga adalah efisiensi *algoritma*. Efisiensi *algoritma* dapat ditinjau dari 2 hal yaitu efisiensi waktu dan memori. Meskipun *algoritma* memberikan keluaran yang benar (paling mendekati), tetapi jika kita harus menunggu berjam-jam untuk mendapatkan keluarannya, *algoritma* tersebut biasanya tidak akan dipakai, setiap orang menginginkan keluaran yang cepat. Begitu juga dengan memori, semakin besar memori yang terpakai maka semakin buruklah *algoritma* tersebut. Dalam kenyataannya, setiap orang bisa membuat *algoritma* yang berbeda untuk menyelesaikan suatu

permasalahan, walaupun terjadi perbedaan dalam menyusun *algoritma*, tentunya kita mengharapkan keluaran yang sama. Jika terjadi demikian, carilah *algoritma* yang paling efisien dan cepat.

C. Batasan Masalah

Penelitian ini memiliki beberapa batasan masalah dengan harapan penelitian terfokus dengan batasan-batasan yang dibuat. Adapun batasan dalam penelitian ini yaitu;

1. Penelitian dilakukan terfokus pada panti asuhan abadi aisyiyah kota parepare,
2. Data anak yang dikategorikan yaitu anak yatim piatu, anak yatim dan anak yang tidak mampu,
3. Dana sumbangan yang dikelola menjadi 2 jenis yaitu sumbangan berupa barang dan sumbangan uang.

D. Tujuan Penelitian

Adapun tujuan penelitian yaitu:

1. Untuk pengembangan aplikasi sebelumnya,
 2. Untuk mempermudah monitoring dana yang masuk dan keluar,
 3. Bisa di akses aplikasinya dimana saja selama masih ada internet,
- Mengenalkan biodata anak panti dan ketampilan anak panti.

E. Sistematika Penulisan

Sistematika penulisan skripsi ini dibagi ke dalam enam bab yang masing-masing menjelaskan bagian-bagian yang berbeda namun tetap memiliki korelasi satu dengan yang lain. Setiap bab-nya terdiri dari beberapa sub tersendiri. Diawali dengan bab pendahuluan dan di akhiri dengan bab penutup yang berisi kesimpulan dan saran-saran. Secara singkat akan diuraikan sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini akan dijelaskan tentang latar belakang penulisan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Bab ini berisi uraian tentang landasan teori-teori dari perancangan **“Pengembangan Aplikasi Dana Sosial Pada Panti Asuhan Berbasis Android”**. Bab ini juga membahas mengenai berbagai teori yang mendasari dari analisis permasalahan yang berhubungan dengan topik yang dibahas.

BAB III : METODOLOGI PENELITIAN

Dalam bab ini membahas mengenai metode penelitian yang digunakan dalam merancang dan membangun sistem. Pada bab ini juga membahas tentang identifikasi masalah, analisis kebutuhan aplikasi yang meliputi kebutuhan masukan, proses dan keluaran aplikasi yang akan dibangun.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini membahas mengenai hasil dari tahap analisis dan perancangan sesuai dengan metode yang digunakan pada alat dan sistem yang dibuat serta hasil dari pengujian.

BAB V : PENUTUP

Pada bab akhir ini berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran yang diberikan untuk pengembangan lebih lanjut agar tercapai hasil yang lebih maksimal.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

B. Penelitian Terdahulu

Berbagai penelitian sebelumnya merupakan hal yang sangat perlu dan dapat dijadikan sebagai data pendukung. Salah satu data pendukung yang menurut peneliti perlu dijadikan bagian tersendiri adalah penelitian terdahulu yang relevan dengan permasalahan yang sedang dibahas dalam penelitian ini. Oleh karena itu, peneliti melakukan kajian terhadap beberapa hasil penelitian berupa tugas akhir/skripsi atau jurnal melalui *internet*.

1. Nur azizah amelia (2014) Universitas Muhammadiyah Parepare. Melakukan penelitian dengan judul “Aplikasi pengelolah data donatur dan dana sumbangan pada panti asuhan” ini di tunjukan kepada pengurus untuk meningkatkan kinerja dan pengelolah data yang ada Aplikasi ini dibuat di buat menggunakan bahasa pemrograman *Visual Basic (VB)* dengan *MySQL* sebagai database dan menggunakan data report dalam pembuatan laporannya. Hasil yang diharapkan dalam penelitian ini adalah mempermudah kinerja pengurus panti asuhan dalam mengolah data-data donatur dan dana sumbangan serta mempermudah dalam pembuatan laporan (Amelia, 2014).
2. Sampurna Dadi Riskiono, Fikri Hamidy dan Tami Ulfia (2020) Universitas Teknologi Indonesia dengan judul Penelitian “Sistem Informasi Manajemen Dana Donatur Berbasis Web Pada Panti Asuhan Yatim Madani” Panti asuhan

yatim madani belum memiliki sistem informasi yang terinterasi untuk mendukung proses pengelolah data dan bantuannya. Aplikasi ini menggunakan *crowdfunding*, dengan model situs donasi penyebaran informasi donatur melalui media sosial shingga mempermudah untuk membuat proyek donasi mempublikasikan dana yang dibutuhkan (Sampurna Dadi Rikiono, 2020)

3. Jajan Romansyah, Asri Mulyani, Dan Dede Kurniadi (2019) Sekolah Tinggi Teknologi Garut dengan judul penelitian “Rancang Bangun Sistem Informasi Penggalangan Dana Sosial Panti Asuhan Berbasis *Web*” penelitian ini mengambil studi kasus panti asuhan di kota Garut. Ketikan melukan *survey* ke lapangan, masih banyak panti asuhan tidak memiliki website sebagai sarana penggalangan dana berbasis *web*. Ketika proses perencanaan menggunakan metode *waterfall* dengan secara terstruktur (Jajang Romansyah, 2019).
4. Yanuar Aditama (2018) Universitas Muhammadiyah Sukarta dengan judul penelitian “perancangan *Website* Sistem Inforamasi Panti Asuhan Yatim Piatu Al Huda” selama ini hanya menggunakan facebook dan sosial media lainnya untuk memeberikan indormasi, sehingga di butuhkan *website* untuk informasi yang lebih *valid* untuk masyarakat. Proses pembuatan *web* menggunakan *Hypertext Preprecesor* (PHP) dengan *framework CodeIngniter*.
5. Sistem informasi panti asuhan inimemiliki fitur untuk mengelolah data panri asuhan serta donatur dapat melakukan donasi. Dari Sistem yang di buat dan diuji menggunakan metode *Blackbox* dan pengujian *user* dari respoden menyatakan sistem ini bermanfaat dan dapat diterima pengguna melalui indikasi rata-rata 83%. Dengan sistem ini proses penyebaran data menjadi cepat, mudah dan menarik minat donatur (Aditama, 2018).

C. Tahap-tahap Perancangan Sistem

1. Use Case

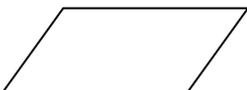
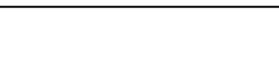
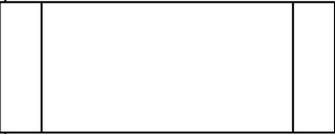
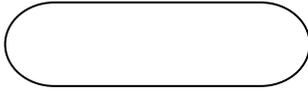
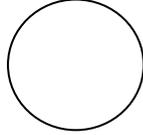
Use case diagram adalah abstraksi dari interaksi antara sistem dan *actor*.

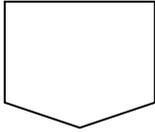
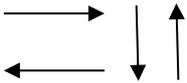
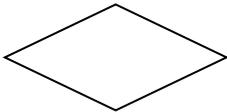
Use case bekerja mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

2. Bagan Alir Program

Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derifikasi bagan alir system. Bagan alir program dibuat dengan menggunakan simbol-simbol sebagai berikut :

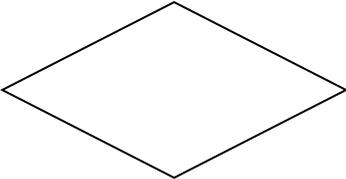
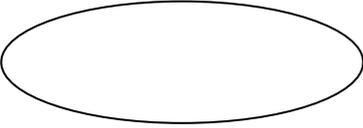
Tabel 2.1 Simbol *Flowchart*

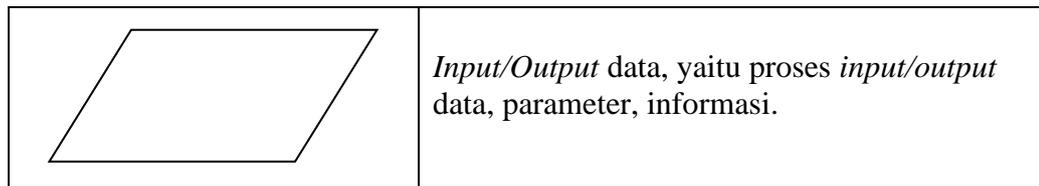
Simbol	Keterangan
	Simbol <i>input/output</i> digunakan untuk mewakili data <i>input/output</i> .
	Simbol proses, digunakan untuk mewakili suatu proses.
	Simbol proses terdefinisi, digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain.
	Simbol titik terminal, digunakan untuk awal dan akhir suatu proses.
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus di halaman yang masih sama.

Simbol	Keterangan
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus di halaman yang berbeda.
	Simbol garis alir, menunjukkan arus dari proses.
	Simbol keputusan, digunakan untuk suatu penyelesaian kondisi di dalam program .

3. Entity Relationship Diagram

Tabel 2.2 Simbol *Entity Relationship Diagram*

NOTASI	KETERANGAN
	Entitas, yaitu kumpulan dari objek yang diidentifikasi secara unik.
	Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain: satu ke satu, satu ke banyak dan banyak ke banyak.
	Atribut, yaitu karakteristik dari <i>entity</i> atau relasi yang merupakan penjelasan detail tentang entitas.
	Garis, hubungan antara <i>entity</i> dengan atributnya dan himpunan entitas dengan himpunan relasi.



Entity Relationship Diagram (ERD) adalah adalah suatu pemodelan konseptual yang didesain secara khusus untuk mengidentifikasi entitas yang menjelaskan data dan hubungan antar data. *ERD* merupakan suatu model untuk menjelaskan hubungan antardata dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *ERD* untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga komponen yang digunakan, yaitu :

- a. *Entity* merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol-simbol dari *entity* ini biasanya digambarkan dengan persegi panjang.
- b. Atribut Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol *elips*.
- c. Hubungan / Relasi ialah hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

Relasi dapat digambarkan sebagai berikut : Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

1) Satu ke satu (*One to one*)

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

2) Satu ke banyak (*One to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3) Banyak ke banyak (*Many to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B. Mempunyai pengertian “Satu baris atau lebih data pada tabel pertama bisa dihubungkan ke satu atau lebih baris data pada tabel ke dua.

4. UML (Unified Modelling Language)

UML (*Unified Modelling Language*) menurut Munawar dalam buku “Pemodelan Visual dengan UML” (2005 : 17) menyatakan UML adalah salah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Sedangkan menurut menurut Adi nugroho (2010:6), “*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek”. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Jurnal CCIT (2008:70) *Unified Modelling Language* (UML) didefinisikan oleh *Good Corporate* sebagai berikut :

Unified Modelling Language (UML) adalah “suatu alat bantu yang sangat handal dalam bidang pengembangan yang berorientasi objek. Hal ini disebabkan karena uml menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem untuk membuat cetak cetak biru atas visi mereka dalam bentuk baku”. *Metode Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu:

a. Sesuatu (*things*)

Ada empat *things* dalam *Unified Modelling Language* (UML) :

1. *Structural things*, bagian yang relative statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.
2. *Behaviorial things*, bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang ruang dan waktu.
3. *Grouping things*, bagian pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat

didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta *subsistem-subsistem*.

4. *An notational things*, merupakan bagian yang memperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap element dalam model UML.

b. Relasi (*relationship*)

Untuk upaya tersebut UML menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya *statis* atau *dinamis*. Ke 9 jenis diagram dalam UML itu adalah :

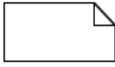
1. *Diagram kelas* Bersifat statis *Diagram* ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. *Diagram* ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula *diagram kelas* memuat kelas-kelas aktif.
2. *Diagram Objek* Bersifat statis. *Diagram* ini memperlihatkan objek-objek serta relasi-relasi antar objek. *Diagram objek* memperlihatkan instalasi statis dari segala sesuatu yang dijumpai pada *diagram kelas*.
3. *Use-Case Diagram* Bersifat statis. *Diagram* ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). *Diagram* ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. *Sequence Diagram* Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. *Collaboration Diagram* Bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
6. *Statechart Diagram* Bersifat dinamis. Diagram state ini memperlihatkan *state-state* pada sistem memuat *state*, *transisi*, *event*, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. *Activity Diagram* Bersifat dinamis. *Diagram* aktivitas ini adalah tipe khusus dari *diagram state* yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.
8. *Component Diagram* Bersifat statis. Dengan komponen ini memperlihatkan organisasi serta kebergantungan sistem atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya. *Diagram* ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.

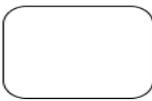
9. *Deployment Diagram* Bersifat statis. *Diagram* ini memperlihatkan konfigurasi saat aplikasi dijalankan. *Diagram* ini membuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. *Diagram* ini sangat berguna pada banyak mesin. (Sholiq, 2006)

Tabel 2.3 *Use Case Diagram*

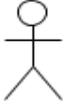
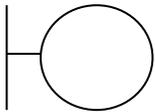
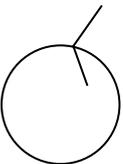
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung yang tidak mandiri (<i>independent</i>).
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek.
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan suatu paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan dalam sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .

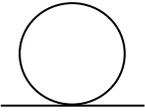
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumberdaya komputasi.

Tabel 2.4 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan cara bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State adalah suatu sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap-tahap tertentu berubah menjadi beberapa aliran.

Tabel 2.5 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Objekt (Partisipan)</i>	<i>Object</i> atau biasa juga disebut partisipan merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun dari secara <i>horizontal</i> . Digambarkan ini sebagai sebuah <i>class</i> (kotak) dengan nama objek di dalamnya yang diawali dengan sebuah titik koma.
2		<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.
3		<i>Life line</i>	<i>Lifeline</i> ini mengidentifikasi keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>lifeline</i> adalah garis putus-putus <i>vertical</i> yang di tarik oleh sebuah <i>object</i> .
4		<i>Collaborations</i>	<i>Activation</i> dinotasikan sebagai kotak segi empat yang digambar pada sebuah <i>lifeline action</i> mengidentifikasi dari sebuah <i>object</i> yang akan melakukan sebuah aksi.
5		<i>Boundary</i>	<i>Boundary</i> terletak dari diantara <i>system</i> dengan tema dunia di sekelilingnya. Semua dari <i>form</i> , laporan-laporan, antarmuka ke perangkat keras seperti <i>printer</i> atau <i>scanner</i> dan antara muka ke <i>system</i> lainnya adalah termasuk dalam kategori.
6		<i>Control</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

7		<i>Entity</i>	<i>Entity</i> ini digunakan menengani informasi yang mungkinakan disimpan secara suatu permanen. <i>Entity</i> ini biasa juga merupakan sebuah <i>table</i> .
---	---	---------------	---

D. Metode Pengujian Perangkat Lunak

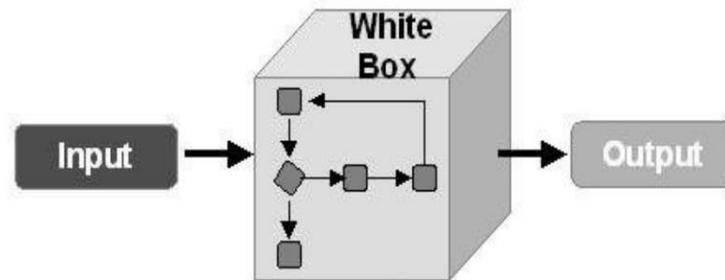
Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, *query*, atau jalur navigasi yang mewakili penggunaan sistem yang *typical*, kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit testing, yaitu :

1. *White Box Testing*

Uji coba *white box testing* merupakan metode perancangan *testcase* yang menggunakan struktural untuk mendapatkan *testcase*, *test* ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika), perangkat lunak di *test* dengan kondisi dan perulangan secara fisik. Contoh pengujian *white box testing* ini merupakan peringatan ketika *user* menginputkan

password user yang salah, untuk kesalahan semacam ini akan memberikan suatu informasi kepada *user* mengenai kesalahan yang dilakukan.



Gambar 2.3 Pengujian *White Box*

Ciri – ciri pengujian *white box testing*:

- a. Pengujian berbasis pengetahuan. Penguji mengetahui cara kerja internal sistem dan menggunakan informasi ini untuk membuat tes yang tepat.
- b. Fokus pada struktur. Metode ini menguji arsitektur, aliran data, jalur eksekusi, dan logika kode yang digunakan dalam program.
- c. Deteksi bug mendalam. Dapat menemukan bug yang tersembunyi dalam kode, seperti kesalahan logika atau cacat dalam aliran kontrol program.
- d. Alat dan teknik. Teknik yang umum digunakan dalam *whitebox testing* termasuk *path testing*, *control flow testing*, *data flow testing*, dan *branch testing*.

Berikut adalah langkah – langkah umum dalam pengujian *white box testing*:

- a. Analisis kode sumber. Pelajari kode sumber aplikasi untuk memahami struktur internal dan alur logika. Identifikasi fungsi, alur kontrol, dan alur data.
- b. Identifikasi kasus uji. Buat kasus uji yang mencakup semua jalur dan kondisi yang diidentifikasi. Pastikan kasus uji dirancang untuk menguji semua cabang dan jalur eksekusi yang mungkin terjadi.
- c. Buat skripsi uji. Kembangkan skrip uji atau prosedur uji yang diperlukan untuk menjalankan kasus uji. Skrip ini akan menjalankan pengujian sesuai dengan desain yang telah dibuat.
- d. Jalankan kasus uji. Eksekusi kasus uji dengan menggunakan skrip uji yang telah dibuat. Pastikan untuk memeriksa hasil dan bandingkan dengan hasil yang diharapkan.
- e. Tulis laporan hasil uji. Dokumentasikan hasil uji, termasuk hasil yang berhasil dan masalah yang ditemukan. Laporkan masalah yang memerlukan perhatian lebih lanjut.
- f. Perbaiki dan uji ulang. Jika ditemukan masalah, koordinasikan dengan pengembang untuk memperbaikinya. Setelah perbaikan dilakukan, jalankan uji ulang untuk memastikan bahwa masalah telah teratasi dan tidak ada masalah baru yang muncul.

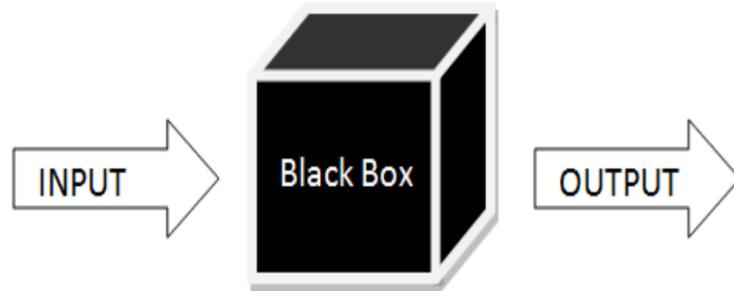
- g. Lakukan pengujian tambahan. Jika perlu, lakukan pengujian tambahan untuk memastikan bahwa perubahan atau perbaikan tidak memengaruhi bagian lain dari aplikasi.
- h. Evaluasi dan penutupan. Setelah semua pengujian selesai, evaluasi keseluruhan hasil pengujian dan pastikan semua kasus uji telah berhasil. Dokumentasikan proses pengujian secara menyeluruh dan tutup fase pengujian.

Contoh penggunaan *white box testing* sebagai berikut:

- a. Menguji bagaimana variabel diproses dalam suatu program
- b. Memeriksa jalur logika dalam algoritma, seperti memastikan semua cabang *if – else* dieksekusi.
- c. Digunakan dalam pengujian unit (*unit testing*), dimana pengembang memeriksa bagian kecil dari kode, seperti fungsi atau modul.

2. Black Box Testing

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitam nya. Sama seperti pengujian *black box*, mengevaluasi hanya dari tampilan luarnya (*interfacenya*), fungsionalitas tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detilnya (hanya mengetahui *input* dan *output*).



Gambar 2.4 Pengujian *Black Box*

Ciri – ciri pengujian *blackbox testing*:

- a. Pendekatan berbasis fungsional. Pengujian dilakukan berdasarkan fungsionalitas eksternal sistem tanpa memperhatikan bagaimana sistem tersebut diprogram.
- b. Fokus pada hasil. Penguji memvalidasi bahwa sistem bekerja sesuai dengan yang diharapkan berdasarkan input yang diberikan dan keluaran yang dihasilkan.
- c. Pengujian pengguna akhir. *Blackbox testing* sering kali mirip dengan cara pengguna akhir akan menggunakan perangkat lunak.
- d. Alat dan teknik. Teknik umum termasuk *equivalence partitioning*, *boundary value analysis* dan *decision table testing*.

Berikut adalah langkah – langkah umum dalam pengujian *blackbox testing* :

- a. Memahami spesifikasi dan persyaratan
 - 1) Pelajari dokumentasi spesifikasi fungsional, persyaratan sistem, dan dokumen desain untuk memahami apa yang diharapkan dari sistem.

2) Identifikasi fungsionalitas utama, alur kerja, dan kebutuhan pengguna.

b. Perencanaan pengujian

1) Buat rencana pengujian yang mencakup tujuan pengujian, ruang lingkup, sumber daya yang diperlukan, dan jadwal.

2) Tentukan jenis pengujian *blackbox* yang akan dilakukan, seperti pengujian fungsional, pengujian batas, pengujian komponen, dll.

c. Desain kasus uji

1) Buat kasus uji berdasarkan spesifikasi fungsional dan persyaratan sistem. Pastikan untuk mencakup berbagai skenario pengguna, input yang valid, dan input yang tidak valid.

2) Rancang kasus uji untuk menguji fungsi-fungsi sistem, alur kerja, dan interaksi antar komponen.

d. Siapkan lingkungan pengujian

1) Siapkan lingkungan pengujian yang mencakup perangkat keras, perangkat lunak, dan konfigurasi yang diperlukan untuk menjalankan pengujian.

2) Pastikan semua data yang diperlukan dan konfigurasi sistem sesuai dengan spesifikasi pengujian.

e. Jalankan kasus uji

- 1) Eksekusi kasus uji yang telah dirancang. Masukkan data uji dan amati bagaimana sistem merespons.
- 2) Bandingkan hasil yang diperoleh dengan hasil yang diharapkan untuk menentukan apakah fungsionalitas sistem berfungsi dengan baik.

f. Dokumentasikan hasil uji

- 1) Catat hasil pengujian, termasuk kasus uji yang berhasil dan yang gagal.
- 2) Dokumentasi setiap masalah yang ditemukan, lengkap dengan deskripsi masalah, langkah-langkah untuk mereproduksi, dan hasil yang diharapkan versus yang diperoleh.

g. Tindak lanjut dan perbaikan

- 1) Laporkan masalah kepada tim pengembangan untuk diperbaiki.
- 2) Setelah perbaikan dilakukan, jalankan uji ulang untuk memastikan masalah telah teratasi dan tidak ada masalah baru.

h. Pengujian regresi

- 1) Lakukan pengujian regresi untuk memastikan bahwa perbaikan atau perubahan tidak memengaruhi bagian lain dari sistem yang sudah ada.

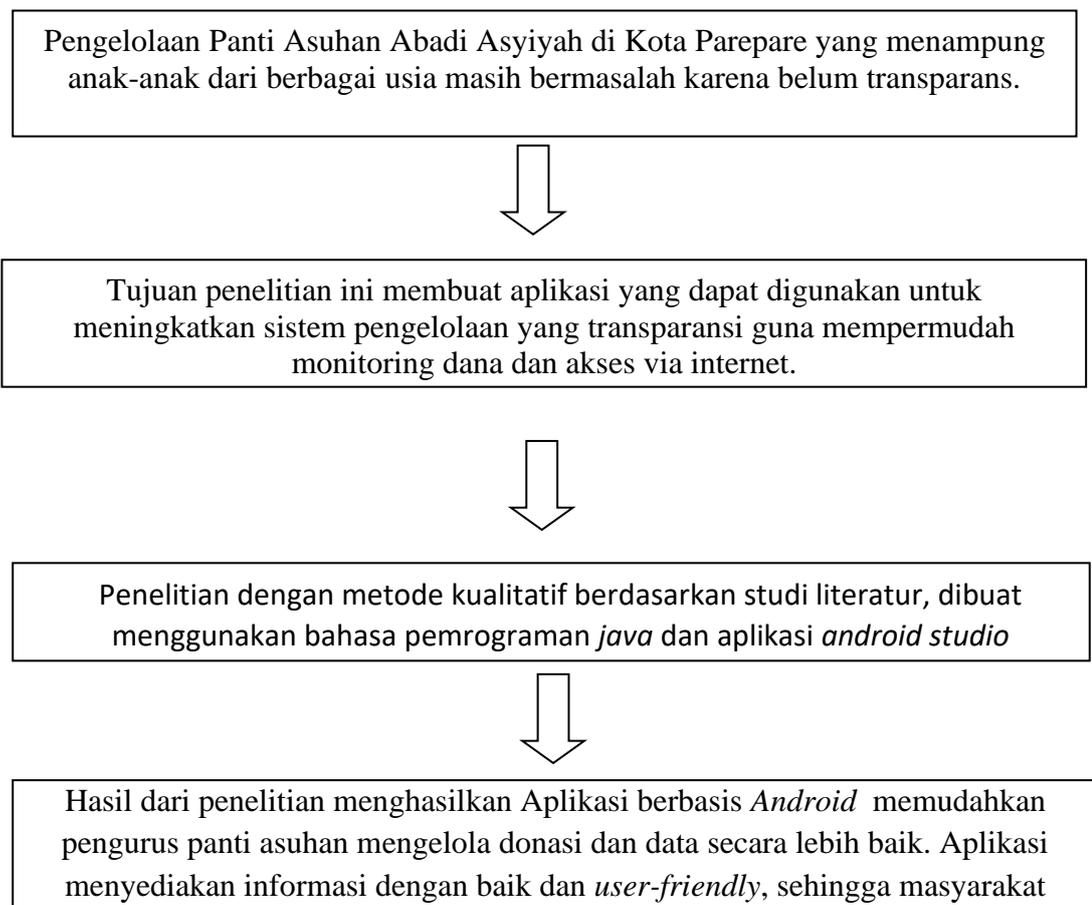
i. Evaluasi dan penutupan

- 1) Tinjau hasil pengujian secara keseluruhan dan pastikan semua kasus uji telah dieksekusi.

Dokumentasikan proses pengujian dan hasilnya secara menyeluruh. Siapkan laporan akhir yang mencakup ringkasan temuan dan rekomendasi.

E. Kerangka Pikir

Untuk lebih memperjelas kerangka berpikir yang akan ditulis dalam penelitian akhir ini. Maka berikut ini digambarkan kerangka pikir tersebut pada kerangka berfikir dibawah:



BAB III

METODOLOGI PENELITIAN

A. Tempat dan Waktu Penelitian

1. Penelitian yang dilakukan oleh penulis bertempat di Panti Asuhan Abadi Aisyiyah No.3 Parepare
2. Pelaksanaan penelitian selama (\pm) 2 bulan.

B. Metode Penelitian

Untuk membantu suatu kelancaran pengumpulan data, maka penulis menggunakan beberapa metode antara lain :

1. Metode Analisis

- a) Tinjauan Pustaka (*Literature Review*), dilakukan dengan cara mengumpulkan data-data secara teoritis yang berhubungan dengan masalah yang menjadi objek penelitian dari literatur-literatur serta buku-buku dari perpustakaan yang digunakan sebagai landasan teori.
- b) Pengamatan (*Observation*), proses mengumpulkan data-data yang diperlukan dengan mengadakan kunjungan secara langsung di Puskesmas Barukku sebagai pelayanan kesehatan tingkat I di Kabupaten Sidrap, mengamati langsung proses pencatatan rekam medis serta bertemu dengan pihak terkait, data-data yang diperoleh kemudian dituangkan dalam bentuk kata-kata maupun skema.

2. Wawancara (*Interview*)

Wawancara merupakan teknik pengumpulan data yang dilakukan melalui tatap muka dan tanya jawab langsung antara pengumpul data terhadap narasumber/sumber data. Metode ini dengan cara mengumpulkan data-data yang diperlukan dengan mengadakan wawancara secara langsung dengan Kepala Puskesmas, Biro Humas dan Tenaga Kesehatan serta pihak-pihak yang terkait.

3. Kajian Pustaka

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data analisis, perancangan, pengujian dan implementasi. Adapun Uraian dari tahapan tersebut adalah sebagai berikut :

a) Persiapan Penelitian

Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta *software* yang digunakan selama penelitian.

b) Studi Literature

Pada tahapan ini peneliti melakukan apa yang disebut dengan kajian pustaka, yaitu mempelajari buku-buku referensi dan hasil penelitian sejenis sebelumnya yang pernah dilakukan oleh orang lain. Tujuannya ialah untuk mendapatkan landasan teori mengenai masalah yang akan diteliti. Teori merupakan pijakan bagi peneliti untuk memahami persoalan yang diteliti dengan benar dan sesuai dengan kerangka berpikir ilmiah.

C. Metode Pengumpulan Data

Pada tahap ini peneliti melakukan wawancara dengan pihak yang terkait dengan penelitian, serta melakukan pencatatan dan pengamatan langsung di tempat penelitian.

1. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang di terapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

2. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

3. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

4. Implementasi

Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk digunakan oleh *user*.

D. Alat dan Bahan

Tahap desain dilakukan untuk menentukan ruang lingkup pembuatan aplikasi yang ditangani, yang selanjutnya dijabarkan dalam bentuk diagram alir. Dalam pembuatan program aplikasi tersebut penulis membutuhkan perangkat keras

(*Hardware*) dan perangkat lunak (*Software*). Adapun *Hardware* dan *Software* yang digunakan adalah :

1. Perangkat keras (*Hardware*)

Perangkat keras yang digunakan untuk membuat aplikasi menggunakan Laptop atau PC (*Personal Komputer*) dengan spesifikasi sebagai berikut :

- a. *Processor AMD A4-9120 Radeon R3, 4 Compute Cores 2C+2G Mhz, 2 Core{s}, 2 Logical*
- b. *Kapasitas Memory 4GB*
- c. *Kapasitas Hardisk 1 TB*

2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam pembuatan aplikasi adalah :

- a. *Sistem Operasi Windows 10*
- b. *PHP, Javascript Sebagai Bahasa Program*
- c. *MySql untuk Database*
- d. *Start UML untuk Rancangan Program*
- e. *Android Studio Software*

3. Bahan Penelitian

Berupa data-data yang langsung dari dalam bentuk observasi dan wawancara langsung di panti asuhan diambil yang berkaitan dengan penelitian ini.

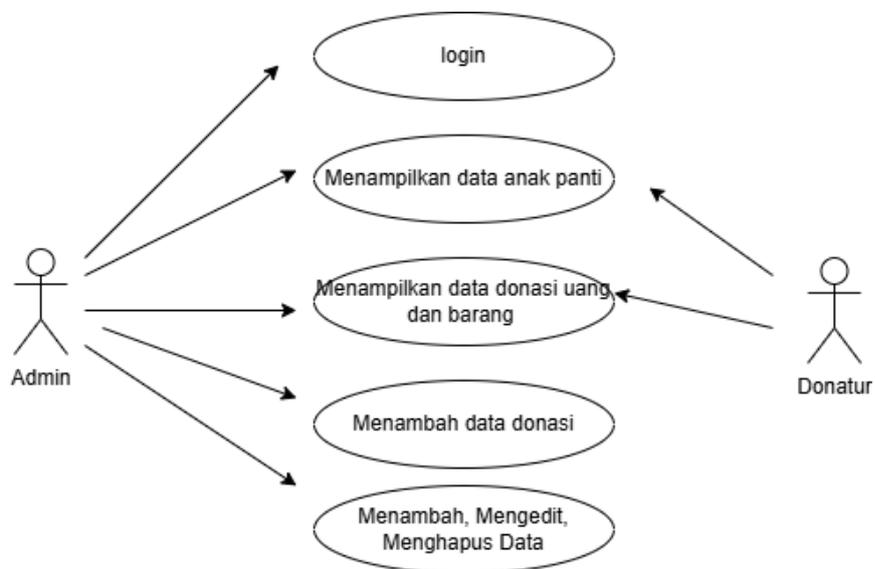
BAB IV

HASIL DAN PEMBAHASAN

A. Rancangan Sistem

1. Sistem Yang Diusulkan.

Sistem yang diusulkan ini menggunakan *Use Case Diagram*. *Use case diagram* menggambarkan interaksi antara *actor* dengan proses atau sistem yang dibuat. *Use case* dan *actor* menggambarkan ruang lingkup sistem yang sedang dibangun. *Use case* meliputi semua hal yang ada pada sistem, sedangkan *actor* meliputi semua hal yang ada diluar sistem. *Actor* termasuk seseorang atau apa saja yang berhubungan dengan sistem yang dibangun.



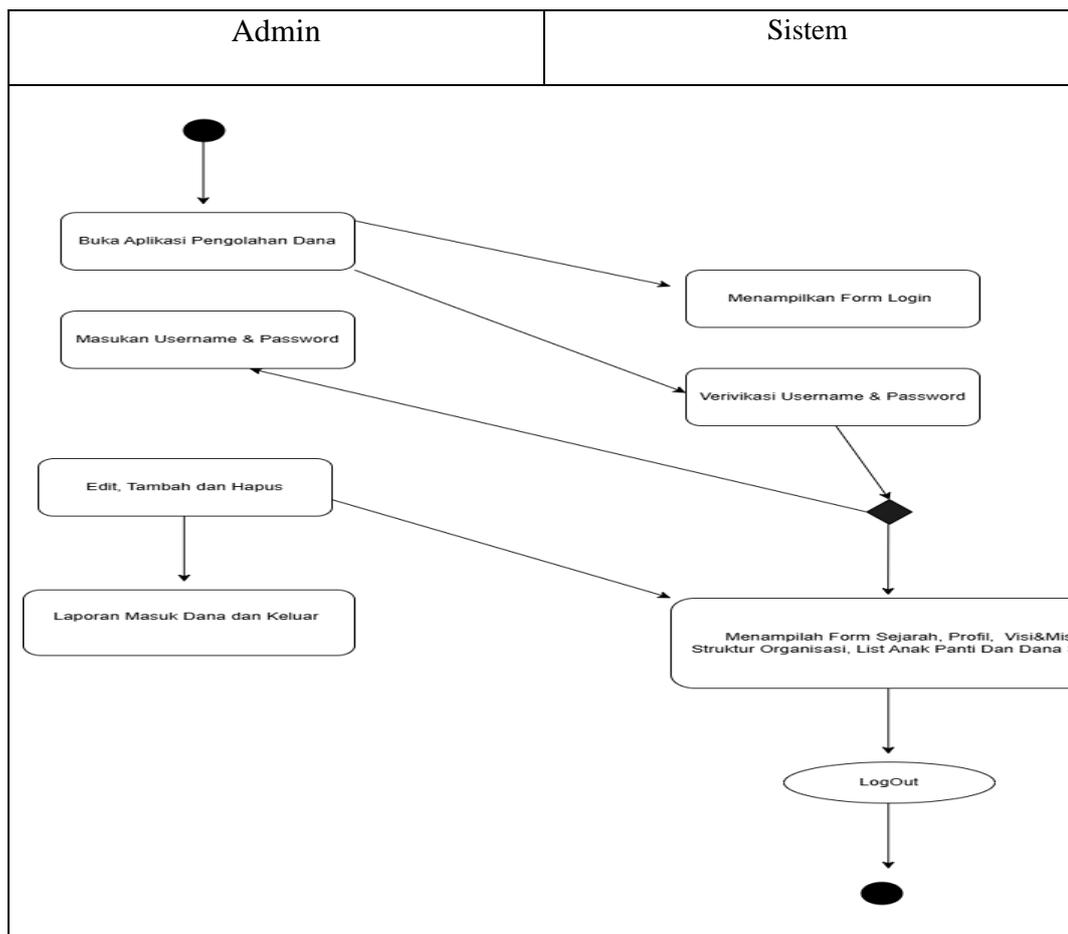
Gambar 4.3 Use Case

B. Analisis Aliran Data UML

Analisis aliran data yang diusulkan ini bertujuan mengetahui suatu proses informasi, dalam analisis sistem penulis menggunakan pengembangan orientasi objek sehingga menggunakan UML.

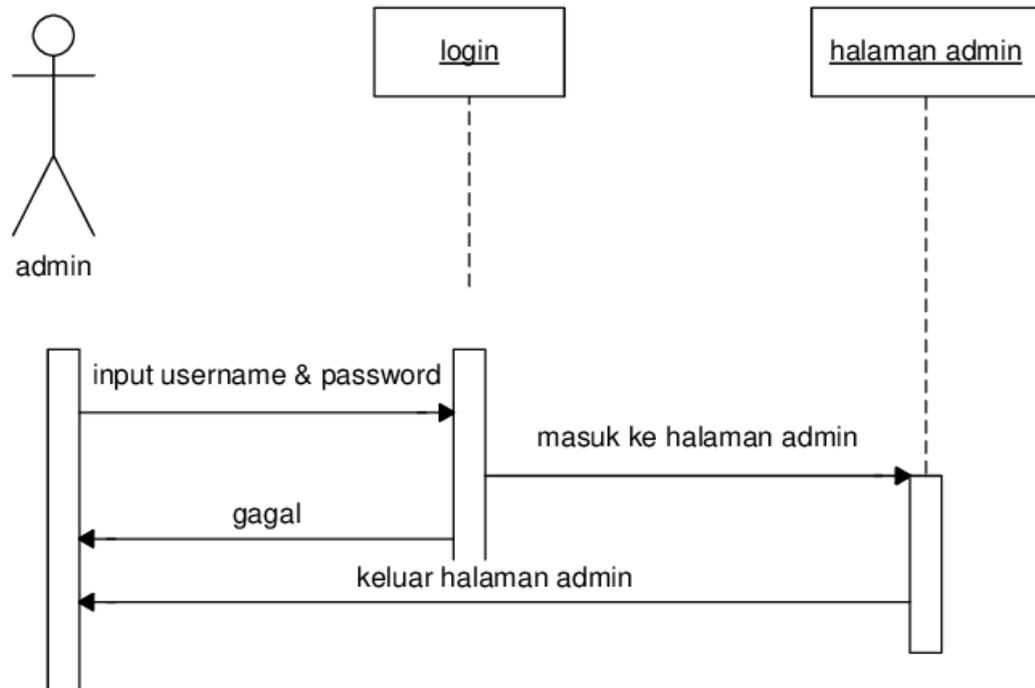
1. Activity Diagram

Activity diagram adalah adalah suatu bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak *Activity diagram* ini menjelaskan tentang aktifitas-aktifitas yang terjadi dalam sebuah aliran proses pada sebuah *system*. Sequence Diagram



Gambar 4.2 Activity diagram

2. Sequence Diagram Admin

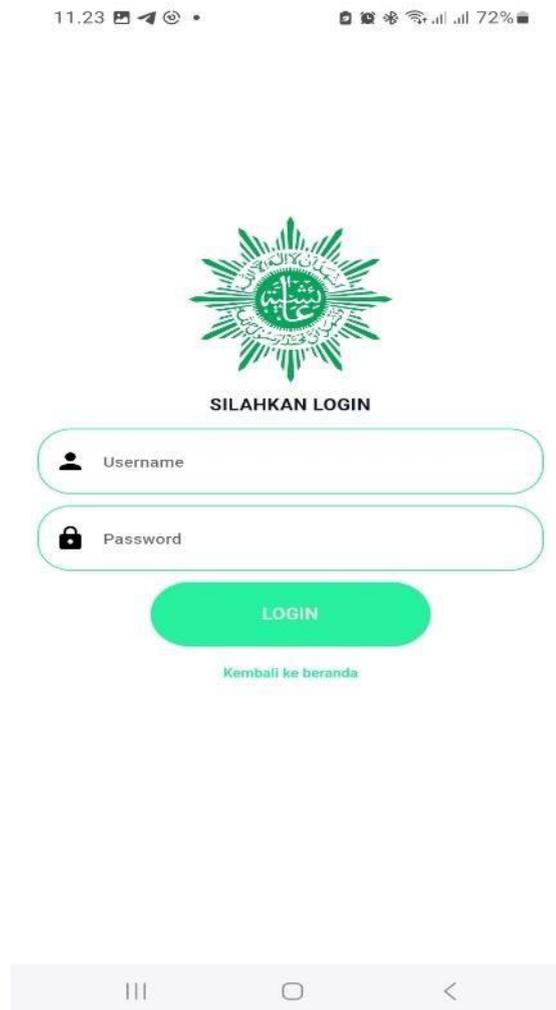


Gambar 4.3 Sequence Diagram Admin

C. Detail Aplikasi

1) Admin

a. Halaman Login



11.23 72%

SILAHKAN LOGIN

Username

Password

LOGIN

Kembali ke beranda

Gambar 4. 4 Login admin

Diatas merupakan tampilan *login admin*. Disini *admin* akan memasukkan *username* dan *password*. Jika *username* dan *password* yang dimasukkan benar, maka admin akan dihadapkan ke halaman seterusnya.

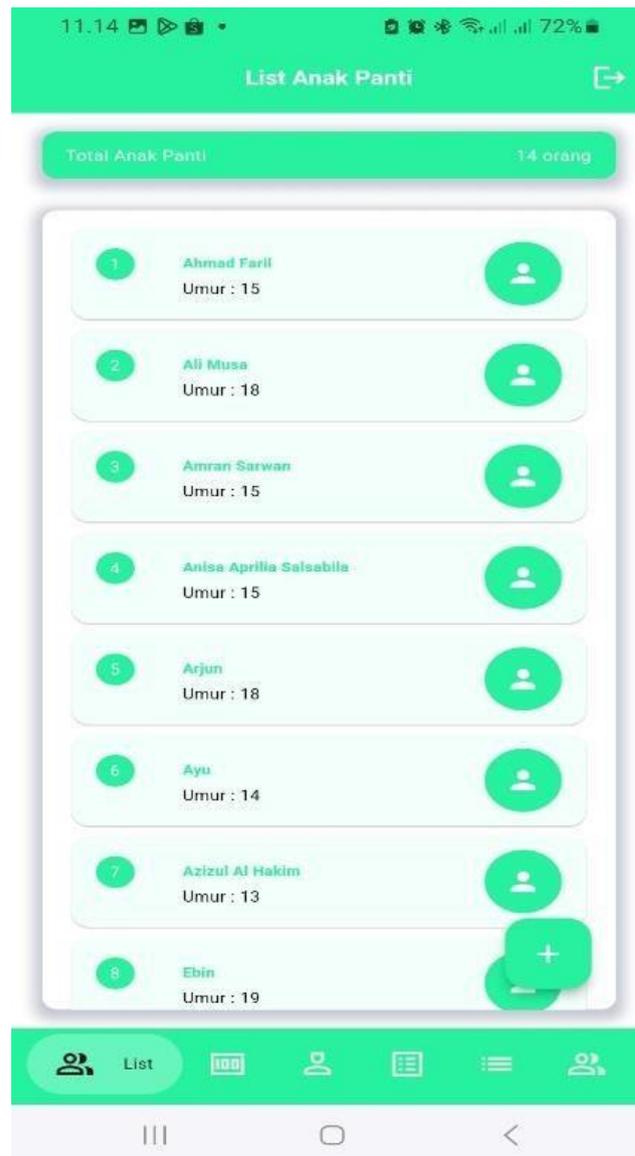
b. Halaman Dana Sosial



Gambar 4. 5 Halaman dana sosial

Diatas merupakan halaman dana sosial untuk admin, disini admin dapat menambah informasi dana sosial baru untuk ditampilkan selanjutnya baik kepada *user* maupun *admin*.

c. Halaman List Anak Panti



Gambar 4. 6 Halaman list anak panti

Diatas merupakan halaman list Anak Panti Asuhan Abadi Aisyiyah, disiniditampilkan list anak panti. Admin dapat menambah siswa baru, mengedit detail siswa yang telah ada serta menghapus anak panti.

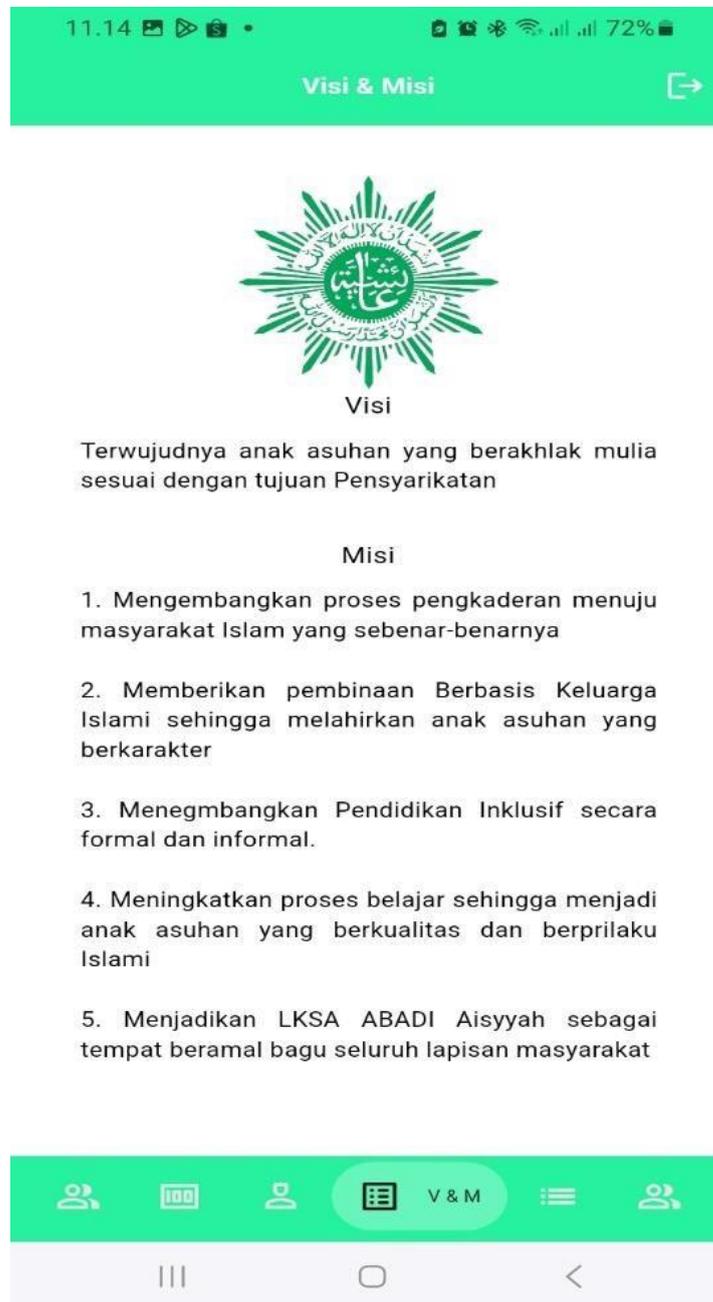
d. Halaman Profil



Gambar 4. 7 Halaman profil

Diatas merupakan halaman profil Panti Asuhan Abadi Aisyiyah. Disini ditampilkan nomor telpon panti asuhan dan alamat panti asuhan.

e. Halaman Visi & Misi



Gambar 4. 8 Halaman visi dan misi

Diatas merupakan halaman visi & misi panti asuhan abadi aisyyiah.

f. Halaman Sejarah Panti Asuhan



Gambar 4.9 Halaman sejarah panti asuhan

Diatas merupakan halaman sejarah panti asuhan, disini dapat melihatinformasi atau sejarah pada halaman ini.

D. Alat dan Bahan

Adapun kebutuhan yang digunakan untuk Membangun sebuah sistem dengan judul “Pengembangan Dana Sosial Pada Panti Asuhan Berbasis Android”.

1. Kebutuhan *Hardware*

Spesifikasi *hardware* pengguna aplikasi ini adalah sebagai berikut:

Tabel 4.1 Spesifikasi *Hardware* Komputer

Jenis	Spesifikasi
<i>Laptop/Komputer</i>	<i>Acer</i>
<i>Processor</i>	<i>Processor Intel Core i7 – 7500U Processor (up to 4.0GHz)</i>
<i>Memory</i>	<i>8 GB DDR 4 2400MHz</i>
<i>Hardisk</i>	<i>500 GB</i>

Tabel 4.2 Spesifikasi *Smartphone* Admin

Jenis	Spesifikasi
Jenis <i>Smartphone</i>	Vivo Y69
Layar	5 inch'
Ram	4 GB

Tabel 4.3 Spesifikasi *Smartphone* User

Jenis	Spesifikasi
Jenis <i>Smartphone</i>	<i>Xiaomi note 5 pro</i>
Layar	6 inch'
Ram	4 GB

2. Kebutuhan *Software*

Spesifikasi *software* pengguna aplikasi ini sebagai berikut :

Tabel 4.4 Spesifikasi *Software* Komputer

Jenis	Spesifikasi
Sistem Operasi	<i>Windows 10 Ultimate</i>

Tabel 4.5 Spesifikasi *Software Smartphone Admin*

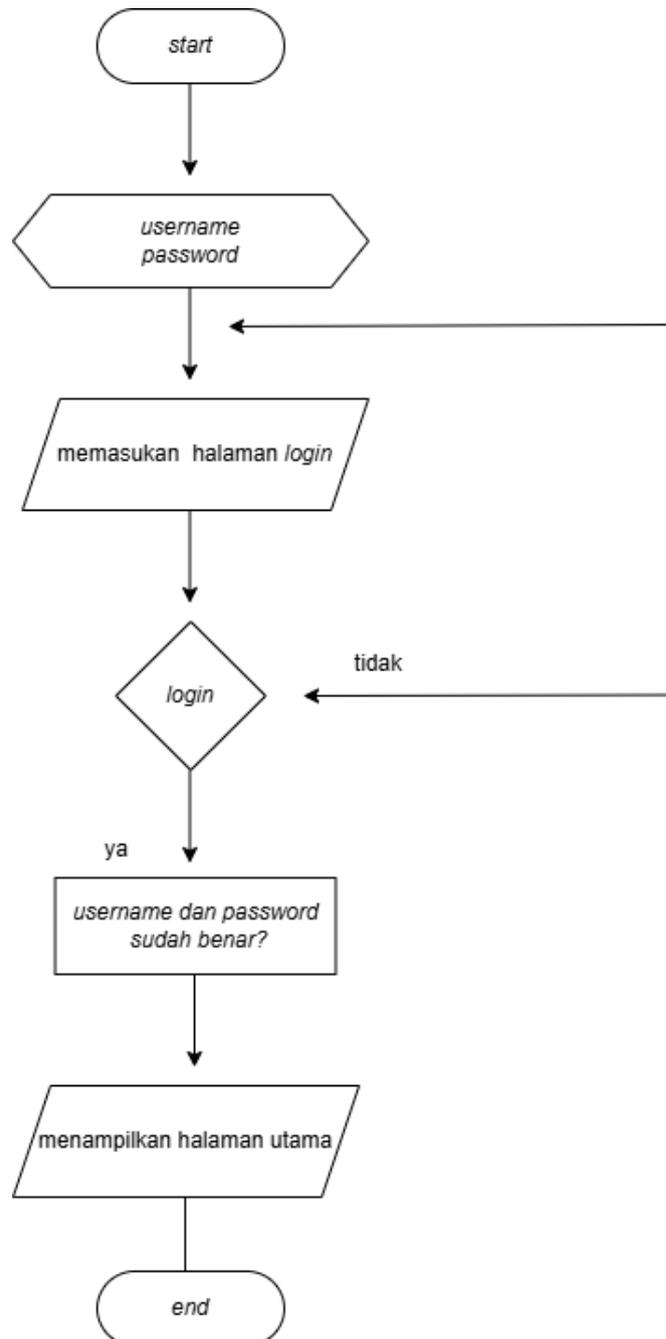
Jenis	Spesifikasi
Versi <i>Android</i>	<i>7.1 Nougat</i>
Versi perangkat lunak	PD1705F_EX_A_1.18.0

Tabel 4.6 Spesifikasi *Software Smartphone User*

Jenis	Spesifikasi
Versi <i>Android</i>	<i>Android 9 Pie</i>
Versi <i>MIUI</i>	<i>MIUI 10 Global 9.6.27 Beta</i>

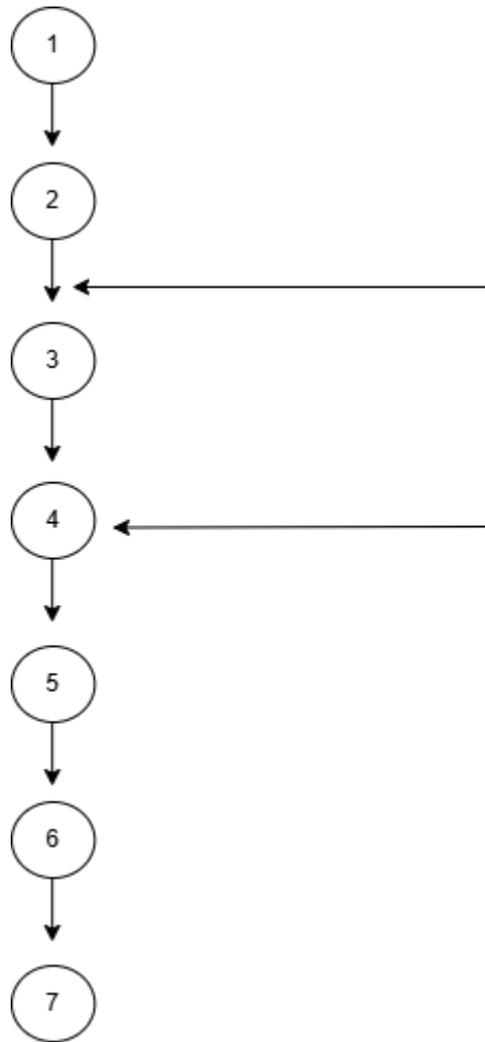
E. Hasil Pengujian

1) White Box Login Admin



Gambar 4. 10 Flowchart login admin

Dari *flowchart* yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4. 11 *Flowgraph login admin*

Menghitung jumlah *Region*, *Cyclomatic Complexity* dan *Independent Path*.

Independent Path.

- 1) *Path 1* = 1-2-3-4-5-7
- 2) *Path 2* = 1-2-3-4-5-6-4-5-6-7

Cyclomatic Complexity.

$$V(G) = E - N + 2$$

$$= 7 - 7 + 2 = 2$$

$$V(G) = P + 1$$

$$= 1 + 1$$

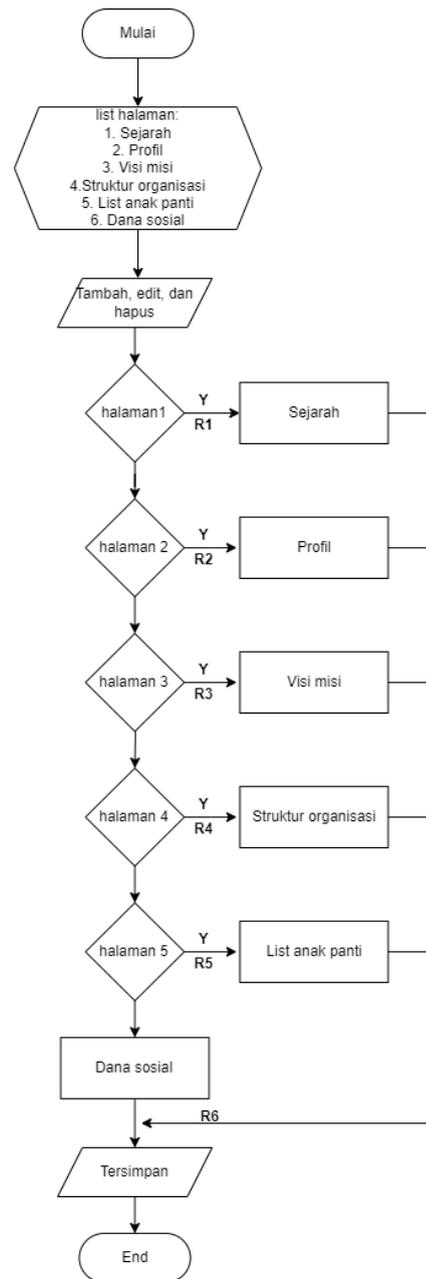
$$= 2$$

$$\text{Region} = 2$$

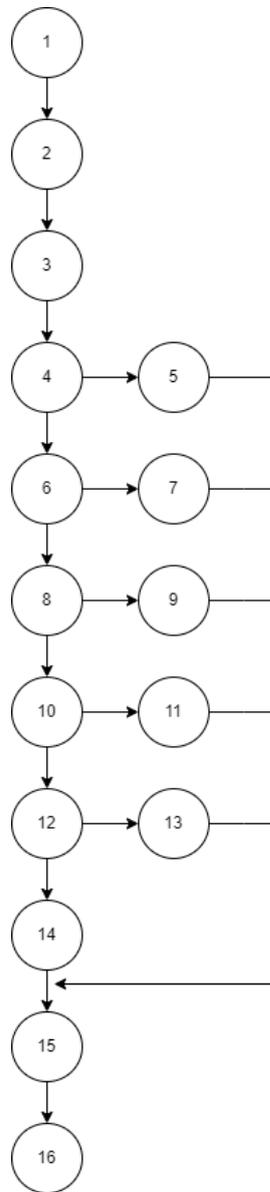
Grafik Matriks

Tabel 4. 4 *Grafik Matriks Login Admin*

	1	2	3	4	5	6	7	E - 1
1		1						$1 - 1 = 0$
2			1					$1 - 1 = 0$
3				1				$1 - 1 = 0$
4					1			$1 - 1 = 0$
5						1		$1 - 1 = 0$
6				1			1	$2 - 1 = 1$
7								$1 - 1 = 0$
8								0
SUM (E+1)								$1 + 1 = 2$

2) *White Box Admin***Gambar 4. 12** *Flowchart admin*

Dari *flowchart* yang digunakan untuk pengujian perangkat lunak, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4. 13 *Flowgraph admin*

Menghitung jumlah *Region*, *Cyclomatic Complexity* dan *Independent Path*.

Independent Path.

$$1) \quad \textit{Path 1} = 1-2-3-4-5-15-16$$

$$2) \quad \textit{Path 2} = 1-2-3-4-6-7-15-16$$

$$3) \quad \textit{Path 3} = 1-2-3-4-6-8-9-15-16$$

$$4) \quad \textit{Path 4} = 1-2-3-4-6-8-10-11-15-16$$

$$5) \quad \textit{Path 5} = 1-2-3-4-6-8-10-12-13-15-16$$

$$6) \quad \textit{Path 6} = 1-2-3-4-6-8-10-12-14-15-16$$

Cyclomatic Complexity.

$$V(G) = E - N + 2$$

$$= 20 - 16 + 2 = 6$$

$$V(G) = P + 1$$

$$= 5 + 1$$

$$= 6$$

Region = 6

Grafik Matriks

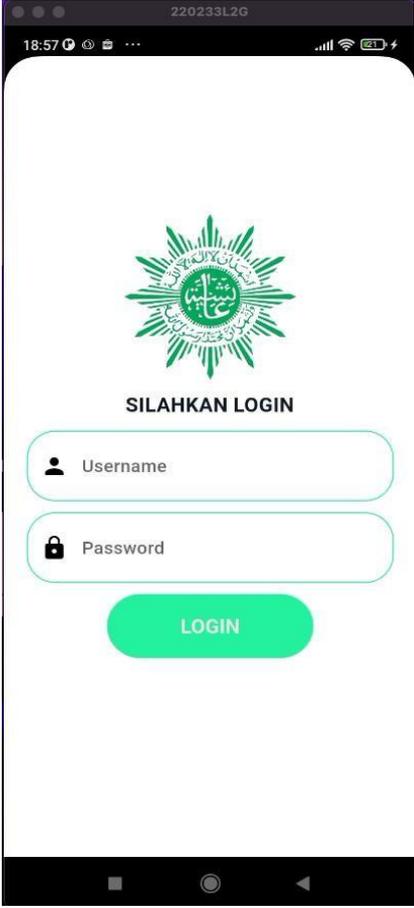
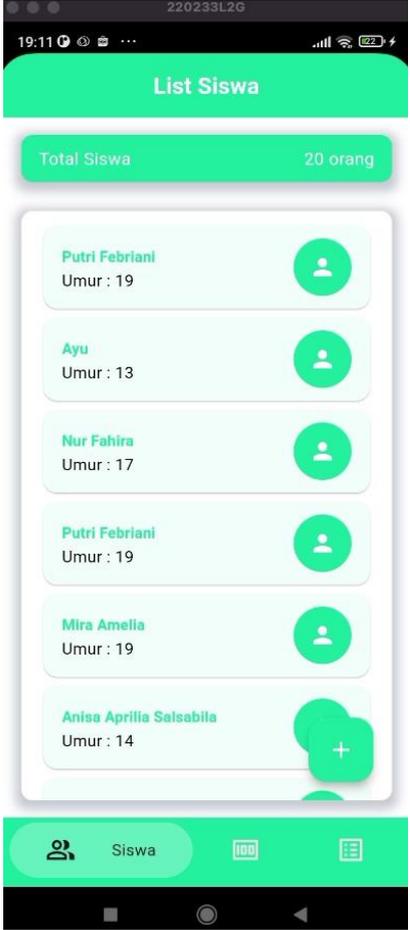
Tabel 4.5 Grafik matriks admin

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	E - 1
1		1															1 - 1 = 0
2			1														1 - 1 = 0
3				1													1 - 1 = 0
4					1	1											2 - 1 = 1
5															1		1 - 1 = 0
6							1	1									2 - 1 = 1
7															1		1 - 1 = 0
8								1	1								2 - 1 = 1
9															1		1 - 1 = 0
10										1	1						2 - 1 = 1
11															1		1 - 1 = 0
12												1	1				2 - 1 = 1
13															1		1 - 1 = 0
14															1		1 - 1 = 0
15																1	1 - 1 = 0
16																	
SUM (E+1)																	5 + 1 = 6

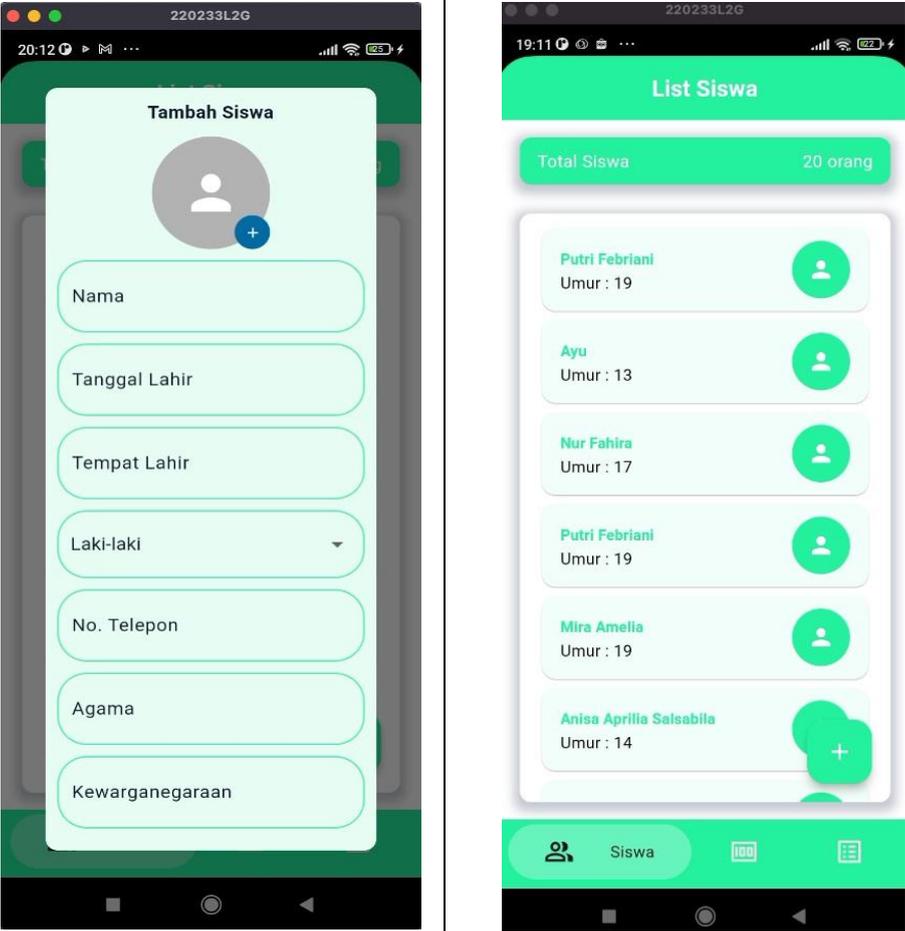
a. *Black Box*

1. *Black Box Login*

Tabel 4. 6 *Black box login admin*

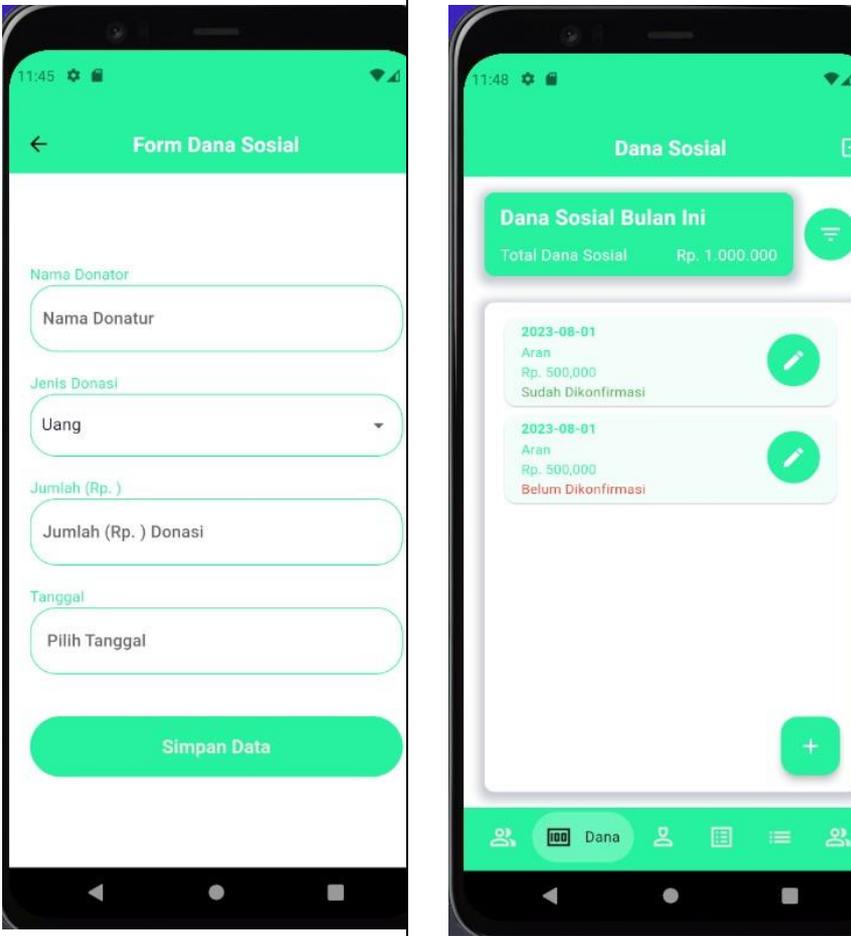
<i>Test Factor</i>	<i>Hasil</i>	<i>Keterangan</i>
Jika admin memasukkan <i>username</i> dan <i>password</i> yang benar	Berhasil	Berhasil, admin memasukkan <i>username</i> dan <i>password</i> yang benar
Screen Shoot		
		

2. *Black Box* Tambah SiswaTabel 4. 7 *Black box* tambah siswa

Test Factor	Hasil	Keterangan
Jika <i>admin</i> menambah siswa baru	Berhasil	Berhasil, siswa baru berhasil ditambahkan oleh admin
Screen Shoot		
		

3. Black Box Tambah Dana Sosial

Tabel 4. 8 Black box testing tambah dana sosial

<i>Test Factor</i>	Ha sil	Keterangan
Jika admin menambah informasi donasi	Berhasil	Berhasil, informasidonasi berhasil ditampilkan
Screen Shoot		
		

BAB V

PENUTUP

A. Kesimpulan

Pengembangan aplikasi dana sosial untuk panti asuhan berbasis *Android* bertujuan untuk meningkatkan kepercayaan akan transparansi pengelolaan donasi dan data anak-anak panti asuhan serta memudahkan keterlibatan masyarakat dalam mendukung kegiatan sosial panti asuhan.

Penelitian ini berhasil mengembangkan Aplikasi Pengelolaan Dana Sosial Panti Asuhan Berbasis *Android* dengan menggunakan *Javascprit, Flutter*, Aplikasi dana sosial berbasis *Android* dapat membantu panti asuhan dalam mengelola donasi dan data donatur dengan lebih baik, meningkatkan transparansi, dan Saran Saran yang dapat diberikan adalah Pengembangan aplikasi dana sosial *Android* untuk panti asuhan yaitu kebutuhan fitur seperti pemantauan dana dan pengelolaan data, diikuti desain antarmuka yang intuitif dan pengembangan aplikasi.

DAFTAR PUSTAKA

- Aditama, Y. (2018). Perancangan Website Sistem Informasi Panti Asuhan Yatim Piatu Al Huda. *Univesitas Muhammadiyah Sukarta*.
- Aditya Fajar Ramadhan, D. (2021). Aplikasi Pengenalan Perangkat Keras Komputer Berbasis Android Menggunakan Augmented Reality. *Jurnal Teknologi dan Sistem Informasi*, 24-31.
- [Apa Itu PHP? Simak Arti, Contoh Penggunaan, dan Manfaatnya \(hostinger.co.id\)](https://www.hostinger.co.id)
- Amelia, N. a. (2014). Aplikasi Penggalangan Data Donatur Dan Dana Sumbangan Pada Panti Asuhan . *Universitas Muhammadiyah Parepare*.
- [Mengenal Apa itu Aplikasi, Fungsi, dan Jenis-Jenisnya \(dewaweb.com\)](https://www.dewaweb.com)
- dilla, n. (2020). universitas muhammadiyah parepare. *penggalangan dana sosial*, 3, 1-65.
- <https://www.niagahoster.co.id/blog/pengertian-http/>
- [Bahasa Pemrograman Dart - Pengertian, Sejarah, Fitur, dan Aplikasi \(alterra.id\)](https://www.alterra.id)
- Fernando, M. (2013). *Membuat Aplikasi Android Augmented Reality Menggunakan Vuforia SDK dan Unity*. Jebres: Solo : Aronline.
- [Pengertian HTTP Beserta Fungsi dan Cara Kerjanya \(niagahoster.co.id\)](https://www.niagahoster.co.id)
- Gustari. (2011). *Pemrograman C#*.
<http://artagoes.blogspot.com/2011/09/pemrograman-C.html?m=1>.
- Harfiah, S. (2021). Aplikasi Media Pembelajaran Gerakan Sholat Sunnah Berbasis Augmented Reality. *Jurnal Sintaks Logika*, 204-208.
- Hasyim, A. H. (2021). *S.T., S.kom., M.T.* CV. Bangun Bumitama.

- I Dewa Gede Wahya Dhiyatmika, D. (2015). Aplikasi Augmented Reality Magic Book Pengenalan Binatang untuk siswa TK. *Lontar Komputer* , 589-595.
- IdCloudHost, M. (2020). *Pengertian Hardware : Fungsi, Jenis, Spesifikasi dan Contoh Hardware (Online)*. (<https://idcloudhost.com/pengertian-hardware-fungsi-jenis-spesifikasi-dan-contoh-hardware/>).
- Jajang Romansyah, A. M. (2019). Rancang Bangun Sitem Informasi Penggalangan Dana Sosial Panti Asuhan Di Kota Garut. *Sekolah Tinggi Teknologi Garut*.
- Jayusman. (2020). *Penjelasan JDK*.
- Kurniawan. (2017). *Android SDK (Software Development Kit)*.
- Lararenjana, E. (2021). *Pengertian Media Pembelajaran dan Jenis-Jenisnya (Online)*. <https://www.merdeka.com/jatim/pengertian-media-pembelajaran-dan-jenis-jenisnya-menarik-diketahui-kl.html>.
- Mahnun, N. (2012). Media Pembelajaran. *Jurnal Pemikiran Islam*, 27-33. Nugroho, A. (2008). *Menjadi Administrator Basis Data ORACLE 10 g*. Bandung: Infomatika Bandung.
- Pengertian, T. (2019). *Pengertian Hardware*. (<https://www.temukanpengertian.com/2013/01/pengertianhardware.html>).
- Reanbarka Dwi Septana, D. (2020). Media Pembelajaran Pengenalan Komponen Hardware Menggunakan Augmented Reality berbasis Android. *Information Management For Educators And Professionals*, 65-7

Salman, A. (2019). Augmented Reality Interaktif Pada Proses Pertumbuhan Tanaman Kacang Hijau. *Prosiding Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, 1-11.

Sampurna Dadi Riskiono, F. H. (2020). Sistem Informasi Manajemen Dana Donatur Berbasis Web. *Universitas Teknologi Indonesia*.

Sampurna Dadi Riskiono, F. H. (2020). Sistem informasi Manajemen Dana Donatur Berbasis Web. *Universitas Teknologi Indonesia*.

Sampurna Dadi Riskiono, F. H. (2020). Sistem Informasi Manajemen Dana Donatur Berbasis web. *Universitas Teknologi Indonesia*.

Sholiq. (2006). *Permodelah Sistem Informasi Berorientasi Objek Dengan Uml*.

[https://www.sekawanmedia.co.id/blog/phpmyadmin-adalah/Yogyakarta:](https://www.sekawanmedia.co.id/blog/phpmyadmin-adalah/Yogyakarta)

Graha Ilmu.

[Apa Itu Android Studio dan Mengapa Anda Harus Mempelajarinya? \(niagahoster.co.id\)](http://niagahoster.co.id)

Wikipedia. (2022). *Smartphone (Online)*. (en.wikipedia.org/wiki/smartphone).

Yovi, M. (2016). *Sejarah Android (Online)*.

<http://woocara.blogspot.kr/2015/02/sejarah-android-dan-nama-nama-versi-anroid.html>.

Yusuf, R. M. (2015). *Unity 3D-Game Engine. (Online)*.

(<http://www.hermantolle.com/class/docs/unity-3D-game-engine>) (di akses pada 18 maret 2022).