

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian terdahulu adalah upaya peneliti untuk mencari perbandingan dan selanjutnya untuk menemukan inspirasi baru untuk penelitian selanjutnya disamping itu kajian terdahulu membantu penelitian dapat memposisikan penelitian serta menunjukkan orsinalitas dari penelitian. Agar penelitian ini menjadi lebih terfokus pada suatu masalah penelitian dan dapat menghasilkan kebaruan penelitian, serta memetakan posisi penelitian yang akan dilakukan, maka perlu melakukan pembelajaran terhadap penelitian-penelitian terdahulu yang sejenis dengan tema penelitian yang akan dilakukan. Berdasarkan hal tersebut, peneliti melakukan studi literatur terhadap hasil penelitian terdahulu dan hanya dijabarkan sebagai berikut:

Inggrit, Maruji dan Henny, 2022. Rancang Bangun Aplikasi Panic Button Berbasis Android di Kota Kendari.

Penelitian yang dilakukan oleh Inggrit, Maruji, dan Henny, dilatarbelakangi oleh masalah keterlambatan pemadam kebakaran di Kota Kendari dalam menangani kasus kebakaran yang disebabkan oleh pelapor sering kali tidak menyebutkan alamat lokasi kejadian secara akurat. Peneliti menggunakan metode perancangan sistem melalui analisis dan pemodelan menggunakan *Unified Modelling Language* (UML). Aplikasi *panic button* berbasis android berhasil mempermudah masyarakat dalam mengajukan laporan kebakaran dan membantu petugas pemadam kebakaran untuk menemukan lokasi pelapor lebih akurat dan cepat[2].

Nein Raka Abiyoga, 2021. Aplikasi *Emergency Panic Button* Untuk *Event Festival* Dan Konser Musik.

Pada penelitian masalah utama adalah cara menghubungi petugas keamanan tanpa mengganggu acara dan tanpa perlu mendatangi petugas. Metode penelitian yang digunakan adalah metode *waterfall* yang meliputi analisis masalah, perancangan desain implementasi desain dan pengujian sistem. Pengujian aspek *usability* menggunakan kriteria *learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction*. Penelitian ini berhasil merancang aplikasi yang mampu menghubungkan pengunjung dengan petugas keamanan secara efektif dan cepat tanpa mengganggu jalannya acara[3].

Naviatul Ilma, Agung Dewa Cendana Feoh, dan Wahyu Ahmat Saif, 2021. Aplikasi Keselamatan Berbasis *Smartphone* dengan Integrasi Sinyal Sos Sebagai Solusi Pembentukan Masyarakat Tanggap Bencana.

Penelitian ini menyoroti kurangnya kesiapsiagaan masyarakat dalam menghadapi bencana alam di Indonesia, di mana angka korban bencana sangat tinggi. Penelitian ini menggunakan metode desain aplikasi keselamatan berbasis *smartphone* yang dilengkapi fitur pengiriman sinyal SOS (*Save Our Souls*). Aplikasi ini memanfaatkan teknologi *Location-Based Service* (LBS) yang memungkinkan deteksi lokasi pengguna dan penentuan tempat evakuasi aman. Pengujian melibatkan simulasi aliran sinyal SOS dari pengguna ke *server* serta respon dari tim penyelamat atau pihak berwenang. Fokusnya adalah pada efektivitas aliran informasi dan kesiapsiagaan pengguna dalam situasi darurat. Hasil penelitian menunjukkan bahwa aplikasi yang dirancang mampu meningkatkan kesiapsiagaan masyarakat dengan memberikan panduan lokasi aman dan bantuan dalam situasi darurat[4].

Moch. Rosy Ilhamsyah, 2024. Analisis *Framing* Peringatan Darurat Pada Akun Instagram Narasi.Tv.

Penelitian yang dilakukan oleh Moch. Rosy Ilhamsyah, masalah utamanya adalah bagaimana makna konstruksi postingan peringatan darurat ini memengaruhi persepsi publik terhadap kondisi demokrasi di Indonesia. Penelitian ini menggunakan metode analisis *framing* dengan pendekatan kualitatif deskriptif. Teknik analisis yang digunakan adalah model *framing* William A. Gamson dan Andre Modigliani, yang fokus pada struktur interpretatif yang mencakup "*core frame*" dan "*condensing symbols*". Hasilnya menunjukkan bahwa konstruksi peringatan darurat pada akun instagram narasi.tv dibingkai dengan warna biru dan simbol garuda sebagai tanda darurat demokrasi[5].

Khairul Fadhly dan Nizirwan Anwar, 2023. Rancang Bangun Aplikasi *Emergency Call* Berbasis *Hybrid Mobile*.

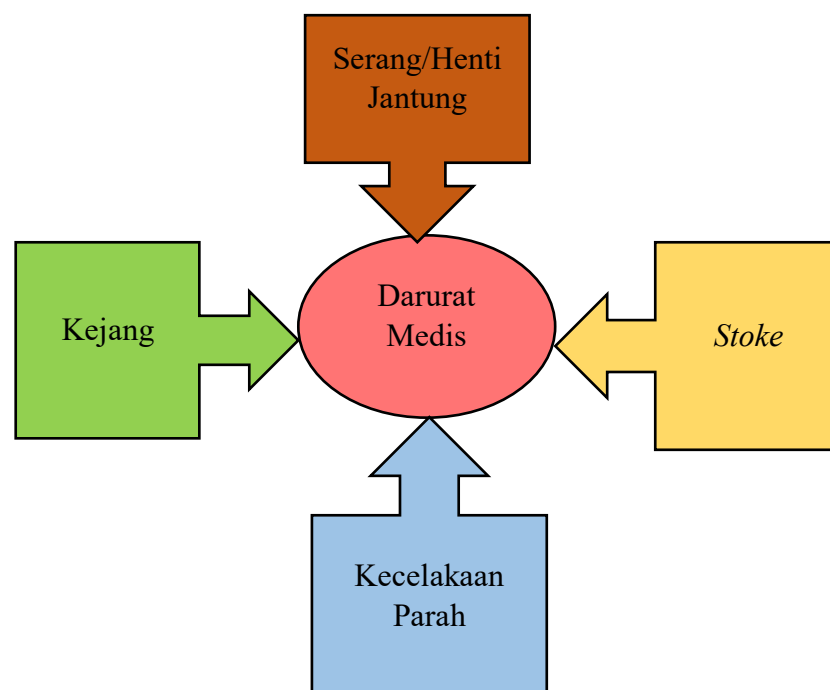
Penelitian ini dilatarbelakangi oleh tingginya tingkat kriminalitas di Kabupaten Tangerang dan kebutuhan masyarakat akan layanan darurat yang cepat dan mudah diakses. Penelitian ini menggunakan metode pengembangan sistem berbasis *Prototype*. Langkah-langkah dalam metode ini meliputi identifikasi kebutuhan, pengembangan, pengujian, dan perbaikan prototipe. Aplikasi ini memungkinkan pengguna mengakses nomor darurat dan mengetahui lokasi layanan publik, seperti kantor polisi, rumah sakit, dan pos pemadam kebakaran di Kabupaten Tangerang [6].

2.2 Landasan Teori

2.2.1 Keadaan Darurat Medis

Keadaan darurat medis adalah situasi kritis yang memerlukan penanganan segera untuk mencegah kematian, kecacatan, atau kerusakan lebih lanjut pada kesehatan seseorang. Keadaan ini melibatkan kondisi seperti serangan jantung,

kecelakaan, pendarahan hebat, atau gangguan pernapasan yang membutuhkan respon cepat dari tenaga medis atau layanan darurat untuk memberikan perawatan yang tepat. Keadaan darurat di Indonesia, seperti yang diatur dalam putusan Mahkamah Konstitusi (MK), mengacu pada situasi kritis yang membutuhkan tindakan cepat dan tepat. Keadaan darurat (*State of Emergency*) memberikan kewenangan kepada pemerintah untuk mengambil langkah-langkah khusus dalam menghadapi ancaman terhadap keamanan, ketertiban, atau stabilitas negara. Langkah-langkah ini dapat mencakup pengetatan hukum, pembatasan hak-hak tertentu, dan penguatan kontrol pemerintah. Dalam konstitusi Indonesia, keadaan darurat diatur dalam Pasal 22 UUD 1945. Undang-Undang Dasar Negara Republik Indonesia Tahun 1945 memberikan Presiden wewenang untuk menetapkan keadaan darurat apabila terdapat ancaman yang mendesak[5]. Contoh darurat medis dapat dilihat dari Gambar 2.1 dibawah ini.



Gambar 2.1 Contoh Darurat Medis

2.2.2 Sistem Peringatan Darurat

Sistem peringatan darurat merupakan seperangkat sistem yang dapat memberikan peringatan kepada masyarakat tentang keadaan yang berbahaya dan dapat membantu mengurangi dampak yang ditimbulkan oleh keadaan berbahaya. Jenis peringatan darurat pada penelitian ini berupa notifikasi seperti pesan yang akan terkirim secara otomatis kepada orang lain yang saling terhubung.

2.2.3 Aplikasi

Aplikasi *mobile* adalah perangkat lunak yang dirancang untuk digunakan pada *smartphone*. Aplikasi ini menawarkan berbagai kemudahan dan dapat diakses oleh siapa saja, kapan saja, dan di mana saja. Oleh karena itu, aplikasi *mobile* menjadi pilihan utama bagi pengguna *smartphone* di era modern saat ini. Dengan banyaknya aplikasi *mobile* yang tersedia, banyak orang memanfaatkannya sebagai dasar untuk membangun *startup*. Dalam perkembangan teknologi yang semakin memudahkan masyarakat, *startup* bukan lagi hal baru, tetapi tetap menjadi topik yang menarik perhatian banyak orang[7]. Penggunaan aplikasi *mobile* berbasis android tidak hanya memudahkan promosi tetapi juga menjadi solusi bagi perusahaan dalam menjangkau pelanggan secara luas serta meningkatkan pendapatan bagi Perusahaan. Ini menunjukkan bahwa inovasi digital dapat meningkatkan daya saing perusahaan di era teknologi [8]. Aplikasi android juga biasa digunakan untuk mendeteksi benda yang hilang seperti mendeteksi pencurian kendaraan [9].

2.2.4 Android

Android adalah platform perangkat lunak *opensource* yang dirancang untuk perangkat *mobile*. Di dalamnya terdapat sistem operasi, *middleware*, dan aplikasi

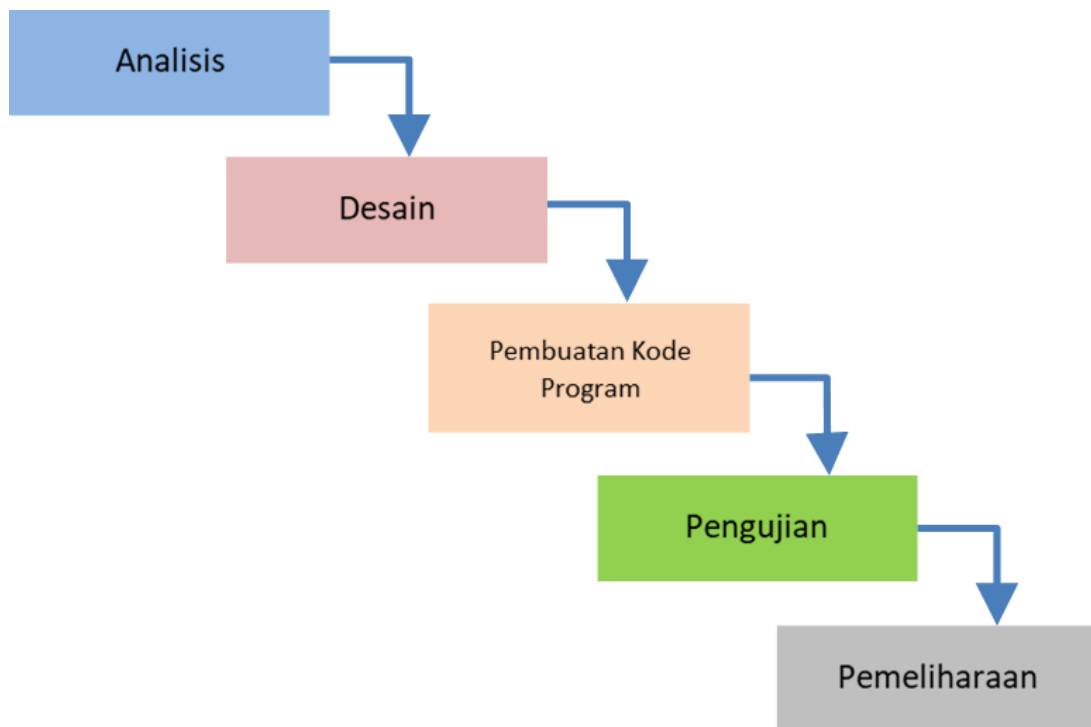
dasar. Android memberikan kesempatan kepada pengembang untuk membuat aplikasi melalui platform terbuka. Sistem operasi android berbasis pada kernel linux 2.6 yang telah disesuaikan untuk kebutuhan perangkat *mobile* [10]. Android *icon* logo dapat dilihat pada Gambar 2.2 berikut ini.



Gambar 2.2 Android

2.2.5 Metode Pengembangan *Waterfall*

Metode pengembangan perangkat lunak merupakan pendekatan sistematis yang digunakan untuk menyusun, merancang, dan mengelola proses pembuatan sistem informasi. Dalam penelitian ini, metode yang digunakan, yaitu metode *Waterfall*, yang merupakan metode pengembangan perangkat lunak dengan pendekatan berurutan, di mana setiap tahapan harus diselesaikan sebelum melanjutkan ke tahap berikutnya [11]. Metode *waterfall* dapat dilihat pada Gambar 2.3 yang ada di bawah. Model *waterfall* juga biasa di definisikan sebagai pendekatan klasik dalam pengembangan perangkat lunak yang dilakukan secara sistematis dan berurutan. Model ini sering dikenal sebagai *classic life cycle* karena mengikuti tahapan pengembangan yang terstruktur dari awal hingga akhir [12]. Berikut tahapan metode *waterfall*:



Gambar 2.3 Metode Pengembangan *Waterfall*

- a. Analisis kebutuhan perangkat lunak yang bertujuan untuk mengidentifikasi dan mendokumentasikan kebutuhan perangkat lunak, termasuk antarmuka pengguna dan dokumen teknis yang dibutuhkan.
- b. Desain yaitu melakukan perancangan perangkat lunak, meliputi struktur data, arsitektur sistem, antarmuka pengguna, dan prosedur pengkodean.
- c. Implementasi kode program (*Code Generation*) yang bertujuan untuk menerjemahkan desain yang telah dibuat menjadi program perangkat lunak. Proses ini menghasilkan kode program yang berfungsi sesuai dengan rancangan sistem yang telah dirumuskan pada tahap desain.
- d. Pengujian (*Testing*), yaitu perangkat lunak diuji dari segi logika dan fungsionalitas untuk memastikan seluruh komponen bekerja sesuai spesifikasi.

- e. Pemeliharaan (*Maintenance*) meliputi langkah-langkah untuk mempertahankan dan mengembangkan sistem agar tetap relevan dengan kebutuhan dan perubahan teknologi. Sistem harus mampu menyesuaikan diri terhadap pembaruan perangkat keras maupun perangkat lunak.

2.2.6 Perancangan Aplikasi

Perancangan aplikasi adalah proses yang mencakup berbagai tahapan untuk menciptakan antarmuka program melibatkan perencanaan desain yang detail, mulai dari menentukan kebutuhan pengguna, merancang alur navigasi, hingga memastikan bahwa setiap elemen antarmuka mendukung pengalaman pengguna yang optimal[13].

Perancangan aplikasi adalah tahap penting dalam pengembangan sebuah alat atau aplikasi, termasuk alat notifikasi keadaan darurat. Pada tahap ini, proses perancangan dimulai dengan mengidentifikasi kebutuhan pengguna berdasarkan hasil analisis yang telah dilakukan sebelumnya. Analisis ini bertujuan untuk memastikan bahwa alat yang dirancang benar-benar memenuhi kebutuhan pengguna secara spesifik. Merancang aplikasi ini mempertimbangkan fitur-fitur yang digunakan, melibatkan pemilihan teknologi yang tepat serta memastikan alat dapat berfungsi dengan optimal di berbagai kondisi darurat yang mungkin terjadi [14].

2.2.7 Basis Data

Basis data adalah sistem yang terdiri dari elemen-elemen penting, yaitu *database*, *file*, *entity*, dan *record*, yang saling berhubungan untuk menyimpan dan mengelola informasi. Dalam pengelolaan basis data, bahasa yang digunakan, seperti SQL, dBase, atau QUEL, memainkan peran penting. Bahasa basis data ini dibagi menjadi dua bentuk, yaitu *Data Definition Language (DDL)* yang digunakan untuk

mendefinisikan struktur basis data dan *Data Manipulation Language* (DML) yang digunakan untuk memanipulasi dan mengambil data dari basis data. Berikut definisi dari 4 elemen penting dalam basis data[15].

a. *Entity*

Entity merupakan representasi dari orang, tempat, kejadian, atau konsep yang informasinya disimpan dalam basis data. Contohnya, dalam konteks lalu lintas, entity mencakup kemacetan, kecelakaan, dan sebagainya.

b. *Atribut*

Setiap *entity* memiliki atribut, yaitu karakteristik atau elemen yang mewakili entity tersebut. Dalam kasus lalu lintas, atribut dapat berupa nama objek, alamat, jenis objek, dan lain-lain. Atribut juga dikenal sebagai elemen data, *field* data, atau item data.

c. *Data Value*

Data value adalah nilai aktual atau informasi yang disimpan dalam setiap elemen data atau atribut.

d. *Database*

Database adalah kumpulan *field* yang saling terkait, yang membentuk struktur data untuk menyampaikan informasi tertentu, seperti kondisi lalu lintas dalam format yang dapat dipahami.

e. *File*

File adalah kumpulan *record* yang seragam dalam elemen dan atributnya, meskipun nilai datanya berbeda-beda.

f. Record

Record adalah kumpulan elemen data yang saling berhubungan untuk menggambarkan *entity* secara lengkap. Setiap *record* mewakili satu informasi atau data tertentu.

2.2.8 Perancangan Basis Data

Database adalah kumpulan data yang terstruktur dan terorganisasi secara sistematis, disimpan secara elektronik, dan dapat mencakup berbagai jenis informasi seperti teks, angka, gambar, video, atau file lainnya. Pengelolaan database dilakukan menggunakan perangkat lunak Sistem Manajemen Basis Data (DBMS), yang memungkinkan proses penyimpanan, pengambilan, pengeditan, dan pengelompokan data. Database dirancang untuk mencegah duplikasi data, menghindari ketidaksesuaian, mempermudah akses dan pembaruan data, serta memastikan kualitas dan data sesuai dengan informasi yang dimasukkan. Penelitian terkait perancangan basis data yang telah dilakukan, menunjukkan pentingnya pemodelan basis data yang mampu meningkatkan pengelolaan informasi perpustakaan. Perancangan basis data pada penelitian ini menggunakan SQLite.

Salah satu penelitian oleh Fitriyani tahun 2019, berjudul "*Perancangan E-Katalog pada Perpustakaan Digital STT-PLN Berbasis Web*" memanfaatkan MySQL bersama *Physical Data Model* dan *Entity Relationship Diagram* (ERD). Penelitian ini menghasilkan sistem informasi perpustakaan digital di STT-PLN melalui pendekatan basis data yang terstruktur. Selanjutnya, penelitian Mailasari tahun 2019, dalam "*Sistem Informasi Perpustakaan Menggunakan Metode Waterfall*" menghasilkan sistem informasi yang tidak hanya mempermudah pengelolaan data perpustakaan

tetapi juga meningkatkan layanan perpustakaan. Dalam penelitiannya, ERD diterapkan sebagai langkah awal perancangan basis data [16].

2.2.9 Implementasi

Implementasi aplikasi berbasis Android melibatkan proses perancangan, pengujian, dan penerapan perangkat lunak pada perangkat *mobile*. Alat yang digunakan pada saat pengimplementasian adalah android studio, figma, *smartphone* dan laptop. Tujuan dari implementasi untuk menerapkan program aplikasi yang telah dirancang sebelumnya. Contoh yang diimplementasikan, yaitu: rancangan data base dan rancangan antar muka [17].

2.2.10 Figma

Figma ialah bagian alat desain yang sering digunakan untuk merancang tampilan aplikasi *mobile*, *desktop*, *website*, dan lainnya. Alat ini dapat diakses melalui sistem operasi *Windows*, *Linux*, atau *Mac* selama terhubung dengan internet. Salah satu keunggulan figma adalah kemampuannya mendukung kolaborasi, di mana beberapa orang dapat bekerja pada proyek yang sama secara bersamaan meskipun berada di lokasi yang berbeda. Hal ini memungkinkan kerja tim yang efisien, menjadikan figma pilihan utama bagi banyak desainer UI/UX untuk membuat prototipe *website* atau aplikasi dengan cepat dan efektif. *User Experience* (UX) sendiri adalah sistem yang mengatur bagaimana pengguna merasakan pengalaman saat menggunakan perangkat lunak, dengan fokus pada tingkat kemudahan dan kenyamanan fungsi perangkat lunak tersebut [18]. Logo figma dapat dilihat dari Gambar 2.4 yang ada di bawah ini.



Gambar 2.4 Figma

2.2.11 SQLite

SQLite adalah sebuah *library* yang berfungsi sebagai mesin database yang bersifat *self-contained*, *serverless*, *zero-configuration*, dan *transactional*. *Self-contained* berarti SQLite hanya memerlukan sedikit dukungan dari *library* eksternal atau sistem operasi. *Serverless* menunjukkan bahwa SQLite dapat mengakses database secara langsung, baik untuk membaca maupun menulis, tanpa melalui proses *server*, sehingga tidak mendukung pengaksesan jarak jauh melalui jaringan komputer seperti intranet atau internet. Sebagian besar mesin SQL lainnya biasanya diterapkan sebagai proses *server* terpisah. *Zero-configuration* menandakan bahwa SQLite dapat digunakan tanpa perlu instalasi sebelumnya. Sebagai database *transactional*, SQLite mendukung transaksi dengan menerapkan prinsip ACID (*Atomic*, *Consistent*, *Isolated*, *Durable*) untuk menjaga konsistensi data saat *query* dijalankan. Meskipun praktis, SQLite memiliki kekurangan, seperti membutuhkan instalasi untuk memproses file database dan memori tambahan pada perangkat Android karena data disimpan secara lokal (tidak online). Berbeda dengan MySQL, yang menggunakan database berbasis online atau jaringan lokal dengan model *client-server* [19]. Logo SQLite dapat dilihat dari Gambar 2.6 yang ada di bawah ini.



Gambar 2.5 SQLite

2.2.12 Native

Aplikasi Native adalah jenis aplikasi yang dirancang secara khusus untuk sistem operasi tertentu pada perangkat bergerak. Karakteristik utamanya mencakup kemampuan untuk mengakses perangkat keras secara langsung dan mendukung fitur antarmuka pengguna serta interaksi yang disediakan oleh lingkungan sistem operasi perangkat tersebut [20]. Native android *development* menggunakan Java mengacu pada proses pengembangan aplikasi android dengan bahasa pemrograman Java, yang merupakan salah satu bahasa utama yang didukung oleh Android SDK. Java memungkinkan pengembang untuk merancang antarmuka pengguna yang dinamis, mengelola aktivitas dalam aplikasi, dan meningkatkan kinerja secara optimal. Android Studio sebagai IDE (*Integrated Development Environment*) utama menyediakan alat yang membantu pengembang menggabungkan Java dengan berbagai pustaka untuk menghasilkan aplikasi yang efisien dan berkinerja baik.

2.2.13 XML

XML (*eXtensible Markup Language*) adalah bahasa markup yang menggunakan serangkaian aturan untuk mendefinisikan sintaks dalam

mendeskripsikan teks atau data dalam dokumen menggunakan tag. XML dirancang untuk menyimpan data secara ringkas dan mudah diatur. Fokus utama XML adalah pada data, yang dapat diolah untuk menghasilkan informasi yang berguna [21]. XML (*Extensible Markup Language*) adalah bahasa markup yang digunakan untuk menyimpan dan mengangkut data. XML memungkinkan pengguna untuk mendefinisikan *tag* dan struktur dokumen mereka sendiri, yang mudah disesuaikan dalam menyajikan berbagai jenis data. Tujuan utama XML adalah menyediakan format yang terstruktur untuk pertukaran data antar sistem yang berbeda, memastikan keutuhan dan kesesuaian data, serta mempermudah proses pengolahan dan pemrosesan informasi. XML bekerja dengan mendefinisikan elemen-elemen dalam dokumen menggunakan *tag* yang dapat dibaca oleh manusia dan mesin. Data disusun secara terstruktur, dengan elemen-elemen yang dibatasi oleh *tag* pembuka dan penutup, yang membuatnya mudah diproses dan terhubung dengan aplikasi lain.

2.2.14 Android Studio

Android studio adalah *integrated development environment* (IDE) resmi untuk pengembangan android yang dirilis oleh *google* dan dikembangkan berdasarkan IntelliJ IDEA (*Integrated Development Environment Application*). Sebelumnya, pengembangan aplikasi android menggunakan IDE (*Integrated Development Environment*) resmi bernama Eclipse, namun sejak diperkenalkannya android studio, *google* telah menjadikannya sebagai IDE utama. Dengan perubahan ini, *google* menghentikan dukungan untuk ADT (*Android Development Tools*) di Eclipse, dan ADT kini hanya tersedia untuk android studio. Android studio dipilih karena menawarkan banyak fitur yang memudahkan proses pengembangan aplikasi,

khususnya bagi programmer pemula. Selain itu, IDE (*Integrated Development Environment*) ini menyediakan berbagai *library* siap pakai yang mempercepat proses pembuatan aplikasi. Meskipun android studio membutuhkan memori yang cukup besar, kelebihanannya, seperti kemudahan penggunaan dan fitur yang lengkap, mampu menutupi kekurangan tersebut. Saat ini, android studio telah mencapai versi 2.1 yang dilengkapi dengan berbagai fitur canggih untuk mendukung pengembangan aplikasi android [22]. Logo android studio dapat dilihat dari Gambar 2.7 yang ada di bawah ini.



Gambar 2.6 Android Studio

2.2.15 Bahasa Pemrograman Java

Bahasa pemrograman, atau yang sering disebut bahasa komputer, adalah sistem instruksi standar yang digunakan untuk mengendalikan perangkat komputer. Bahasa ini terdiri dari aturan sintaksis dan semantik yang digunakan untuk mendefinisikan program komputer. Dengan bahasa pemrograman, seorang programmer dapat menentukan jenis data yang akan diproses, cara penyimpanan dan transfernya, serta tindakan yang harus dilakukan pada berbagai kondisi. Salah satu bahasa pemrograman yang digunakan dalam penelitian ini adalah java. Diperkenalkan

pada pertengahan tahun 1990-an oleh SUN *Microsystems*, java adalah sebuah bahasa yang digunakan untuk membangun dan menjalankan perangkat lunak dalam berbagai lingkungan, baik komputer independen maupun jaringan [23].

Java adalah bahasa pemrograman tingkat tinggi yang dirancang untuk dapat dijalankan di berbagai *platform* dan sifatnya yang sederhana, di mana pengembang dapat dengan mudah mengimplementasikan aplikasi menggunakan sintaks yang jelas dan mudah dipahami. Selain itu, Java juga memiliki arsitektur yang netral terhadap *platform*, artinya program yang ditulis dalam Java dapat dijalankan di berbagai sistem operasi tanpa memerlukan modifikasi kode. Java juga berorientasi objek, yang memungkinkan pengembang untuk mengorganisir kode atau memudahkan pemeliharaan serta pengembangan lebih lanjut. Java dapat berjalan di berbagai platform perangkat keras dan perangkat lunak, seperti *Microsoft Windows*, *Linux*, *Solaris*, dan *Mac OS*. Ini dimungkinkan oleh dua komponen utama dalam *platform* Java, yaitu *Java Virtual Machine* (JVM) yang bertanggung jawab untuk menjalankan kode Java, serta *Java Application Programming Interface* (API) yang menyediakan serangkaian pustaka yang mempermudah pengembangan aplikasi [24]. Logo bahasa pemrograman java dapat dilihat dari Gambar 2.8 yang ada di bawah ini:



Gambar 2.7 Java

2.2.15 Pengujian

Pengujian merupakan tahap utama dalam proses pengembangan perangkat lunak, karena kegagalan dalam pelaksanaan pengujian dapat berdampak signifikan, termasuk mengakibatkan kerugian finansial yang besar, menurunkan kualitas produk, memengaruhi reputasi perusahaan, dan meningkatkan risiko kegagalan sistem saat digunakan oleh pengguna. Dalam melaksanakan pengujian, penting untuk memilih teknik yang sesuai, yaitu metode yang tepat dalam mendeteksi kesalahan atau *bug* yang sebelumnya tidak teridentifikasi. Pemilihan teknik yang tepat akan membantu memastikan peningkatan kualitas perangkat lunak secara keseluruhan. Adapun metode yang diusulkan untuk menguji kualitas aplikasi ini adalah metode *Black Box* [25].

a. *Black Box Testing*

Pengujian *Black Box* adalah metode pengujian yang bertujuan untuk memverifikasi hasil eksekusi aplikasi berdasarkan data uji yang diberikan, guna memastikan bahwa fungsionalitas aplikasi sesuai dengan persyaratan (*requirement*). Pengujian ini berfokus pada antarmuka (*interface*) dan pengujian fungsional aplikasi, serta memastikan kesesuaian alur fungsi yang dibutuhkan oleh pengguna.

Metode ini tidak melibatkan pengujian terhadap *source code* program. Tahapan pengujian *Black Box* meliputi:

1. Membuat *test case* untuk menguji fungsi-fungsi dalam aplikasi.
2. Membuat *test case* untuk memeriksa kesesuaian alur kerja suatu fungsi dengan kebutuhan dan permintaan pengguna.
3. Mengidentifikasi *bug* atau *error* berdasarkan tampilan antarmuka aplikasi.

b. UAT

User Acceptance Test (UAT) adalah proses pengujian yang dilakukan untuk memastikan bahwa aplikasi yang dikembangkan telah memenuhi kebutuhan dan persyaratan pengguna. Adapun contoh *kuesioner* dapat dilihat dalam Tabel 2.1 berikut ini.

Tabel 2.1 Contoh *Kuesioner* UAT

NO	PERNYATAAN	SB	B	C	K	SK
1.	Analisis kebutuhan aplikasi telah diselesaikan sebelum pengembangan dimulai.					
2.	Desain antarmuka pengguna telah ditentukan berdasarkan kebutuhan pengguna.					
3.	Pengembangan aplikasi dilakukan sesuai dengan desain yang telah disetujui.					
4.	Proses pengujian dilakukan setelah tahap pengembangan selesai.					
5.	Setiap bug yang ditemukan selama pengujian dicatat dan diperbaiki.					
6.	Pengguna dapat mengakses aplikasi setelah berhasil melakukan <i>login</i> menggunakan username dan <i>password</i> untuk penggunaan pertama kali.					
7.	Pengguna dapat mengaktifkan aplikasi dengan perintah suara yang memudahkan mereka dalam keadaan darurat tanpa harus menavigasi antarmuka.					
8.	Aplikasi mengirimkan notifikasi darurat ke kontak yang telah ditentukan.					
9.	Menurut saya, aplikasi ini mudah digunakan dalam situasi darurat.					
10.	Aplikasi siap untuk diluncurkan atau digunakan oleh pengguna.					

Varibel:

- SB = Sangat Baik (Skor 5)
 B = Baik (Skor 4)
 C = Cukup (Skor 3)
 K = Kurang (Skor 2)
 SK = Sangat Kurang (Skor 1)

Rumus:

Skor Maksimal = Jumlah Responden x Nilai Tertinggi

Skor Minimal = Jumlah Responden x Nilai Terendah
 Total Skor = (Responden Sk x 1) + (Responden K x 2) ... + (Responden SB x 5)

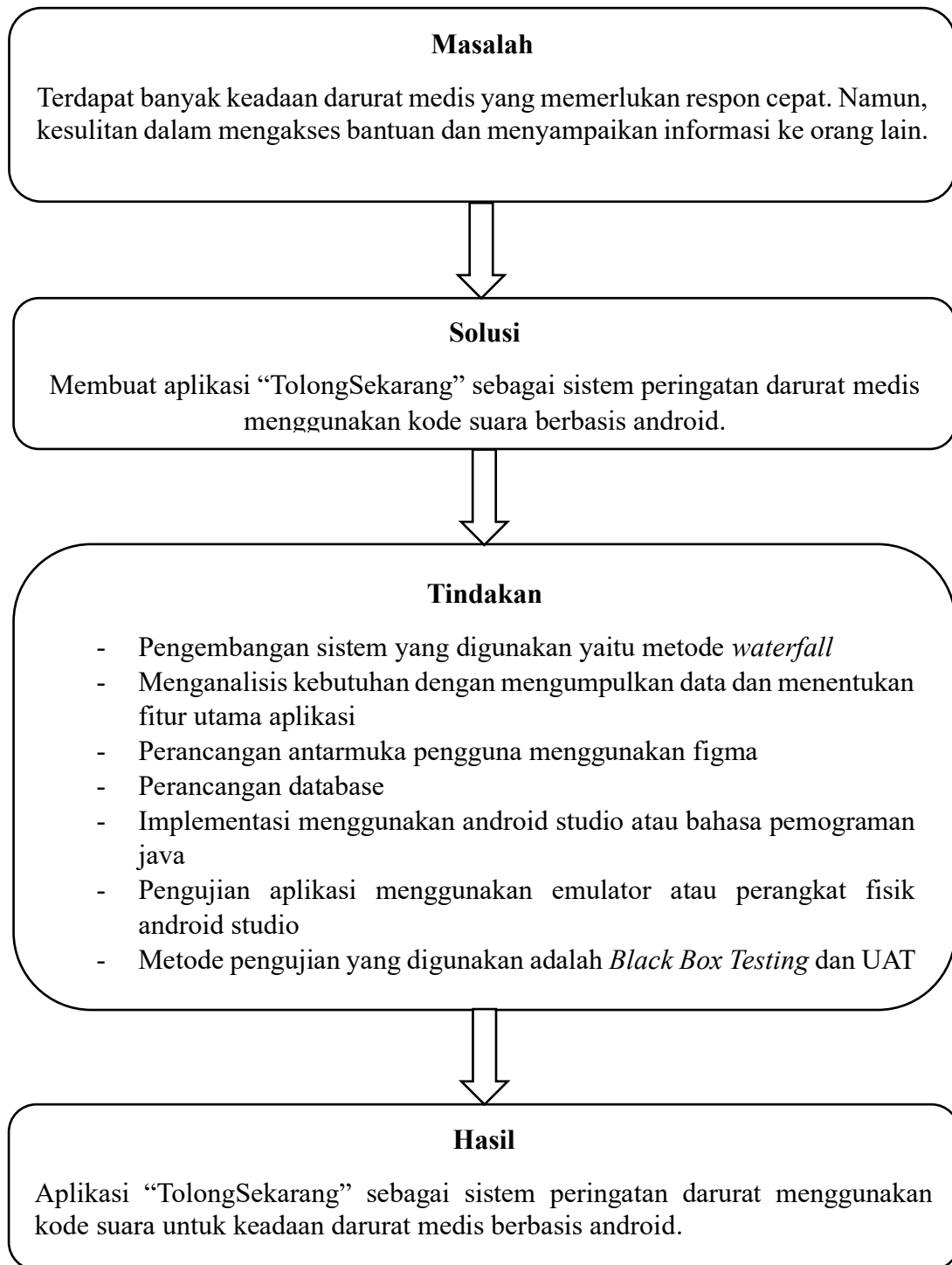
$$\text{Indeks(\%)} = (\text{Total skor} / \text{Skor maksimal}) \times 100\%$$

Predikat Penilaian:

0% - 19% = Sangat Kurang
 20% - 39% = Kurang
 40% - 59% = Cukup
 60% - 79% = Baik
 80% - 100% = Sangat Baik

2.3 Kerangka Pikir

Kerangka berpikir penelitian ini ditunjukkan pada gambar di bawah ini:



Gambar 2.9 Kerangka Pikir