BAB I

PENDAHULUAN

A. Latar Belakang

Dalam kehidupan sehari-hari, kecelakaan atau kondisi darurat yang memerlukan pertolongan pertama sering kali tidak dapat dihindari. Pertolongan pertama pada kedaruratan (P3K) merupakan tindakan awal yang sangat penting untuk dilakukan sebelum bantuan medis profesional tiba. Pengetahuan dan keterampilan mengenai P3K menjadi kebutuhan yang sangat vital bagi masyarakat umum. Banyak nyawa yang bisa diselamatkan jika tindakan P3K yang tepat dapat segera diberikan. Namun, seringkali masyarakat mengalami kesulitan untuk mengakses informasi P3K yang benar dan akurat dengan cepat.

Teknologi informasi dan komunikasi telah mengalami perkembangan pesat dalam beberapa dekade terakhir. Perkembangan ini telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk dalam bidang kesehatan. Salah satu contoh yang paling nyata adalah penggunaan aplikasi berbasis *android* yang dapat diakses dengan mudah melalui smartphone. Aplikasi berbasis *android* memiliki potensi besar untuk digunakan sebagai media penyuluhan dan edukasi kesehatan karena mudah diakses oleh berbagai kalangan masyarakat.

Di sisi lain, kesadaran masyarakat tentang pentingnya pengetahuan P3K masih tergolong rendah yang menunjukkan bahwa banyak masyarakat yang belum mengetahui langkah-langkah P3K yang benar. Oleh karena itu, diperlukan suatu

upaya untuk meningkatkan pengetahuan masyarakat mengenai P3K melalui media yang mudah diakses dan digunakan.

Aplikasi panduan P3K berbasis *android* merupakan solusi yang tepat untuk menjawab kebutuhan ini. Dengan memanfaatkan kemajuan teknologi, aplikasi ini dapat menyediakan informasi dan panduan P3K secara cepat dan akurat.Aplikasi mobile memiliki keunggulan dalam hal kemudahan akses, interaktivitas, dan kemampuan untuk menyediakan informasi real-time. Aplikasi ini diharapkan dapat menjadi alat edukasi yang efektif dan efisien bagi masyarakat dalam menghadapi situasi darurat.

Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengembangkan sebuah aplikasi panduan P3K berbasis *android* yang dapat membantu masyarakat dalam memberikan pertolongan pertama secara cepat dan tepat. Aplikasi ini akan dilengkapi dengan berbagai fitur yang memudahkan pengguna untuk mengakses informasi P3K sesuai dengan jenis kecelakaan atau kondisi darurat yang terjadi. Diharapkan dengan adanya aplikasi ini, kesadaran dan pengetahuan masyarakat tentang P3K akan meningkat, sehingga dapat mengurangi risiko cedera serius atau kematian akibat keterlambatan penanganan.

B. Rumusan Masalah

Berdasarkan latar belakang masalah yang dipaparkan, rumusan masalah dalam penelitian ini adalah: Bagaimana merancang dan membangun aplikasi panduan p3k berbasis *android*

C. Tujuan Penelitian

Penelitian ini bertujuan untuk merancang dan mengembangkan sebuah aplikasi panduan pertolongan pertama pada kedaruratan (P3K) yang dapat diakses melalui perangkat *android*. Aplikasi ini diharapkan dapat memberikan panduan langkah-langkah P3K yang tepat dan akurat untuk berbagai jenis kecelakaan atau kondisi darurat.

D. Batasan Masalah

Berdasarkan rumusan masalah di atas, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini dibatasi hanya pada pengembangan aplikasi berbasis *android*. Aplikasi tidak dikembangkan untuk platform lain seperti *iOS* atau *windows phone*.
2. Aplikasi ini hanya mencakup panduan P3K untuk jenis-jenis kecelakaan dan kondisi darurat umum seperti luka, terkilir,dislokasi sendi, dan lain-lain. Kondisi medis yang kompleks dan memerlukan penanganan khusus oleh tenaga medis profesional tidak akan dibahas dalam aplikasi ini.
3. Aplikasi ini hanya akan menyediakan informasi panduan P3K dalam bentuk teks dan gambar. Fitur-fitur tambahan seperti simulasi interaktif atau video pelatihan langsung tidak termasuk dalam ruang lingkup pengembangan aplikasi ini.

E. Manfaat Penelitian

Adapun manfaat yang dapat diberikan dari penelitian ini adalah sebagai berikut:

1. Bagi akademik

   Sebagai bahan referensi bagi penulis lain untuk mengembangkan kemampuan di bidang yang sama.

2. Bagi penulis

   Penulis dapat mengimplementasikan ilmu yang didapat selama proses perkuliahan.

3. Bagi masyarakat

   Masyarakat dapat lebih mudah memperoleh informasi yang akurat dan cepat dari aplikasi terkait pertolongan pertama pada keadaan darurat saat terjadi kecelakaan.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Pertolongan pertama pada kedaruratan (P3K)

Pertolongan pertama pada kedaruratan (P3K) adalah langkah-langkah awal yang diambil untuk menolong korban kecelakaan sebelum mendapatkan bantuan medis profesional. P3K memiliki peran yang sangat penting dalam menyelamatkan nyawa, mencegah komplikasi lebih lanjut, dan mempercepat proses pemulihan korban. Dalam banyak situasi darurat, waktu sangatlah kritis, dan tindakan cepat serta tepat dari orang-orang di sekitar korban dapat membuat perbedaan besar antara hidup dan mati.



Gambar 2.1 Ilustrasi pertolongan pertama pada kedaruratan

P3K mencakup berbagai tindakan yang dapat dilakukan oleh orang awam maupun petugas non-medis. Beberapa tindakan dasar dalam P3K meliputi menghentikan pendarahan, memberikan bantuan pernapasan, mengimobilisasi patah tulang, serta menangani luka bakar dan cedera lainnya. Pengetahuan

mengenai P3K harus dimiliki oleh masyarakat umum agar mereka dapat memberikan bantuan yang efektif saat terjadi kecelakaan. Pelatihan dan edukasi mengenai P3K sangat diperlukan untuk meningkatkan kesiapsiagaan masyarakat dalam menghadapi situasi darurat.

Pengetahuan mengenai P3K harus dimiliki oleh masyarakat umum agar mereka dapat memberikan bantuan yang efektif saat terjadi kecelakaan (Sari,Yuliani, & Sukmawati, 2023).

2. Android

Android adalah sistem operasi untuk perangkat seluler berbasis *Linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. *Android* dikembangkan oleh sebuah perusahaan kecil di *Silicon Valley* bernama *Android* Inc. Selanjutnya, *Google* mengadopsi sistem operasi tersebut pada tahun 2005 dan mendeklarasikannya sebagai sistem operasi "*Open Source*". Hasilnya, siapa pun dapat menggunakannya secara gratis, termasuk kode sumber yang digunakan untuk mengkompilasi sistem operasi tersebut. *Android* sendiri merupakan sistem operasi dengan pengguna terbanyak di dunia, karena sifatnya yang terbuka bagi para *developer* untuk membuatnya lebih fungsional.



Gambar 2.2 Logo *Android*

Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang dapat digunakan pada berbagai perangkat penggerak. Smartphone pertama yang menggunakan *Android* adalah *HTC Dream*, yang dirilis pada tahun 2008 (Junadhi et al., 2021).

Adapun Kelebihan dari android yaitu:

a. Platform lengkap. Menyediakan alat yang berguna untuk membuat aplikasi yang kemudian dapat dikembangkan lebih lanjut oleh pengembang.
b. Platform *Open Source*. Pengembang mudah mengembangkannya karena bersifat terbuka
c. Platform gratis. Pengembang dapat dengan bebas mengembangkan, mendistribusikan, dan memperdagangkan sistem operasi Android tanpa harus membayar royalti untuk mendapatkan lisensi

3. *Android Studi*o

Android Studio merupakan IDE (*Integrated Development Environment*) untuk mengembangkan aplikasi *Android*. Aplikasi ini diterbitkan oleh *Google* pada tanggal 16 Mei 2013 dan tersedia secara gratis dengan lisensi *Apache* 2.0. Android Studio menggantikan perangkat lunak pengembangan Android sebelumnya, yaitu *Eclipse*.

IDE (*Integrated Development Environment*) adalah aplikasi untuk pengembang perangkat lunak yang berisi fungsi-fungsi terintegrasi yang diperlukan untuk membangun perangkat lunak, seperti editor kode, *debugger*, *kompiler*, dan sebagainya. Android Studio sendiri dikembangkan berdasarkan

IntelliJ IDEA, mirip dengan *Eclipse*, disertai dengan plugin ADT (*Android Development Tools*).

Android Studio memiliki fitur:

a. Proyek berdasarkan *Gradle Build.*

b. *Refactoring* cepat dan perbaikan *bug*.

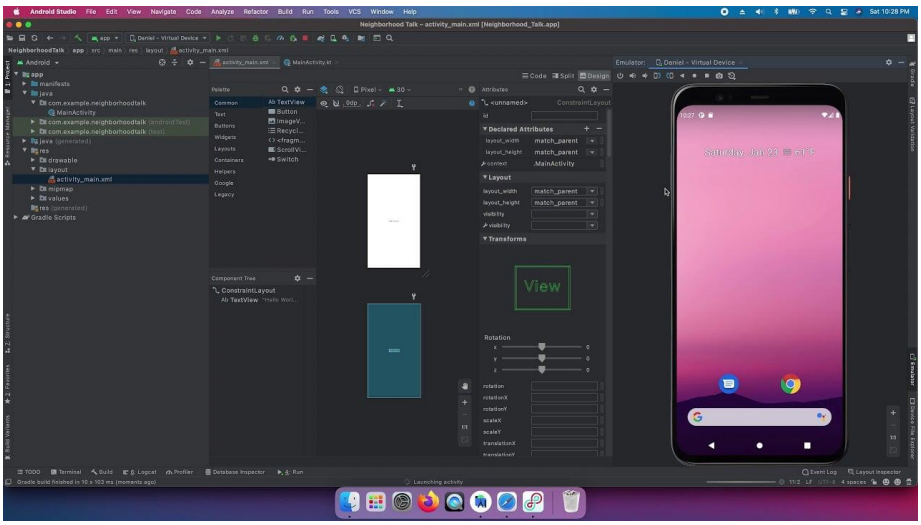
c. Tools baru bernama "*Lint*" mengklaim dapat dengan cepat memantau kecepatan, kegunaan, dan kompatibilitas aplikasi.

d. Mendukung *Proguard* dan penandatanganan aplikasi untuk keamanan.

e. Memiliki *GUI* aplikasi *Android* lebih mudah.

f. Didukung oleh *Google Cloud Platform* untuk setiap aplikasi yang dikembangkan.

Android Studio dipilih karena memiliki banyak fitur yang memudahkan para programmer, khususnya programmer tingkat dasar yang ingin mempelajari lebih lanjut tentang *Android*. Meski menggunakan *Android Studio* memakan waktu, kelebihan fungsionalitas dan dukungan yang diberikan membuatnya menjadi pilihan yang baik dalam pengembangan aplikasi *Android*. Perangkat lunak *Android Studio* menjadi salah satu software yang cukup populer sebab termasuk ke dalam *software* yang mudah dalam penggunaannya. Selain itu, perangkat lunak ini juga sudah mendukung berbagai macam fitur menarik untuk pembuatan aplikasi (Yanti & Lesmideryati, 2022).



Gambar 2.3 Tampilan *Android Studio*

Meski menggunakan *Android Studio* memakan cukup banyak RAM pada perangkat PC, namun *Android Studio* memiliki sejumlah keunggulan lain, yaitu:

a. Instant Run

b. Intelligent Code Editor

c. Sistem Versi yang Flexibel

d. Dioptimalkan untuk semua perangkat Android

e. Didesain untuk Tim

4. Java

Java adalah bahasa pemrograman berbasis komputer ke berorientasi objek pemrograman komputer. *Java* dirancang seperti itu sehingga ukurannya kecil,simple dan portable (dapat dialihkan antara yang berbeda platform dan sistem operasi). Program yang mana dihasilkan dalam bahasa *Java* dapat berupa applet (aplikasi kecil yang berjalan di atas *browser web*) atau dalam bentuk aplikasi mandiri yang berjalan dengan program Java Interpreter.



Gambar 2.4 Logo *Java*

Java adalah bahasa pemrograman yang populer, dikembangkan oleh *Sun Microsystems*. Salah satu penggunaan terbesar Java adalah dalam pembuatan aplikasi *native* untuk Android. Bahasa pemrograman ini bersifat *multiplatform*, artinya dapat digunakan di berbagai *platform*, seperti *desktop*, *Android*, dan bahkan untuk sistem operasi *Linux* (Sibarani et al., 2019).

Menurut Moch. Anif, R.A.M. Dyah Ayuningtyas, dan Dwi Agung Nugroho (2020), bahasa pemrograman java memiliki beberapa karakteristik, antara lain:

a. Sederhana Bahasa *Java* menggunakan syntax yang mirip dengan bahasa pemrograman *C++*. Akan tetapi syntax di bahasa pemrograman *Java* banyak yang telah diperbaiki, terutama dengan menghilangkan pointer yang rumit dan *multiple inheritance*. Java juga menggunakan *garbage collection* dan *automatic memory allocation*.

b. Berorientasi Objek Bahasa pemrograman Java merupakan bahasa pemrograman yang berorientasi objek sehingga memungkinkan program untuk dibuat secara modular dan digunakan kembali.

c. Terdistribusi Bahasa pemrograman Java dibuat untuk memudahkan distribusi aplikasi dengan terintegrasinya networking libraries ke dalam Java.

d. *Interpreted* Program Java dijalankan dengan menggunakan Interpreter, yaitu JVM (*Java Virtual Machine*). Hal tersebut mengakibatkan source code Java yang telah dikompilasi menjadi bytecodes (sebuah format yang tidak tergantung pada arsitektur tertentu yang didesain untuk mengirimkan kode ke banyak platform hardware dan *software* secara efisien) dapat dijalankan dalam berbagai platform.

e. *Robust* Bahasa pemrograman *Java* mempunyai reliabilitas yang tinggi. Kompiler pada bahasa ini memiliki kemampuan untuk mendeteksi kesalahan yang lebih baik dibandingkan bahasa pemrograman lain. Bahasa Java memiliki *Runtime Exception Handling* yang berfungsi untuk membantu mengatasi kesalahan dalam pemrograman.

f. *Secure Java*, sebagai bahasa pemrograman aplikasi internet dan terdistribusi, mempunyai beberapa mekanisme keamanan yang berfungsi menjaga agar aplikasi yang dijalankan tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

g. *Architecture Neutral* Program *Java* tidak bergantung pada platform dimana program akan dijalankan. Cukup dengan membuat program yang dapat dijalankan di berbagai platform dengan menggunakan Java *Virtual Machine*.

h. *Portable Source code* (kode sumber) dan program pada Java dapat dibawa dengan mudah ke berbagai platform yang berbeda tanpa harus dilakukan kompilasi ulang.

i. *Multithreaded* Bahasa pemrograman Java dapat membuat sebuah program yang mampu melakukan beberapa pekerjaan secara simultan dan bersamaan.

j. *Performance* Kinerja bahasa pemrograman ini sering kali disebut kurang, tetapi kinerjanya bisa ditingkatkan dengan menggunakan compiler Java lain seperti buatan Microsoft, Symantec, maupun Inprise yang menggunakan JIT (Just In Time) Compilers. k. Dynamic Java dapat didesain untuk mampu dijalankan di lingkungan yang dinamis. Perubahan suatu class dengan menambahkan properties maupun metode mampu dilakukan tanpa menggangu program yang menggunakan class tersebut.

5. *Library* Java

Menurut Yanti, N., dan Lesmideryati, D. (2022), "*Library* Java adalah bagian penting dari bahasa pemrograman Java. Dengan memanfaatkan *library* yang tepat, Anda dapat mengembangkan aplikasi Java yang lebih cepat, berkualitas, dan aman."

Keunggulan *Library* Java:

a. Menghemat waktu dan tenaga: Penggunaan *library* Java dapat mengurangi kebutuhan untuk menulis kode dari awal, sehingga mempercepat proses pengembangan aplikasi.

b. Meningkatkan kualitas dan keamanan: *Library* Java yang dikembangkan dengan baik telah melalui pengujian dan perbaikan sehingga meminimalisir bug dan meningkatkan keamanan aplikasi.
c. Standarisasi dan interoperabilitas: Banyak *library* Java mengikuti standar pemrograman yang umum, sehingga memudahkan kolaborasi dan integrasi dengan komponen lain.
d. Variasi dan kelengkapan: Terdapat banyak *library* Java yang tersedia untuk berbagai kebutuhan, mulai dari pengembangan antarmuka pengguna, *database*, jaringan, hingga *machine learning*.

Faktor Pemilihan *Library*:

a. Kebutuhan fungsional: Pilih *library* yang menyediakan fungsi dan fitur sesuai kebutuhan aplikasi Anda.
b. Dokumentasi dan komunitas: Ketersediaan dokumentasi yang jelas dan komunitas aktif memudahkan proses belajar dan mengatasi masalah.
c. Lisensi dan dukungan: Perhatikan jenis lisensi (*open source* atau komersial) dan ketersediaan dukungan teknis.
d. Performa dan popularitas: Pertimbangkan efisiensi dan kestabilan library, serta tingkat popularitas dan adopsi *developer* lain.

Contoh *Library* Java Populer:

a. *ZXing*: *ZXing ("Zebra Crossing")* adalah API populer untuk pemrosesan kode QR di Java.
b. *Spring Framework*: *Framework web development* dan *enterprise* dengan fitur lengkap dan fleksibel.

c. *Apache Commons Lang*: *Library* berisi utilitas umum *untuk string, date, collection,* dan lainnya.

d. *Apache POI*: *Library* untuk membaca dan menulis file Microsoft Office (Word, Excel, PowerPoint).

e. *JavaFX*: *Framework* untuk pengembangan aplikasi *desktop* dan *mobile* dengan *user interface* yang kaya.

f. *Guava*: *Library* berisi koleksi utilitas tambahan untuk bahasa Java, memperkaya *functionality* standar.

Saat memilih *library* Java, penting untuk mempertimbangkan kebutuhan fungsional, dokumentasi dan komunitas, lisensi dan dukungan, performa dan popularitas. Dengan mempertimbangkan faktor-faktor tersebut, akan dapat menemukan *library* Java yang tepat bagi kebutuhan sebuah aplikasi yang akan dikerjakan (Fatdha, et al., 2021).

6. JDK (*Java Development Kit)*

JDK (*Java Development Kit*) adalah paket perangkat lunak yang menyediakan berbagai fungsi dan alat yang diperlukan untuk pengembangan aplikasi dengan bahasa pemrograman Java. JDK meliputi komponen-komponen kunci seperti:

a. *Java Runtime Environment* (JRE): Komponen ini memungkinkan pengguna untuk menjalankan aplikasi Java yang telah dikompilasi. JRE berisi JVM, kumpulan kelas-kelas dasar, dan perpustakaan yang diperlukan untuk menjalankan program Java.

b. *Java Virtual Machine* (JVM): JVM adalah mesin virtual yang menjalankan bytecode Java. JVM bertanggung jawab untuk mengeksekusi program Java dengan menerjemahkan bytecode menjadi instruksi yang dapat dijalankan oleh sistem operasi dan perangkat keras target. JVM juga memiliki tugas lain, seperti manajemen memori, garbage collection, dan keamanan.
c. Kompiler Java: Kompiler Java mengonversi kode sumber Java menjadi bytecode, yaitu kode yang berbentuk bahasa mesin virtual yang dapat dipahami oleh JVM.

Menurut Fatdha, T. S. E., Junadhi, Susanti, Yenni, H., dan Zoromi, F. (2021), "JDK adalah paket perangkat lunak yang sangat penting bagi *developer* Java. Dengan adanya JDK, pengembang dapat dengan mudah mengembangkan aplikasi Java yang berkualitas dan aman."

Berikut adalah beberapa keuntungan menggunakan JDK:

a. Portabilitas: Aplikasi Java yang dikompilasi dapat dijalankan di berbagai platform dan sistem operasi tanpa perlu kompilasi ulang. Hal ini karena JVM bertanggung jawab untuk menerjemahkan *bytecode* menjadi instruksi yang dapat dijalankan oleh sistem operasi dan perangkat keras target.
b. Keamanan: JVM memiliki *built-in security features* yang membantu melindungi aplikasi Java dari serangan *malware* dan lainnya.
c. Efisiensi: JVM menggunakan teknik-teknik optimasi untuk meningkatkan kinerja aplikasi Java.

7. SDK (*Software Development Kit)*

Dalam pengembangan aplikasi mobile, SDK (*Software Development Kit*) merupakan komponen penting yang menyediakan berbagai fungsi dan API yang diperlukan untuk membangun aplikasi mobile. SDK dapat membantu pengembang untuk membangun aplikasi mobile yang berkualitas dan efisien (Fitri et al., 2022).

Keberadaan SDK mempermudah pengembang untuk mengakses berbagai fitur sistem dan kemampuan khusus dari platform yang dituju, memastikan bahwa aplikasi yang dihasilkan dapat berfungsi secara optimal dan responsif.

Dalam dunia pengembangan perangkat lunak, SDK menjadi komponen krusial yang mendefinisikan bagaimana sebuah aplikasi dapat diintegrasikan dan beroperasi di atas platform tertentu. Dengan berbagai alat dan API yang tersedia, SDK memungkinkan pengembang untuk fokus pada logika dan fungsionalitas aplikasi tanpa harus memikirkan detail teknis dari platform yang digunakan. Hal ini memfasilitasi proses pengembangan yang lebih cepat dan efektif, sekaligus memastikan bahwa aplikasi yang dihasilkan sesuai dengan standar kualitas dan kinerja yang diharapkan.

Dalam konteks Android, SDK menjadi jembatan antara ide dan realisasi aplikasi. Dengan dukungan dari SDK, pengembang dapat mengeksplorasi potensi penuh dari sistem operasi Android dan menghasilkan aplikasi yang memanfaatkan keunikan serta kecanggihan platform tersebut. Sebagai alat yang mendukung, SDK berperan penting dalam menginspirasi inovasi dan kreativitas

pengembang, memungkinkan terciptanya aplikasi yang memenuhi berbagai kebutuhan pengguna dengan beragam fitur dan fungsionalitas.

8. MYSQL

MySQL adalah salah satu jenis database yang terkenal dan termasuk jenis *RDBMS (Relational Database Management System)*. Kepopuleran *MySQL* disebabkan karena *MySQL* menggunakan bahasa *SQL* sebagai bahasa dasar untuk query dan bersifat open source di berbagai platform.

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*), di mana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.



Gambar 2.5 Logo *MYSQL*

Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya, terutama dalam kecepatan. Berikut beberapa keistimewaan MySQL, antara lain:

a) Portability

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac OS X Server, Solaris Amiga*, dan masih banyak lagi.

b) Multiuser

MySQL dapat digunakan oleh beberapa pengguna (*user*) dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

c) Security

MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses pengguna dengan sistem perizinan yang mendetail serta kata sandi terenkripsi.

d) *Scalability* dan *Limits*

MySQL mampu menangani database dalam skala besar, dengan jumlah rekaman lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya (Huda, 2022).

9. PHP



Gambar 2.6 Logo *PHP*

PHP adalah bahasa *server-side* yang secara khusus digunakan untuk aplikasi *web*. Bahasa *PHP* berjalan di sisi *server*, artinya ia berjalan pada *server* dan bisa dimasukkan di antara bahasa *HTML* (*HyperText Markup Language*). Oleh karena itu, kode *PHP* tidak akan terlihat lagi dan hasil akhir dalam bentuk *HTML* akan dikirimkan ke *browser* (Sutarman, 2020). Karena *PHP* dijalankan di komputer *server*, itu dikenal sebagai bahasa pemrograman sisi *server*. Ini berbeda dengan bahasa pemrograman sisi klien, seperti *JavaScript*, yang beroperasi pada klien, yaitu *web browser*.

Pada awalnya, *PHP* adalah singkatan dari *Personal Home Page*. *PHP* digunakan untuk mengembangkan situs *web* pribadi, sesuai dengan namanya. Seiring berjalannya waktu, *PHP* berkembang menjadi bahasa pemrograman *web* yang sangat kuat, tidak hanya digunakan untuk membuat halaman *web* dasar tetapi juga untuk situs-situs populer seperti *Wikipedia*, *WordPress*, *Joomla*, dan lainnya yang dikunjungi oleh jutaan orang.

10. Phpmyadmin

Phpmyadmin adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen *MySQL*. Dengan menggunakan *phpmyadmin*, anda dapat membuat *database*, membuat tabel, meng-*insert*, menghapus dan meng-*update data* dengan *GUI* dan terasa lebih mudah, tanpa perlu mengetikkan perintah *SQL* secara *manual*. Karena berbasis *web*, maka *phpmyadmin* dapat dijalankan di banyak *OS*, selama dapat menjalankan *web server* dan *MySQL*

11. API

API (*Application Programming Interface*) merupakan sekumpulan perintah, fungsi, serta protokol yang digunakan oleh programmer saat membangun suatu perangkat lunak pada sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi, sehingga mempermudah proses pengembangan perangkat lunak. API menyediakan lapisan abstraksi yang mengizinkan aplikasi untuk berkomunikasi dengan sistem operasi atau layanan lainnya tanpa harus memahami detail implementasinya.

API terdiri dari berbagai jenis, termasuk API berbasis web, API sistem operasi, API perpustakaan perangkat lunak, dan API aplikasi. Misalnya, API berbasis web memungkinkan aplikasi untuk berkomunikasi dengan server melalui protokol *HTTP* atau *HTTPS*, sedangkan API sistem operasi memungkinkan perangkat lunak untuk mengakses sumber daya sistem seperti *file*, perangkat keras, atau proses lainnya.

API juga dapat mendukung berbagai fungsi, mulai dari autentikasi pengguna, pengambilan dan pengiriman data, hingga manajemen sesi dan transaksi. Dengan menggunakan API, pengembang dapat mempercepat waktu pengembangan, meningkatkan efisiensi, dan memastikan interoperabilitas antara berbagai sistem dan aplikasi. API juga mendukung modularitas dalam pengembangan perangkat lunak, yang memungkinkan kode untuk digunakan kembali di berbagai aplikasi dan konteks.

12. UML (*Unified Modelling Language*)

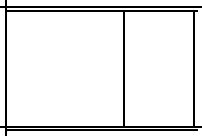
UML (*Unified Modeling Language*) adalah alat desain sistem berorientasi objek. Secara filosofis kemunculan UML terinspirasi dari konsep yang sudah ada yaitu konsep pemodelan *Object Oriented* (OO), karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh objek dan digambarkan atau dicatat dalam simbol-simbol yang cukup spesifik. proses standar dan independen.

UML adalah pemodelan untuk membantu proses perancangan sistem sehingga meminimalisir kesalahan dalam membuat program. Penerapan UML ini menggambarkan struktur aktor yang terlibat, aktifitas setiap aktor, proses danmekanisme dimana memberikan kemudahan perancangan dalam membatu proses pengkodingan menjadi sebuah aplikasi (Voutama, 2022).

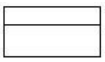
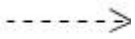
UML memiliki tiga kategori utama yaitu diagram struktur, diagram perilaku, dan diagram interaksi. Dimana setiap kategori mempunyai diagram yang menjelaskan arsitektur sistem dan saling terintegrasi.

Adapun daftar simbol UML yaitu :

Tabel 2.1 *Symbol Use Case Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *Use Case*.
2		*Dependency*	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (*Independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		*Generalization*	Hubungan di mana objek anak *(Descendent*) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (*Ancestor*).
4		*Include*	Menypesifikasikan bahwa *Use Case* sumber secara Eksplisit.
5		*Extend*	Menypesifikasikan bahwa *use case* target memperluas perilaku dari *use* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menypesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemenelemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

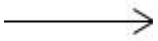
Tabel 2.2 *Symbol Class Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan di mana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

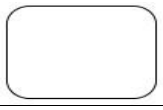
Tabel 2.3 *Symbol Sequence Diagram*

No.	AMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat in*form*asi in*form*asi tentang aktivitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat in*form*asi in*form*asi tentang aktivitas yang terjadi

Tabel 2.4 *Symbol State Chart Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai Link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2.5 *Symbol Acitivity Diagram*

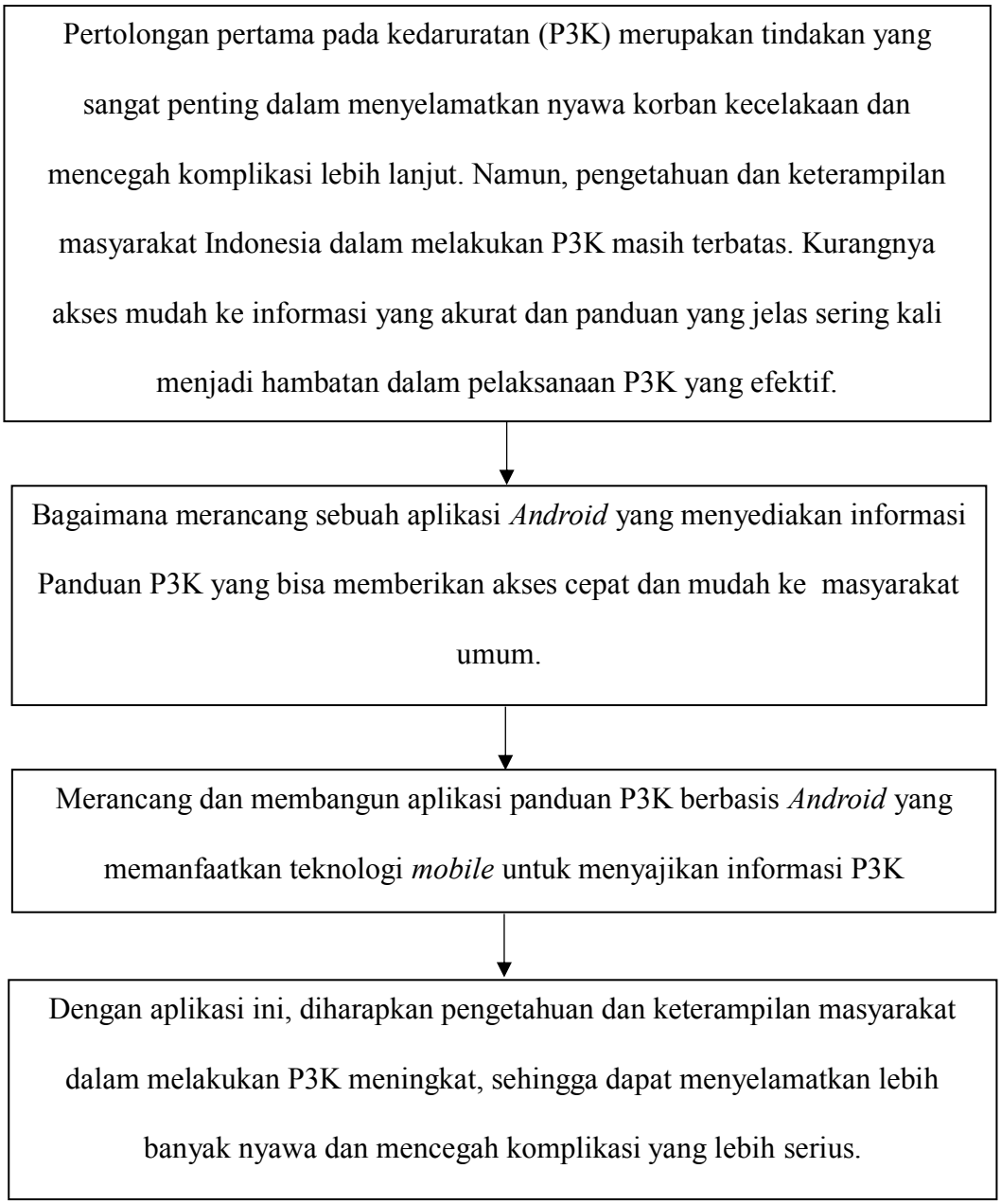
No.	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Memperlihatkan bagaimana interaksi masing-masing antarmuka
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

B. Kajian Hasil Penelitian Terdahulu

1. (Rahmawati, N.,et al., 2023). "Pengembangan Aplikasi *Mobile* Panduan P3K dan Petunjuk Arah Sarana Kesehatan Berbasis *Android*." Universitas Pamulang.Penelitian ini bertujuan untuk mengembangkan aplikasi *mobile* panduan P3K dan petunjuk arah sarana kesehatan berbasis *Android*. Aplikasi ini dilengkapi dengan informasi tentang berbagai jenis cedera, langkah-langkah pertolongan pertama, dan peta interaktif yang menunjukkan lokasi sarana kesehatan terdekat.
2. (Pratama, Y., & Rachmatika, R., 2023). "Aplikasi Edukasi Pertolongan pertama pada kedaruratan (P3K) Berbasis *Android*." Universitas Pamulang.Penelitian ini bertujuan untuk mengembangkan aplikasi edukasi pertolongan pertama pada kedaruratan (P3K) berbasis *Android*. Aplikasi ini dilengkapi dengan materi edukasi tentang berbagai jenis cedera, langkah-langkah pertolongan pertama, dan video tutorial.
3. (Tari, S., et al., 2023). "Rancang Bangun Aplikasi Pembelajaran Pertolongan pertama pada kedaruratan (P3K) Berbasis *Android*." Universitas Islam Negeri Sunan Kalijaga Yogyakarta.Penelitian ini bertujuan untuk merancang dan membangun aplikasi pembelajaran pertolongan pertama pada kedaruratan (P3K) berbasis *Android*. Aplikasi ini dilengkapi dengan materi edukasi interaktif, simulasi, dan penilaian untuk membantu meningkatkan keterampilan pengguna dalam memberikan pertolongan pertama

C. Kerangka Pikir

Untuk Memahami alur peneltian diatas, diuraikan ke dalam kerangka berpikir yang akan disajikan dalam bentuk diagram ini:



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian kualitatif deskriptif, dimana informasi yang dikumpulkan berbentuk kata-kata, gambar, bukan angka.Dengan pendekatan (*Library Research*) penelitian ini dilakukan dengan menggunakan buku sebagai referensi untuk penulisan.

B. Lokasi dan Waktu

Penelitian dilakukan di Perpustakaan Umum Kota Parepare, Ujung Sabbang, Kecamatan Ujung, Kota Parepare, Sulawesi Selatan 91114. Penelitian ini akan berlangsung selama ± 2 (dua) bulan di tahun 2023.

C. Alat dan Bahan

1. Perangkat keras (*Hardware*)
   a. Laptop *Asus Business Notebook* dengan spesifikasi *hardware*:
      - *Processor* : *Intel Core* i5 11th i5-11320H @ 3.2GHz
      - *Installed RAM* : 8GB
      - *SSD* : 512GB
      - *Graphic Card* : *Intel(R) Iris(R) Xe Graphic*
2. Perangkat lunak (*Software*)
   a. Windows 11 *Home Single Language*
   b. Android Studio

c. Java

d. JDK

e. SDK

D. Teknik Pengumpulan Data

1. Studi Pustaka

   Studi pustaka merupakan strategi pengumpulan data dari berbagai sumber literatur termasuk buku, internet, makalah, jurnal dan bahan lain yang relevan dengan penelitian yang dilakukan.

2. Observasi

   Observasi dilakukan dengan cara mengumpulkan data-data yang akan dijadikan bahan dasar dalam perancangan sistem.

3. Wawancara

   Dokumentasi yaitu teknik mengumpulkan data dengan mencari atau memilah-milah informasi dalam literatur yang relevan dengan rumusan masalah.

E. Metode Pengujian

Dalam penelitian ini, digunakan 2 (dua) metode dalam pengujian datanya yaitu *blackbox testing* dan *whitebox testing*:

1. *Blackbox testing*

   Blackbox testing terfukus pada fungsional dari program yang ada. Pada *Blackbox testing* diuji dengan cara menjalankan program kemudian diamati apakah program tersebut apakah berhasil atau tidak. *Blackbox testing*

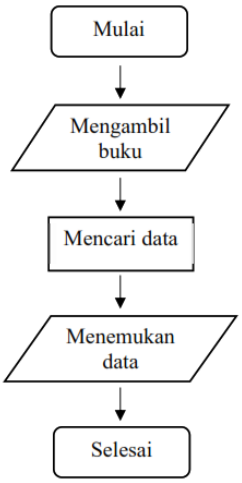
menggunakan teknik *equivalence partitions* yang merupakan pengujian berdasarkan masukan setiap menu yang terdapat pada program, setiap menu masukan dilakukan pengujian melalui klasifikasi dan pengelompokan berdasar fungsinya.

2. *Whitebox testing*

Whitebox testing bertujuan untuk mengetahui apakah struktur pada aplikasi yang dibuat sudah sesuai dengan ketentuan. *whitebox testing* menitikberatkan pada pengujiaan dengan mengecek detail peracangan perangkat lunak. *whitebox testing* dimilai dengan mendefiniskan semua alur dari perangkat lunak, kemudian membangun kasus yang akan digunakan dalam proses pengujian, kemudian menguji kasus tersebut untuk memperoleh hasilnya.

F. Desain Sistem

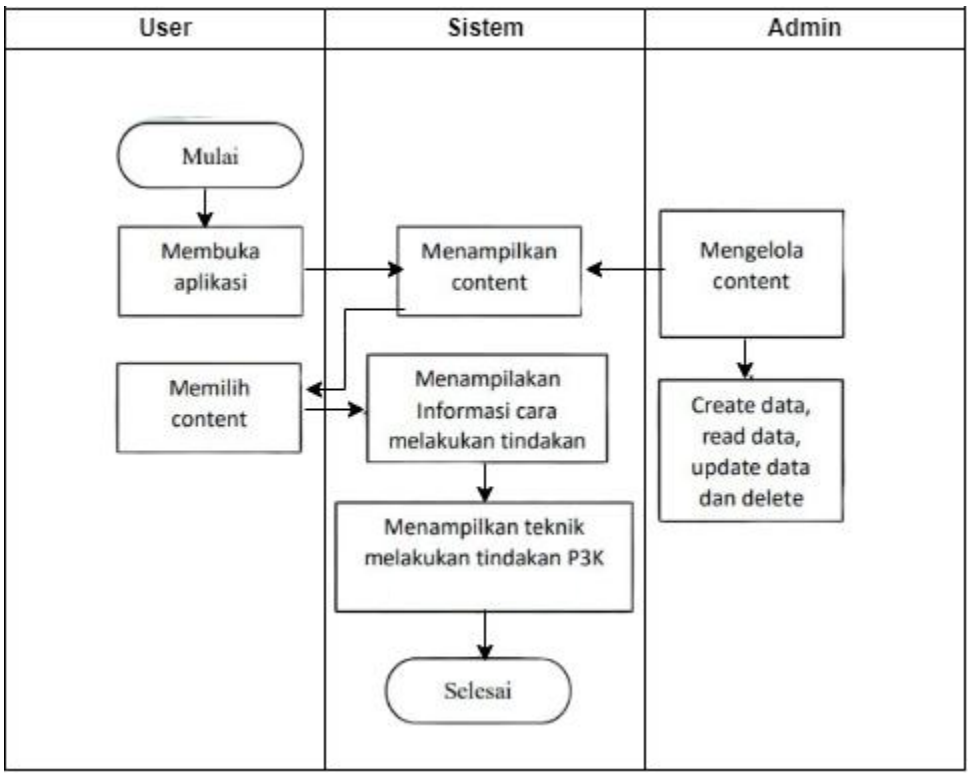
1. Desain *system* yang berjalan



Gambar 3.1 *Flowchart* sistem yang berjalan

Pada desain sistem yang berjalan masih mengandalkan pencarian informasi dari buku atau internet, sehingga butuh waktu lama dan bisa menyulitkan saat darurat. Karena tidak ada fitur pencarian otomatis, pengguna harus mencari sendiri, yang bisa jadi sulit jika tidak paham istilah medis. Selain itu, tanpa navigasi yang jelas dan hanya berupa teks, panduan jadi kurang praktis dan sulit dipahami. Akibatnya, respons jadi lebih lambat, padahal dalam keadaan darurat butuh tindakan cepat. Maka dari itu, sistem berbasis teknologi yang lebih mudah dan efisien sangat dibutuhkan.

2. Desain *system* yang diusulkan



Gambar 3.2 *Flowmap* sistem yang diusulkan

User memulai dengan membuka aplikasi dari perangkat *Android* mereka. Setelah aplikasi terbuka, *user* akan melihat halaman awal yang menyambut mereka.

Di halaman awal, terdapat fitur pencarian yang memungkinkan user untuk mencari panduan pertolongan pertama berdasarkan situasi darurat yang mereka hadapi. *User* dapat memasukkan kata kunci atau memilih dari daftar situasi darurat yang sudah disediakan. Setelah *user* menemukan panduan yang sesuai, mereka akan memilihnya. Aplikasi akan menampilkan detail teknik pertolongan pertama yang sesuai dengan situasi yang dicari. Panduan ini akan disajikan dengan jelas, langkah demi langkah, untuk membantu user dalam situasi darurat.

Admin memiliki akses ke fitur manajemen konten dalam aplikasi. *Admin* dapat menampilkan semua konten panduan pertolongan pertama yang tersedia dalam sistem. *Admin* dapat menambahkan panduan pertolongan pertama baru ke dalam aplikasi. Saat menambahkan, admin akan mengisi detail yang diperlukan seperti judul, deskripsi, langkah-langkah, dan gambar pendukung jika diperlukan. *Admin* juga dapat mengedit panduan yang sudah ada untuk memastikan informasi selalu terbaru dan akurat. Pengeditan ini mencakup perubahan pada teks, gambar, dan langkah-langkah dalam panduan. Jika ada panduan yang tidak lagi relevan atau akurat, admin dapat menghapusnya dari sistem. Penghapusan ini memastikan bahwa *user* hanya mendapatkan informasi yang *up-to-date* dan bermanfaat. Semua perubahan yang dilakukan admin pada konten panduan akan secara otomatis terlihat oleh *user* dalam aplikasi.

BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Aliran Data UML

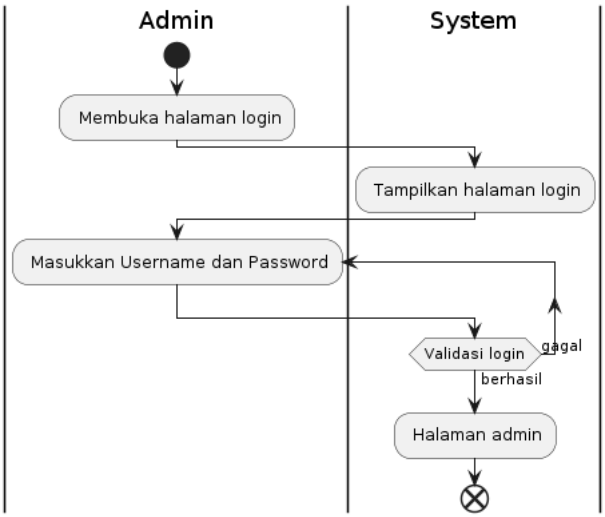
Perencanaan sistem ini akan menganalisis aliran data dari sistem menggunakan diagram UML. Rancangan ini akan mencakup beberapa diagram penting, seperti *Acitivity Diagram*, dan *Sequence Diagram*.

1. *Acitivity Diagram*

Acitivity Diagram adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, mengspesifikasikan dan membangun perangkat lunak. *Acitivity Diagram* ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pada sebuah sistem.

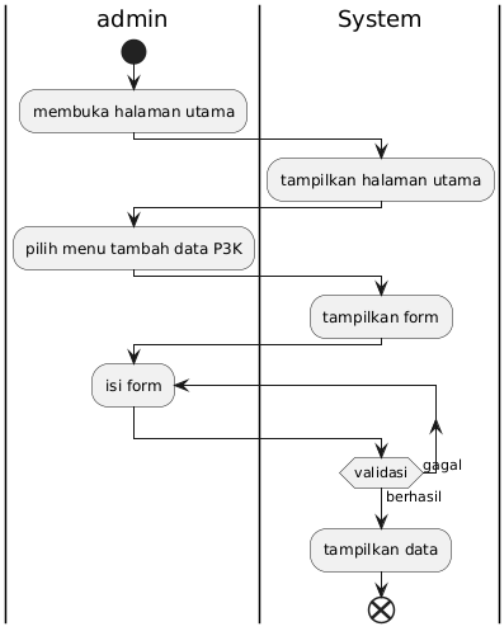
a. *Acitivity Diagram* Admin

1) *Acitivity Diagram* login



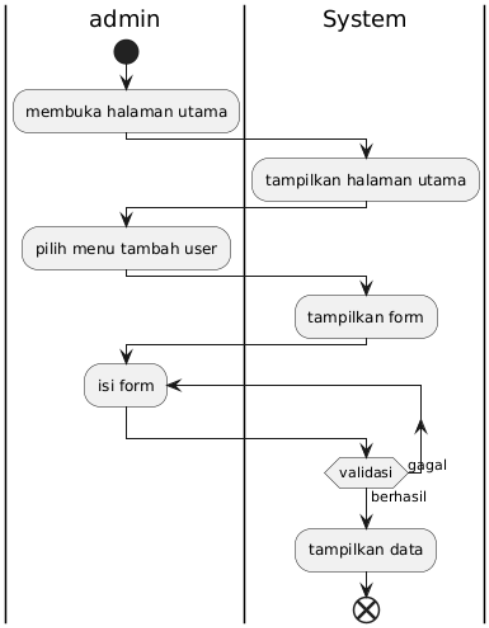
Gambar 4. 1 *Acitivity Diagram* login

2) *Acitivity Diagram* tambah data P3K



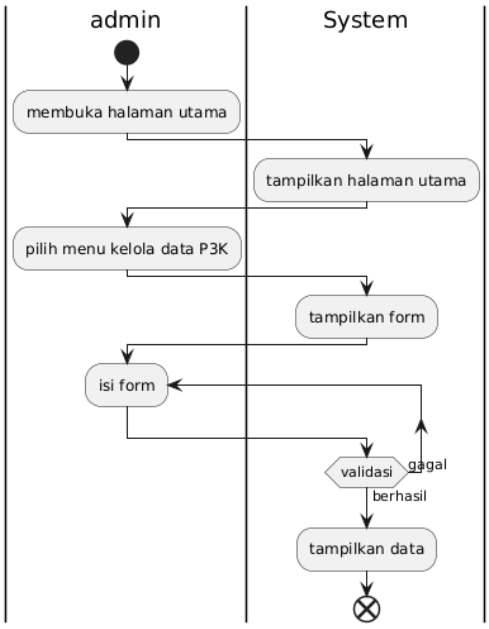
Gambar 4. 2 *Acitivity Diagram* tambah data P3K

3) *Acitivity Diagram* tambah data user



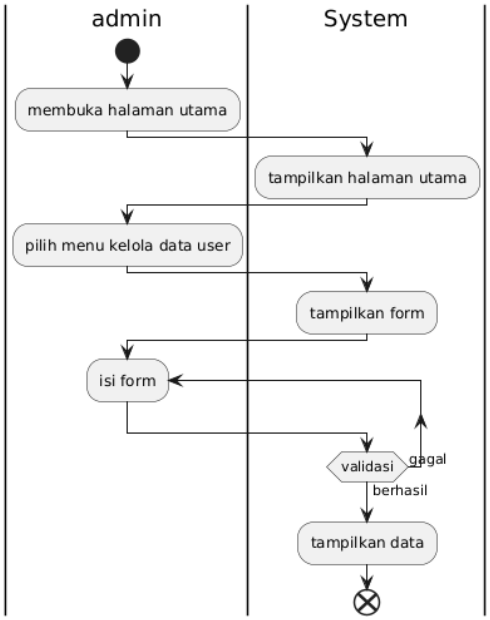
Gambar 4. 3 *Acitivity Diagram* tambah data user

4) *Acitivity Diagram* kelola data P3K



Gambar 4. 4 *Acitivity Diagram* kelola data P3K

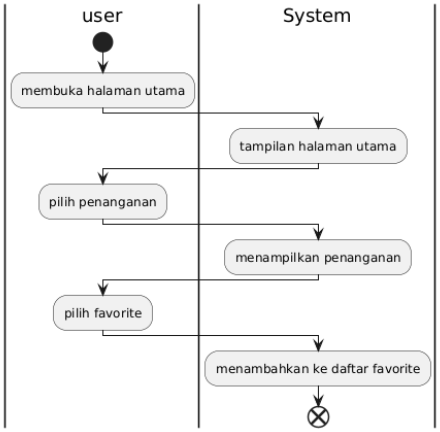
5) *Acitivity Diagram* kelola data user



Gambar 4. 5 *Acitivity Diagram* kelola data P3K

b. *Acitivity Diagram* User

1) *Acitivity Diagram favorite*



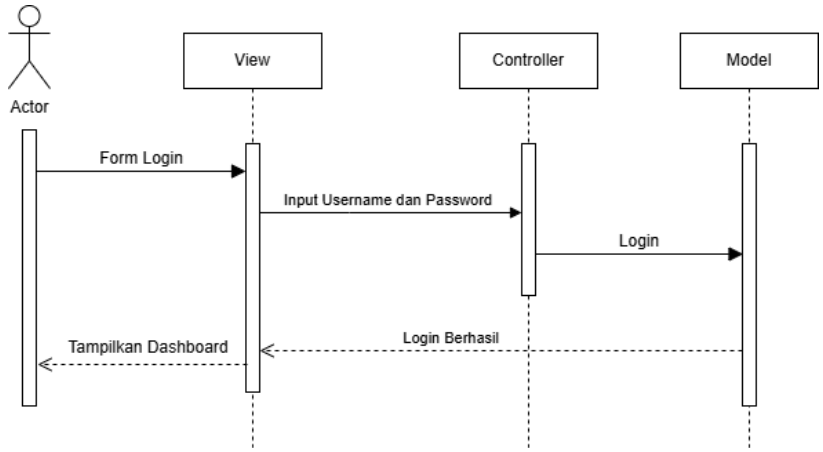
Gambar 4. 6 *Acitivity* Diagram *Favorite*

2. *Sequence Diagram*

Sequence Diagram merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan message (pesan) apa yang dikirim dan kapan pelaksanaannya.

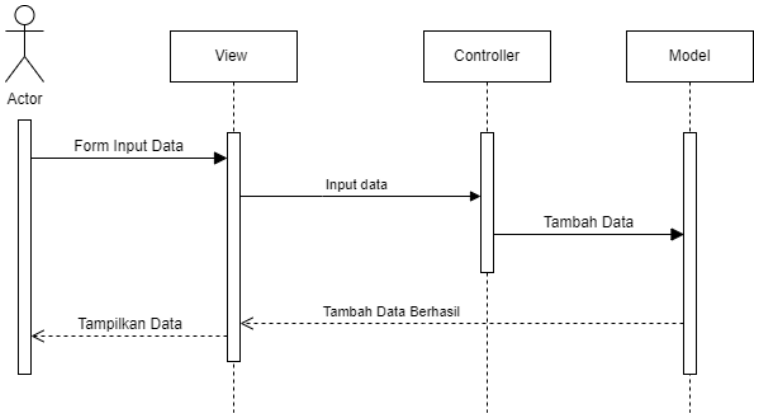
a. *Sequence Diagram* Admin

1) *Sequence Diagram* login



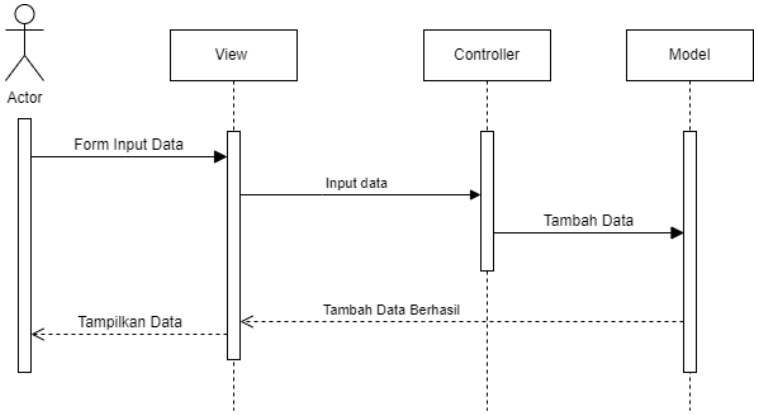
Gambar 4. 7 *Sequence Diagram* login

2) *Sequence Diagram* tambah data P3K



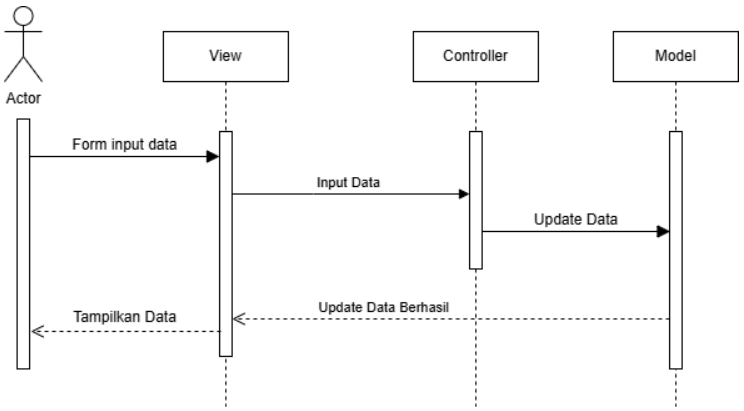
Gambar 4. 8 *Sequence Diagram* tambah data P3K

3) *Sequence Diagram* tambah data user



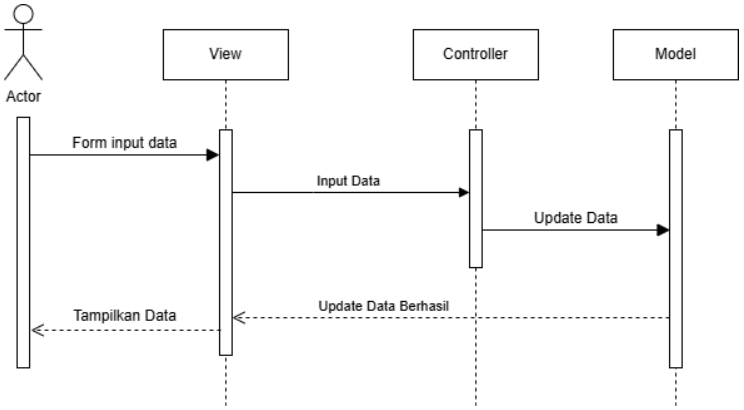
Gambar 4. 9 *Sequence Diagram* tambah data user

4) *Sequence Diagram* kelola data P3K



Gambar 4. 10 *Sequence Diagram* kelola data P3K

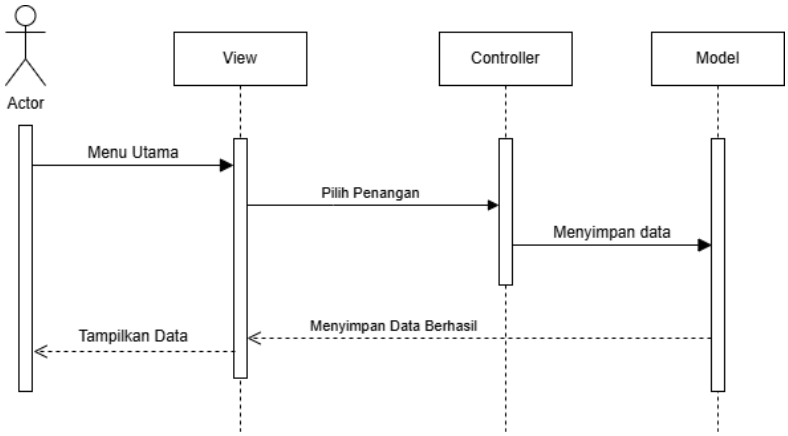
5) *Sequence Diagram* kelola data user



Gambar 4. 11 *Sequence Diagram* kelola data user

b. *Sequence Diagram* User

1) *Sequence Diagram favorite*



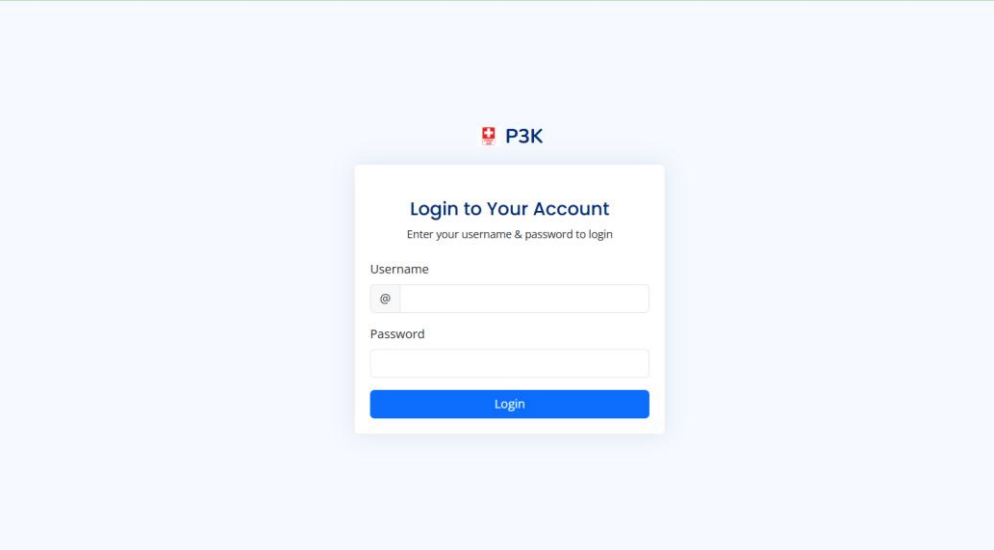
Gambar 4. 12 *Sequence* Diagram *favorite*

B. Detail Sistem

1. Admin

a. Halaman login

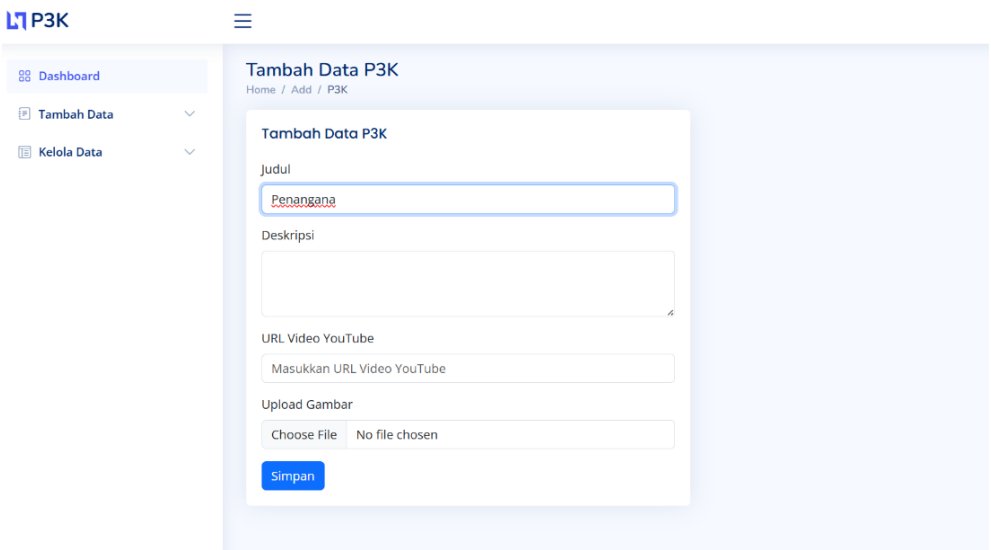
Halaman Login merupakan titik awal yang digunakan oleh admin untuk memperoleh akses masuk ke beragam fitur yang tersedia di dalam halaman menu admin. Ini adalah pintu gerbang yang memungkinkan admin untuk masuk ke dalam sistem dengan mengautentikasi identitas mereka.



Gambar 4. 13 Halaman login

b. Halaman tambah data P3K

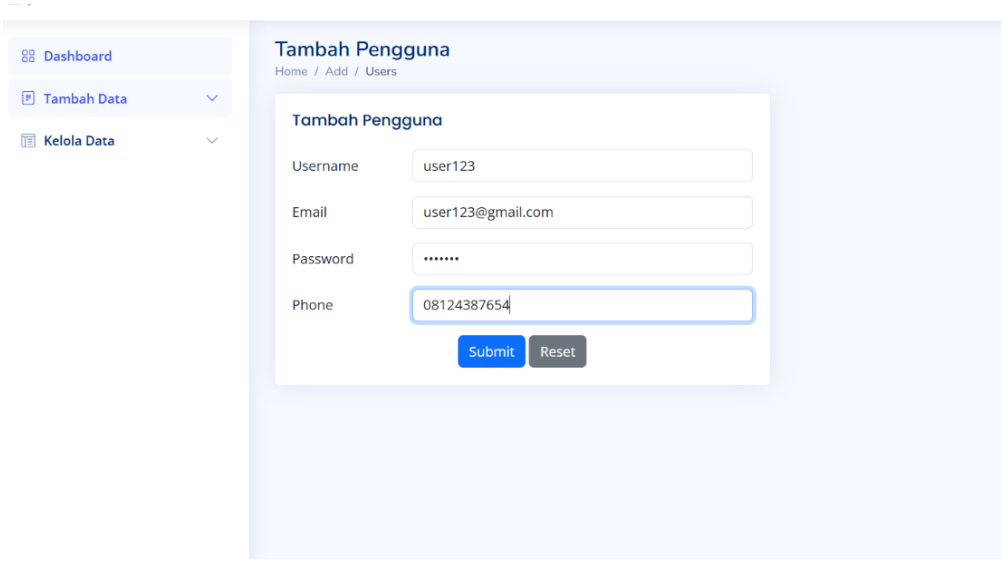
Halaman Tambah Data P3K memungkinkan admin untuk menambahkan informasi terkait pertolongan pertama pada berbagai kondisi kegawatdaruratan. Pada halaman ini, admin dapat menginput data seperti nama kondisi darurat, langkah-langkah pertolongan pertama, gambar ilustrasi, serta video panduan jika tersedia.



Gambar 4. 14 Halaman tambah data P3K

c. Halaman tambah data user

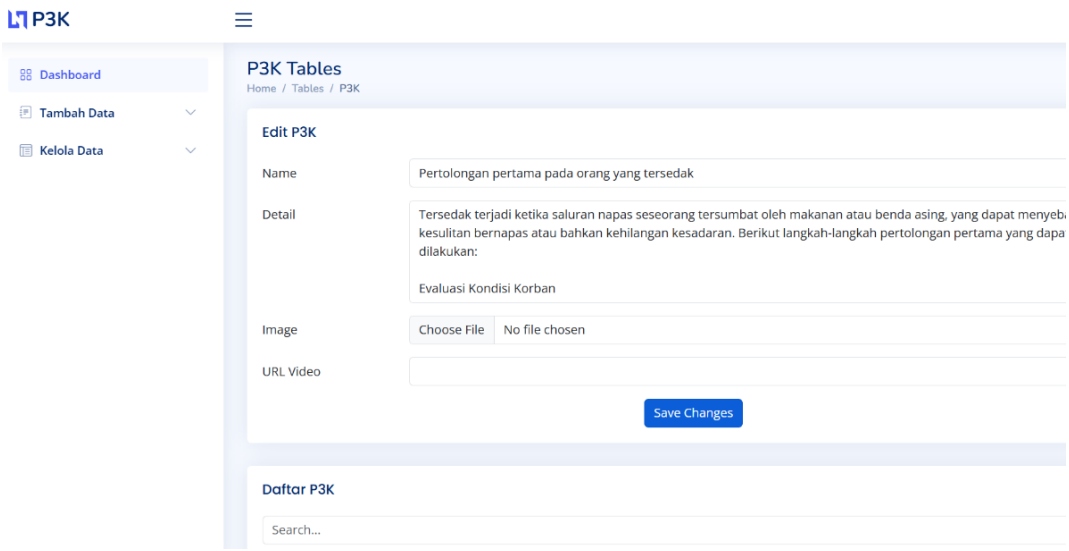
Halaman Tambah Data User digunakan oleh admin untuk menambahkan akun pengguna baru ke dalam sistem. Pada halaman ini, admin dapat menginput informasi seperti username, email, dan password.



Gambar 4. 15 Halaman tambah data user

d. Halaman kelola data P3K

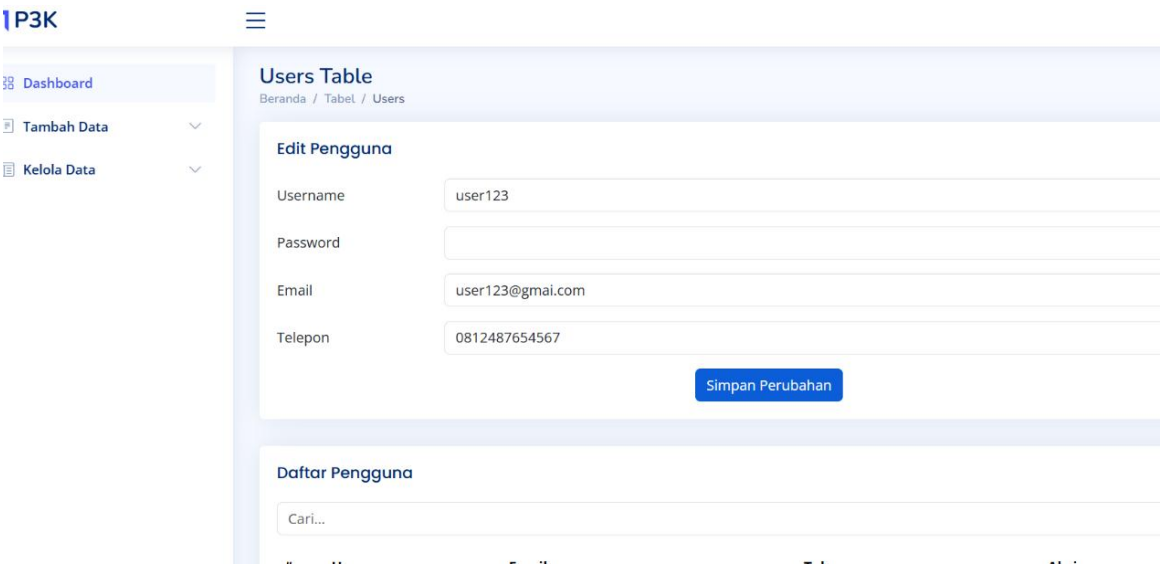
Halaman Kelola Data P3K memungkinkan admin untuk mengelola informasi terkait pertolongan pertama pada kegawatdaruratan. Pada halaman ini, admin dapat melihat daftar data P3K yang telah dimasukkan, melakukan pencarian, mengedit informasi jika ada perubahan, serta menghapus data yang tidak diperlukan.



Gambar 4. 16 Halaman kelola data P3K

e. Halaman kelola data user

Halaman Kelola Data User memungkinkan admin untuk mengelola informasi pengguna dalam sistem. Pada halaman ini, admin dapat melihat daftar pengguna yang terdaftar, mencari pengguna berdasarkan nama atau email, mengedit data pengguna jika diperlukan.

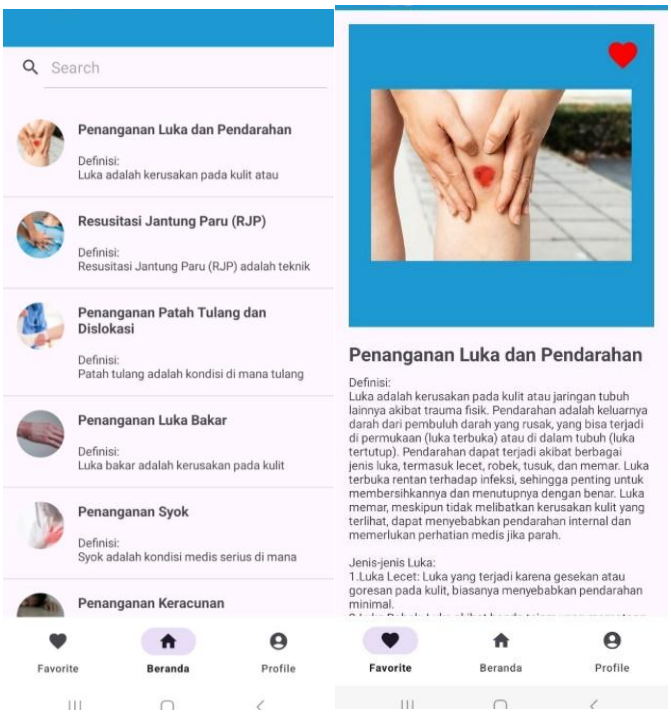


Gambar 4. 17 Halaman kelola data user

2. User

a. Halaman detail penanganan

Halaman Detail Penanganan merupakan halaman yang ditampilkan setelah pengguna berhasil login. Pada halaman ini, pengguna dapat melihat daftar lengkap panduan Pertolongan Pertama pada Kegawatdaruratan (P3K). Setiap daftar mencakup informasi seperti nama kondisi darurat, langkah-langkah pertolongan pertama, gambar ilustrasi, serta video panduan jika tersedia.

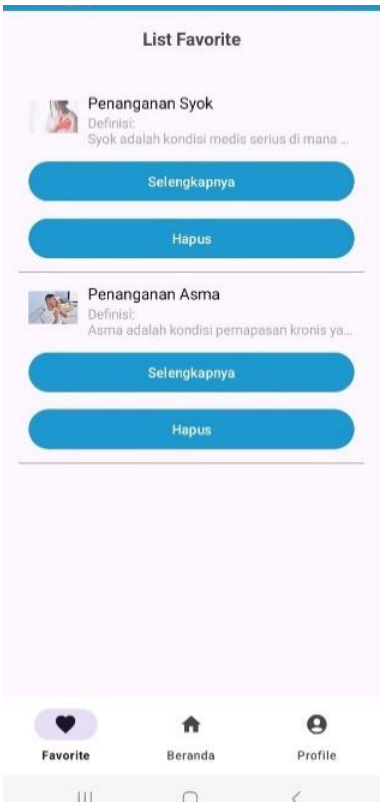


Gambar 4. 18 Detail Penanganan

b. Halaman *favorite*

Halaman *Favorite* memungkinkan pengguna untuk menyimpan dan mengakses daftar penanganan Pertolongan Pertama pada Kegawatdaruratan (P3K) yang sering digunakan atau dianggap penting.

Pengguna dapat menandai suatu penanganan sebagai favorit, sehingga informasi tersebut lebih mudah ditemukan tanpa perlu mencari ulang.



Gambar 4. 19 Halaman Favorite

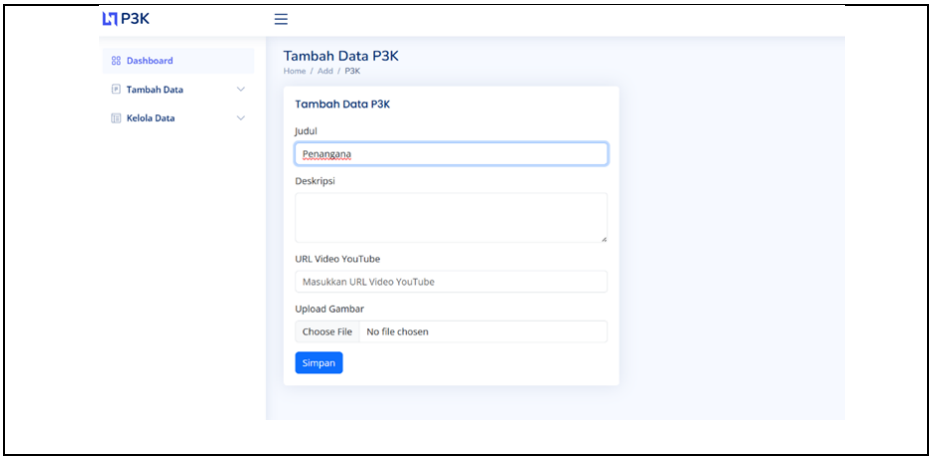
C. Pengujian Sistem

1. *Black Box*

a. *Black Box testing* tambah P3K

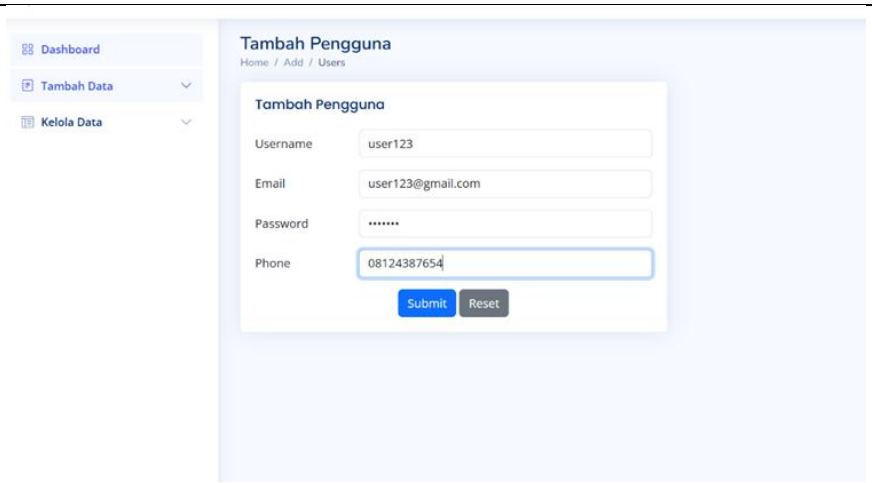
Tabel 4. 1 *Black Box* testing tambah P3K

Tes Faktor	Hasil	Keterangan
Admin mengisi form tambah P3K dan menekan tombol simpan	✓	Berhasil, tampil informasi bahwa simpan data berhasil
Screenshot		



b. *Black Box testing* tambah user

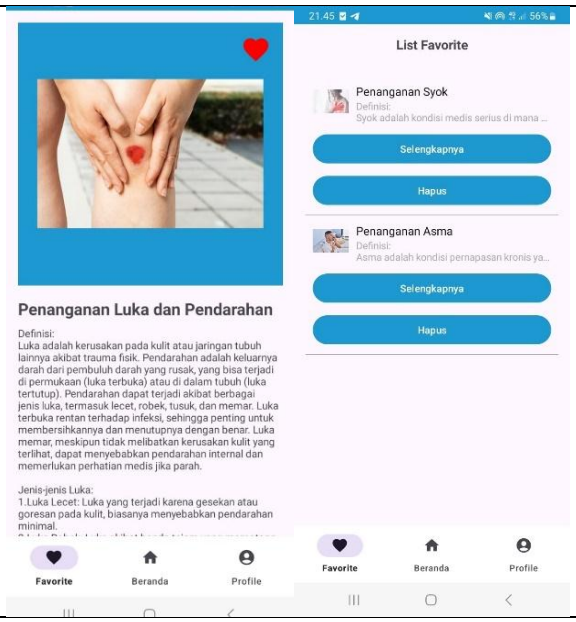
Tabel 4. 2 *Black Box* testing tambah user

Tes Faktor	Hasil	Keterangan
Admin mengisi form tambah user dan menekan tombol simpan	✓	Berhasil, tampil informasi bahwa simpan data berhasil
Screenshot		
		

c. *Black Box testing favorite*

Tabel 4. 3 *Black Box* testing *favorite*

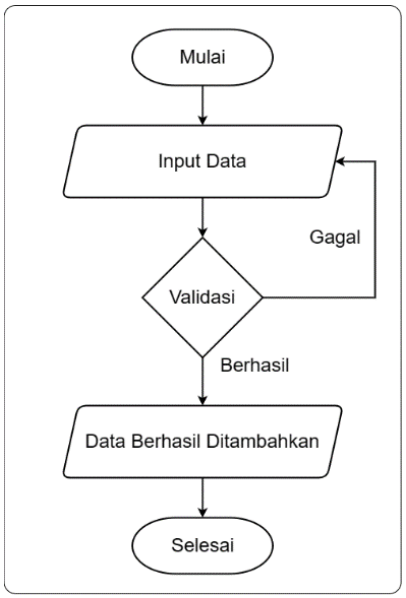
Tes Faktor	Hasil	Keterangan
User menekan ikon sukai di detail penanganan	✓	Berhasil, penangan muncul di *favorite*

Screenshot
Penanganan Luka dan Pendarahan Definisi: Luka adalah kerusakan pada kulit atau jaringan tubuh lainnya akibat trauma fisik. Pendarahan adalah keluarnya darah dari pembuluh darah yang rusak, yang bisa terjadi di permukaan (luka terbuka) atau di dalam tubuh (luka tertutup). Pendarahan dapat terjadi akibat berbagai jenis luka, termasuk lecet, robek, tusuk, dan memar. Luka terbuka rentan terhadap infeksi, sehingga penting untuk membersihkannya dan menutupnya dengan benar. Luka memar, meskipun tidak melibatkan kerusakan kulit yang terlihat, dapat menyebabkan pendarahan internal dan memerlukan perhatian medis jika parah. Jenis-jenis Luka: 1.Luka Lecet: Luka yang terjadi karena gesekan atau goresan pada kulit, biasanya menyebabkan pendarahan minimal. Favorite Beranda Profile 21.45 56% List Favorite Penanganan Syok Definisi: Syok adalah kondisi medis serius di mana ... Selengkapnya Hapus Penanganan Asma Definisi: Asma adalah kondisi pernapasan kronis ya... Selengkapnya Hapus Favorite Beranda Profile

2. *White box*

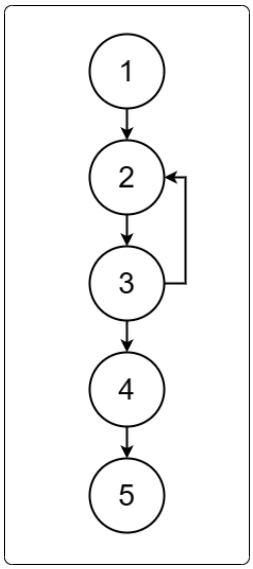
a. *White box testing* tambah P3K

1) Flowchart



Gambar 4. 20 Flowchart tambah P3K

2) Flowgraph



Gambar 4. 21 Flowgraph tambah P3K

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : = E – N + 2

V(G)

E (*edge*) = 5

N (*node*) = 5

P (Predikat = 1

node)

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

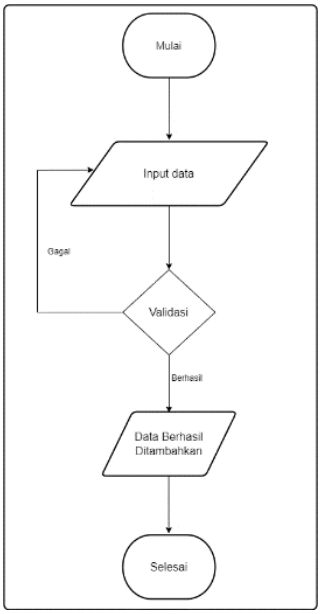
(4) Grafik matriks tambah P3K

Tabel 4. 4 Grafik matriks tambah P3K

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					1 + 1 = 2

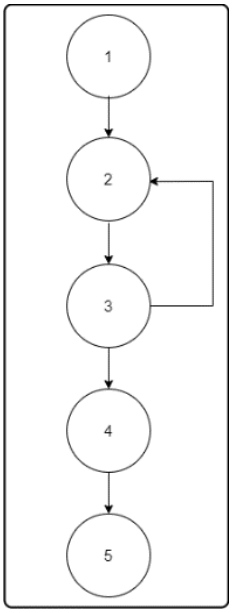
b. *White box testing* tambah user

1) Flowchart



Gambar 4. 22 Flowchart tambah user

2) Flowgraph



Gambar 4. 23 Flowgraph tambah user

Berdasarkan gambar diatas dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (Predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *flowgraph* di atas memiliki *Region* = 2

(3) *Independent path* pada *flowgraph* tersebut yakni:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

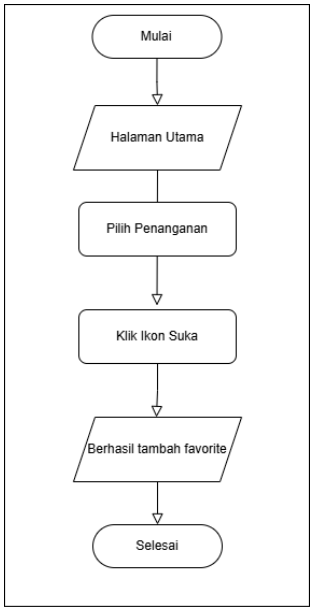
(4) Grafik matriks Tambah User

Tabel 4. 5 Grafik matriks Tambah User

	1	2	3	4	5	E-1
1		1				1 – 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					1 + 1 = 2

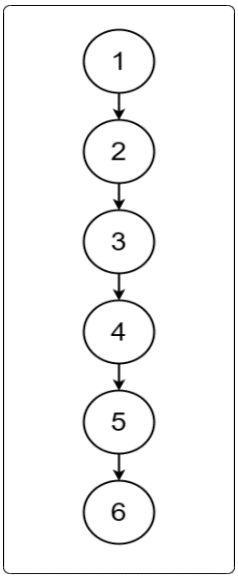
c. *White box testing favorite*

1) Flowchart



Gambar 4. 24 Flowchart *favorite*

2) Flowgraph



Gambar 4. 25 Flowgraph *favorite*

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node.*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 4

N (*node*) = 5

P (Predikat *node*) = 0

Penyelesaian :

V(G) = E – N + 2

= 4 – 5 + 2

= 1

Predikat (P) = P + 1

= 0 + 1

= 1

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 1*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 4 – 5

(4) Grafik matriks *favorite*

Tabel 4. 6 Grafik matriks *favorite*

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3				1		1 – 1 = 0
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					0 + 1 = 1

BAB V

PENUTUP

A. Kesimpulan

Penelitian ini menghasilkan sebuah aplikasi panduan Pertolongan Pertama pada Kegawatdaruratan (P3K) berbasis Android yang dirancang untuk memberikan informasi cepat, akurat, dan mudah diakses oleh masyarakat dalam menangani situasi darurat. Dengan fitur-fitur seperti panduan langkah-langkah P3K, pengelolaan data oleh admin, serta aksesibilitas yang mudah bagi pengguna, aplikasi ini diharapkan dapat meningkatkan kesadaran dan pemahaman masyarakat mengenai pentingnya tindakan P3K yang tepat. Implementasi sistem berbasis Android memungkinkan informasi P3K disajikan secara lebih interaktif melalui teks, gambar, dan video, sehingga pengguna dapat memperoleh pemahaman yang lebih baik dalam menghadapi kondisi darurat.

B. Saran

Untuk pengembangan lebih lanjut, aplikasi ini dapat ditingkatkan dengan fitur tambahan seperti konsultasi langsung dengan tenaga medis, notifikasi darurat, serta integrasi dengan layanan kesehatan setempat. Selain itu, uji coba yang lebih luas dengan melibatkan berbagai kalangan masyarakat perlu dilakukan untuk mengevaluasi efektivitas aplikasi dalam meningkatkan keterampilan P3K. Pengembang juga dapat mempertimbangkan penerapan teknologi kecerdasan buatan (AI) untuk memberikan rekomendasi pertolongan pertama yang lebih personal dan sesuai dengan kondisi pengguna.

DAFTAR PUSTAKA

Anif, M., Ayuningtyas, R.A.M. Dyah, & Nugroho, D.A. (2020). Karakteristik Bahasa Java yang Menjadi Faktor Utama Popularitasnya. Jurnal Sistem Informasi Universitas Negeri Semarang, 17(1), 1–10.

Fitri, R., & Astuti, P. (2022). Pengembangan Aplikasi Mobile Berbasis Android Menggunakan Metode Waterfall. Jurnal Sistem Informasi dan Teknologi Informasi, 16(2), 109-116.

Lestari, D., & dkk. (2022). Firebase: Platform Pengembangan Aplikasi Seluler Inovatif. Jurnal Teknologi Informasi dan Komunikasi, 11(2), 112-123.

Munawar, G., Sibarani, N. S., & Wisnuadhi, B. (2018). Analisis Performa Aplikasi Android pada Bahasa Pemrograman Java dan Kotlin. Dalam Prosiding Industrial Research Workshop and National Seminar. Industrial Research Workshop and National Seminar.

Pratama, Y., & Rachmatika, R. (2023). Aplikasi Edukasi Pertolongan pertama pada kedaruratan (P3K) Berbasis Android. Universitas Pamulang.

Rahmawati, N., et al. (2023). Pengembangan Aplikasi Mobile Panduan P3K dan Petunjuk Arah Sarana Kesehatan Berbasis Android. Universitas Pamulang.

Sari, N. A., Yuliani, D., & Sukmawati, D. (2023). Dampak Pelatihan P3K Terhadap Pengetahuan dan Kemampuan Masyarakat dalam Menangani Kecelakaan di Puskesmas Kumun. Jurnal Kesehatan Tambusi, 5(4), 572-583.

Tari, S., et al. (2023). Rancang Bangun Aplikasi Pembelajaran Pertolongan pertama pada kedaruratan (P3K) Berbasis Android. Universitas Islam Negeri Sunan Kalijaga Yogyakarta.

Voutama, A. (2022). Sistem Antrian Cucian Mobil Berbasis Website Menggunakan Konsep CRM dan Penerapan UML. Komputika: Jurnal Sistem Komputer, 11(1), 102–111. https://doi.org/10.34010/komputika.v11i1.4677

Yanti, N., & Lesmideryati, D. (2022). Pelatihan Pembuatan Aplikasi Android Berbasis Android Studio untuk Guru dan Siswa SMAIT Al Auliya Balikpapan. Prosiding Seminar Nasional Terapan Riset Inovatif (SENTRINOV), 8(3), 114–123.