**PROPOSAL SKRIPSI**

**PENGEMBANGAN APLIKASI MOBILE BERBASIS ANDROID MENGGUNAKAN PROTOKOL** ***MESSAGE QUEUE TELEMETRY TRANSPORT* (MQTT) UNTUK SISTEM MONITORING KETINGGIAN BANJIR**

**Oleh:**

**NURUL IZZAH**

**1809075036**



**FAKULTAS TEKNIK**

**UNIVERSITAS MULAWARMAN**

**SAMARINDA**

**2022**

PROPOSAL SKRIPSI

**PENGEMBANGAN APLIKASI MOBILE BERBASIS ANDROID MENGGUNAKAN PROTOKOL *MESSAGE QUEUE TELEMETRY TRANSPORT* (MQTT) UNTUK SISTEM MONITORING KETINGGIAN BANJIR**

Diajukan sebagai salah satu syarat untuk menyelesaikan pendidikan pada Program Studi Strata 1 Teknik Elektro, Fakultas Teknik, Universitas Mulawarman

**Oleh:**

**NURUL IZZAH**

**1809075036**



**FAKULTAS TEKNIK**

**UNIVERSITAS MULAWARMAN**

**SAMARINDA**

**2022**

PROPOSAL SKRIPSI

**PENGEMBANGAN APLIKASI MOBILE BERBASIS ANDROID MENGGUNAKAN PROTOKOL *MESSAGE QUEUING TELEMETRY TRANSPORT* (MQTT) UNTUK SISTEM MONITORING KETINGGIAN BANJIR**

Oleh:

Nurul Izzah

1809075036

Telah dibahas dalam Rapat Dosen Pembimbing pada 20 Juni 2022 dan dinyatakan memenuhi syarat sebagai Skripsi, dengan Dosen Pembimbing :

1. Dr. Eng. Ir. Aji Ery Burhandenny, S.T., M.AIT.
2. Fatkhul Hani Rumawan, S.T., M.T.

Ketua Program Studi S1 Teknik Elektro

Fakultas Teknik, Universitas Mulawarman,

**Ir. Muslimin, S.T., M.T., IPM., ASEAN Eng.**

**NIP. 19760331 200812 1 001**

# Judul

Pengembangan Aplikasi *Mobile* Berbasis Android Menggunakan Protokol *Message Queue Telemetry Transport* (MQTT) Untuk Sistem Monitoring Ketinggian Banjir.

# Latar Belakang

Kalimantan Timur merupakan salah satu provinsi di Indonesia yang memiliki hutan tropis lembab dengan intensitas curah hujan yang tinggi, yaitu dapat mencapai 2.000 mm per tahun, bahkan curah hujan di hutan tropis lembab Kalimantan Timur rata-rata lebih dari 2.400 mm/tahun. Hal ini sejalan dengan Pola Ilmiah Pokok (PIP) Universitas Mulawarman sebagai kampus *Tropical Studies* dengan tema Hutan Tropika Basah dan Lingkungannya yang di kenal memiliki suhu, kelembaban serta curah hujan yang rata-rata tinggi sehingga dari Pola Ilmiah Pokok tersebut ditransformasikan dalam bentuk pengembangan aplikasi yang sejalan dengan pamaknaan pola ilmiah pokok di Fakultas Teknik sebagai Fakultas *Tropical Engineering*, sehingga penulis memanfaatkan teknologi yang ada untuk dilakukan pengembangan dalam bidang komputasi yang berkaitan dengan kawasan tropis yang disesuaikan dengan kondisi yang ada.

Melalui Badan Pusat Statistik (BPS) Provinsi Kalimantan Timur mencatat curah hujan tertinggi di Kota Samarinda pada tahun 2019 adalah 253,3 mm per bulan dan tahun 2020 adalah 269,1 mm per bulan, dengan total curah hujan pada tahun 2019 adalah 1.358,1 mm per tahun dan total curah hujan tahun 2020 adalah 1.970 mm per tahun. Berdasarkan data tersebut dapat disimpulkan bahwa kenaikan curah hujan yang tinggi menjadi salah satu penyebab utama banjir di Samarinda.

Berbagai macam upaya yang dilakukan oleh Pemerintah Kota Samarinda dalam mengendalikan banjir diantaranya pemeliharaan saluran drainase kota, pembenahan sungai-sungai yang melintasi kota, dan berbagai studi terkait pengendalian banjir, tapi masalah banjir belum juga teratasi sampai sekarang, karena masih ditemukan titik genangan air di beberapa bagian jalan Kota Samarinda ketika musim penghujan datang.

Informasi peringatan dini bencana banjir umumnya masih dilakukan secara manual. Hal Ini mengakibatkan informasi kepada masyarakat maupun relawan untuk penanggulangan bencana terlambat. Penelitian ini bertujuan untuk melengkapi penelitian sebelumnya, yaitu Implementasi Sistem Monitoring Sensor Jarak, Suhu, Kelembapan dan *Rain Drop* Berbasis *Internet Of Things* (IoT) Menggunakan *Long Range* (LORA), pada penelitian tersebut menghasilkan data monitoring dari beberapa sensor yang berbeda, dimana data tersebut tersimpan di LORIOT *cloud* *server*, dari data penelitian tersebut maka dilakukan pengembangan aplikasi *mobile* berbasis android untuk melakukan monitoring ketinggian banjir dengan mengambil data dari LORIOT *cloud* *server* untuk ditampilkan ke aplikasi *mobile* dengan menerapkan protokol komunikasi *Message Queuing Telemetry Transport* (MQTT) untuk mengakses data dari perangkat *Internet Of Things* (IoT) melalui *platform* LORIOT *cloud server*. Pada pengembangan aplikasi monitoring ini menggunakan *Application Programming Interface* (API) untuk menyampaikan data *request* dari pengguna pada *server* sehingga data monitoring banjir dapat dikirimkan ke aplikasi *mobile.*

# Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka rumusan masalah dari penelitian ini adalah sebagai berikut :

1. Belum adanya aplikasi *mobile* yang menyajikan informasi secara *real time* mengenai ketinggian banjir di Samarinda ?
2. Bagaimana merancang dan mengembangkan aplikasi monitoring banjir berbasis android menggunakan protokol komunikasi MQTT ?
3. Bagaimana mengimplementasikan *Firebase* *real time* *database* untuk monitoring ketinggian banjir pada aplikasi *native* android?

# Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Bagi Penulis

Penulis dapat mengaplikasikan ilmu yang didapatkan selama perkuliahan dalam bentuk pengembangan aplikasi berbasis android untuk monitoring banjir di wilayah Samarinda sehingga bermanfaat dalam memberikan informasi kepada masyarakat akan resiko banjir.

1. Bagi Universitas

Dapat dijadikan sebagai referensi penelitian bagi mahasiswa Universitas Mulawarman yang ingin mengembangkan aplikasi monitoring yang serupa dengan menggunakan *platform* Android dan bahasa pemrograman Java.

1. Bagi Pengguna

Dapat mempermudah pengguna untuk tanggap bencana, waspada dan siap siaga ketika terjadi banjir sehingga meminimalisir dampak dan kerugian yang diakibatkan oleh banjir, serta masyarakat bisa melakukan tindakan penyelamatan lebih awal sebelum banjir terjadi.

# Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang dan membuat *user interface* aplikasi Android yang dapat melakukan monitoring ketinggian banjir dari jarak jauh secara *real time.*
2. Merancang *back end* aplikasi yang dapat mengakses data dari LORIOT dengan memanfaatkan protokol komunikasi MQTT.
3. Mengimplementasikan *Firebase* *real time* *database* untuk monitoring ketinggian banjir pada aplikasi *mobile* berbasis Android.
4. Menguji performansi sistem menggunakan metode *Load testing* dengan menggunakan *tools* JMeter.
5. Menguji kompatibilitas aplikasi dengan menggunakan beberapa *device* yang berbeda spesifikasinya.
6. Menguji data *publish* dan *subscribe* untuk mengetahui kesesuaian data yang dikirim dan diterima.

# Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah sebagai berikut :

1. Visualisasi daerah peta pada aplikasi menampilkan sesuai lokasi *node* IoT diletakan.
2. Data yang dimonitoring adalah data status, data ketinggian air, dan lokasi.
3. Penelitian ini menggunakan data penelitian sebelumnya yang sudah tersimpan di LORIOT *cloud server*.
4. Pengujian yang dilakukan adalah pengujian performansi pada *device* aplikasi, dan pengujian data *publish* dan *subscribe* dari LORIOT *cloud server* ke aplikasi Android.
5. Pengembangan *user interface* dan *Back end* aplikasi menggunakan Android Studio dengan bahasa pemrograman Java (*full* *native*).
6. Menggunakan *Firebase* sebagai *database*.
7. Tidak membahas tentang keamanan aplikasi.

# Tinjauan Pustaka

## Aplikasi Native

Aplikasi *native* yaitu aplikasi yang dibangun khusus yang ditujukan untuk *operating system* tertentu dengan bahasa pemrograman sesuai *platform* aplikasi tersebut. Misalnya, aplikasi Android *native* ditulis menggunakan bahasa pemrograman Java dengan menggunakan *tools Eclipse*, sementara aplikasi iOS/iPhone dibuat menggunakan bahasa *Objective*-C dan *tools* Xcode. Aplikasi *native* tidak bisa digunakan pada *multiplatform* seperti penggunaan *framework* lainnya sehingga pengembangan aplikasi ini hanya berfokus pada satu *platform* saja yaitu Android (Liu, 2013).

## Android Studio

Android Studio merupakan *integrated development* (IDE) resmi untuk pengembangan aplikasi android, yang ditetapkan IntelliJ IDEA. Android Studio adalah perangkat lunak IDE untuk mengembangkan aplikasi Android. Android Studio dibangun oleh IntelliJ IDEA yang merupakan perangkat lunak pengembangan terintegrasi dengan bahasa pemrograman Java / Semua fitur yang ada di IntelliJ IDEA juga dihadirkan di Android Studio untuk menambah banyak fitur dalam rangka meningkatkan produktivitas para pengembang aplikasi Android, diantaranya yaitu:

1. Sistem *build* Gradle yang *fleksibel*.
2. *Emulator* Android yang memiliki banyak fitur.
3. Mendukung untuk membangun aplikasi Android dengan menggunakan *smartphone*, Android TV, Android Wear, dan perangkat Android lainnya.
4. *Instant Run* yang dapat melakukan perubahan pada aplikasi tanpa harus menginstal ulang aplikasi.
5. Template kode dan integrasi GitHub untuk membantu pengembang membangun aplikasi yang bisa di akses oleh publik dan mengimpor kode sampel.
6. Bermacam - macam *tools* dan kerangka pengujian.
7. Didukung dengan bahasa pemrograman C dan C++.

Android Studio diluncurkan oleh Google pada tanggal 16 Mei 2013 di Konferensi Google I/O untuk menggantikan Eclipse sebagai perangkat lunak IDE resmi untuk *platform* Android. Saat ini Android Studio telah mencapai versi terbaru yaitu Android Studio Chipmunk versi 2021.2.1 Patch 2 pada Agustus 2022 dan bisa diunduh tanpa biaya untuk semua sistem Operasi seperti Windows, macOS dan Linux (DiMarzio, 2016).



**Gambar 1. Aplikasi Mobile**

1. Java Development Kit (JDK)

*Java Development Kit* (JDK) yaitu program untuk pengembang aplikasi untuk menulis applet pada aplikasi Java. JDK terdiri dari kumpulan *runtime* dengan lapisan sistem operasi dan alat sehingga program yang dikembangkan dapat di kompilasi, debug, dan dijalankan pada applet dan aplikasi dapat ditulis dalam bahasa Java.

1. Software Development Kit (SDK)

Pengembangan *software* yang mendukung programer untuk membangun aplikasi pada *platform* khusus Android. Android SDK termasuk proyek yang memiliki sampel dengan kode sumber, pengembangan *software*, emulator, dan *library* yang dibutuhkan untuk membangun sebuah aplikasi Android. Aplikasi ditulis menggunakan bahasa pemrograman Java dan bisa berjalan di Dalvik, virtual mesin yang dirancang khusus untuk digunakan pada Linux kernel.

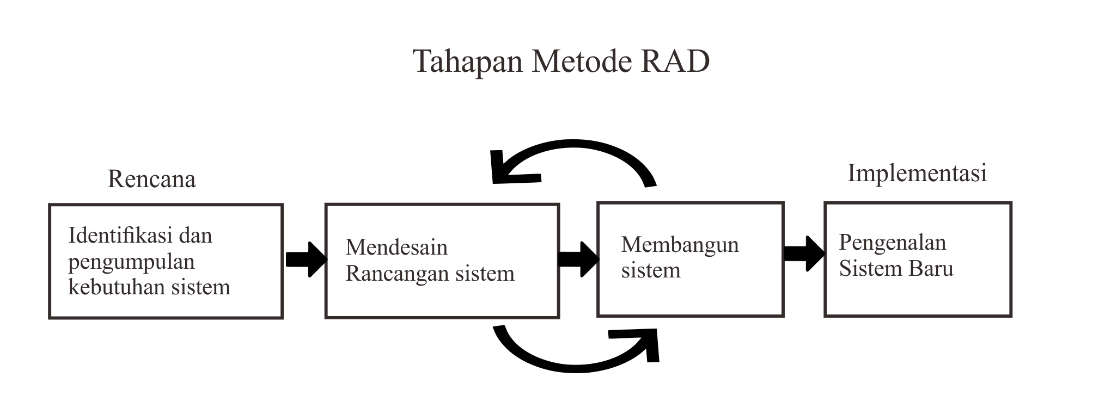
(Burd, 2017).

## Java

Java adalah bahasa pemrograman yang berbasis pada pemrograman berorientasi objek (OOP). Bahasa ini awalnya dikembangkan oleh James Gosling saat masih di Sun Microsystems, pada tahun 2010 Java secara resmi diakuisisi oleh Oracle. Bahasa pemrograman Java dikenal sebagai bahasa yang sederhana, arsitektur berorientasi objek, multithreaded, kuat, dinamis dan aman. *Platform* Java memiliki dua komponen yaitu *Java Virtual Machine* dan *Java Application Programming Interface* (API) (Burd, 2017).

## *Rapid Application Development* (RAD)

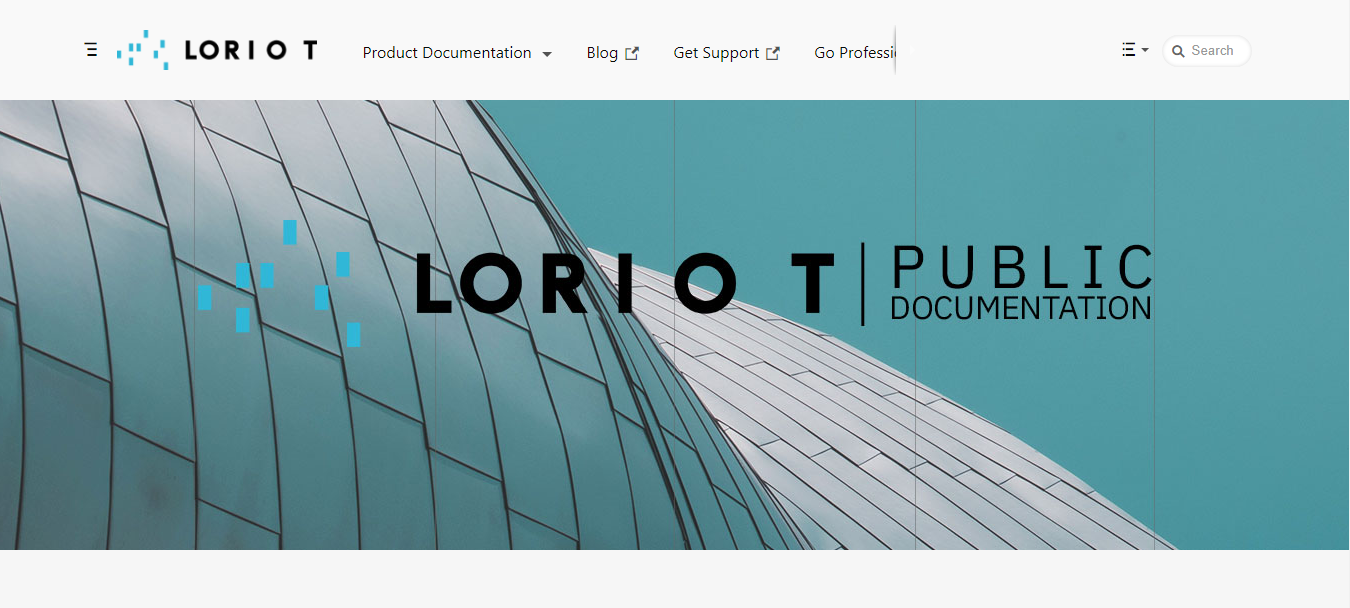
*Rapid Application Development* adalah tahapan pengembangan perangkat lunak yang berfokus pada siklus pengembangan aplikasi secara cepat. Definisi lain menyatakan bahwa metode pengembangan perangkat lunak RAD merupakan metode pengembangan yang menggunakan berbasis orientasi objek untuk pengembangan sistem. Tahapan dalam metode RAD terbagi menjadi 4 tahapan yaitu tahap perencanaan kebutuhan, tahap desain sistem, tahap membangun sistem dan tahap implementasi atau penyelesaian produk (Mukti & Puspita, 2019).



**Gambar 2. Tahapan Metode RAD (Faqih et al,. 2018)**

## LORIOT

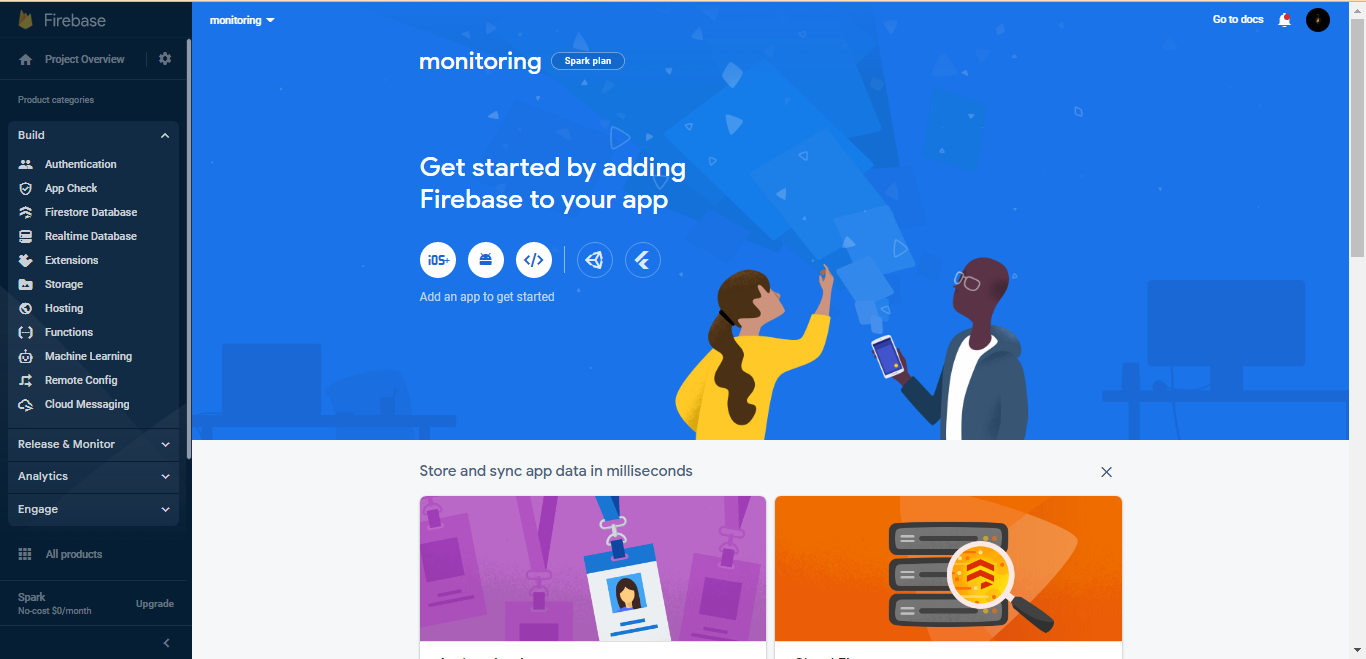
LORIOT merupakan perusahaan IoT global yang menyediakan platform LoRa IoT yang didirikan di Swiss pada tahun 2015, dengan misi menjadi solusi penerapan IoT jangka panjang yang dapat digunakan di seluruh dunia maupun di luar angkasa. LORIOT menyediakan infrastruktur LoRaWAN yang dapat terintegrasi dengan perangkat lunak sehingga mudah digunakan dan memungkinkan pengguna dapat membangun, mengoperasikan, dan menskalakan jaringan IoT dengan aman (Loriot.io, 2022).



**Gambar 3. Platform LORIOT**

## *Firebase*

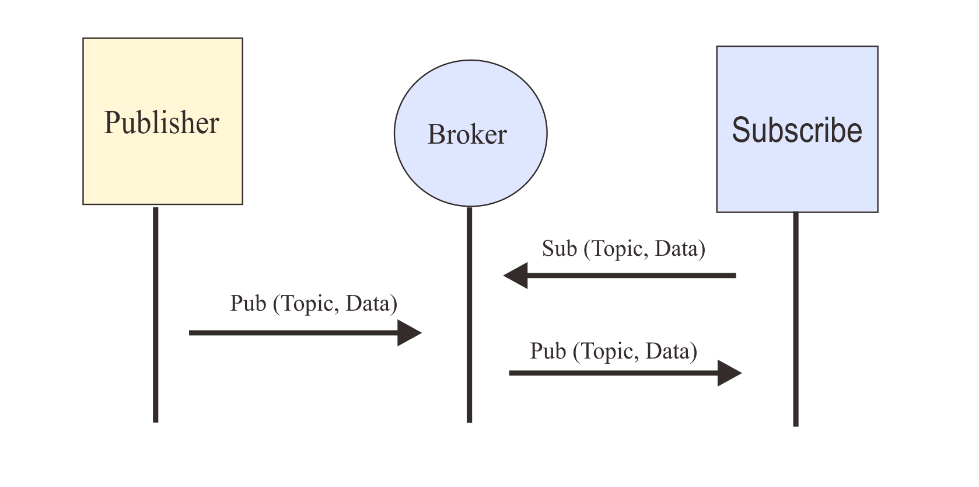
*Firebase* merupakan *database* NoSQL yang disediakan oleh *Firestore*. *Firebase* tidak menggunakan sistem relasi seperti *database* seperti MySQL dll. Metode penyimpanan data di *database* NoSQL menggunakan API yang menggunakan format JSON (JavaScript *Object Notation*). *Firebase* disebut juga sebagai *Cloud-Hosted* yang bisa menyimpan serta menyinkronkan data untuk setiap klien yang terkoneksi. Setiap kali ada data terbaru maka data akan tersimpan di *cloud* dan *mengupdate* data secara otomatis (Moroney, 2017).



**Gambar 4. Database Firebase**

## *Message Queue Telemetry Transport* (MQTT)

MQTT adalah salah satu protokol pertukaran pesan dengan model *publish/subscribe* yang dapat digunakan untuk komunikasi pada sistem komputer dan pada penerapannya *device* akan terhubung pada sebuah broker dan mempunyai topik tertentu. MQTT ini bisa berjalan diatas stack TCP/IP yang dirancang untuk komunikasi mesin ke mesin, atau *device* lainnya yang tidak memiliki alamat khusus (Tigli, 2016).



**Gambar 5. Desain Sistem MQTT (Tigli, 2016)**

## *Unified Modelling Language* (UML)

UML merupakan suatu metode pemodelan secara visual yang digunakan sebagai perancangan sistem berorientasi objek dan juga pendokumentasian sistem aplikasi. UML banyak digunakan untuk menggambarkan kebutuhan, analisis dan desain serta menggambarkan arsitektur dalam pemrograman berorientasi objek (PBO). UML diagram terdapat beberapa jenis yaitu *structure diagram*, *behaviour diagram* dan *interaction diagram* (Rumpe, 2016).

## Google Maps API

Google Maps adalah teknologi milik Google yang digunakan untuk menampilkan peta atau mencari lokasi. Google Maps API (*Application Programming Interface*) Android bisa digunakan oleh pengembang aplikasi untuk menampilkan peta dengan bantuan data dari *Google Maps*. Pengembang juga dapat memanggil API untuk menambahkan marker pada lokasi dari peta wilayah tertentu sesuai keinginan. API ini memungkinkan pengembang untuk menambahkan beberapa fitur seperti:

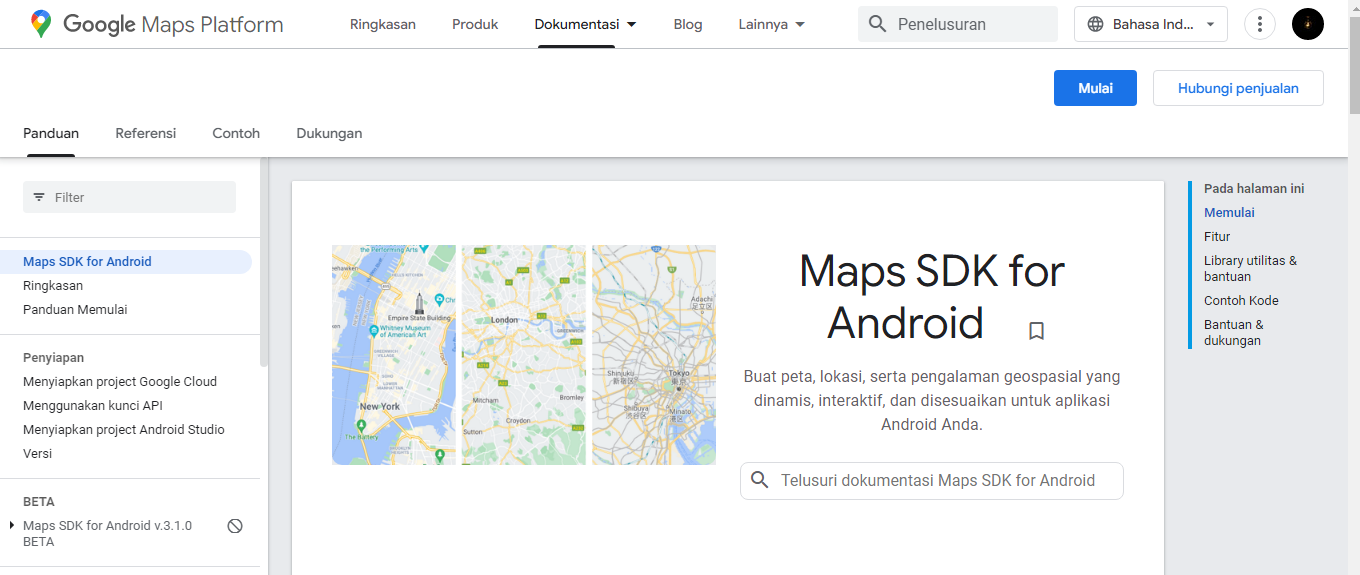
a. Posisi ikon penanda pada area peta atau disebut *marker*

b. Set set segmen garis (*Polylines*)

c. Segmen Tertutup (*Poligon*)

d. Kumpulan gambar yang ditampilkan di atas peta dasar (*Overlay Tile*)

(Dincer & Uraz, 2013)**.**



**Gambar 6. API Google Maps**

## Apache JMeter

Apache JMeter adalah aplikasi *open source* yang dibuat menggunakan 100% Java murni yang digunakan sebagai *tools* untuk melakukan pengujian *load* dan performa pada aplikasi atau *website*. Apache JMeter akan membuat beberapa simulasi pengguna yang akan mengakses *server* dengan jumlah user dan jeda *request* yang berbeda tergantung konfigurasi yang diatur pada Apache JMeter (JMeter, 2022).



**Gambar 7. Aplikasi JMeter**

## Penelitian Terkait

Penelitian ini menggunakan beberapa referensi dalam melakukan pengembangan aplikasi monitoring ketinggian banjir baik dari jurnal, buku, skripsi yang dipublikasi di internet.

Tabel 1. Daftar Referensi Penelitian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nama Peneliti | Judul Penelitian | Metode Penelitian | Kesimpulan Penelitian |
| 1. | (Megawaty et al., 2017) | Aplikasi *Mobile* Darurat Bencana Pada Kabupaten Musi Banyuasin | Metode yang digunakan pada penelitian ini adalah metode Mobile-D dimana metode ini pengembangan aplikasi menggunakan pendekatan agile. | Penelitian ini menghasilkan sebuah aplikasi *mobile* darurat bencana pada kabupaten musi banyuasin ini, aplikasi ini dirancang menggunakan bahasa pemrograman PHP dan basis data MYSQL serta Android Studio untuk membuat *user* *interface*. Aplikasi rawan bencana ini bisa melaporkan beberapa bencana diantaranya banjir, longsor dan angin puting beliung. |
| 2. | (Saragih et al., 2020) | Aplikasi Pemantauan Banjir Berbasis Android Menggunakan Komunikasi Lora | Metode yang digunakan adalah metode *eksperimental* | Penelitian ini menghasilkan prototype banjir (kit banjir) sebagai alat monitoring dilengkapi dengan website sebagai *server* untuk mengirim data monitoring ke aplikasi Android untuk ditampilkan secara *real time* pada aplikasi *mobile* yang bisa diakses oleh pengguna. Aplikasi ini dirancang menggunakan *framework* angular JS dan *framework* Laravel untuk pembuatan website, dan bahasa pemrograman Java untuk pembuatan aplikasi Android. |
| 3. | (Mukti & Puspita, 2019) | Sistem Informasi Peringatan Dini Bencana Pada Kota Pagar Alam Berbasis Mobile | Metode *Rapid Application Development* (RAD) | Penelitian ini menghasilkan sebuah sistem aplikasi yang dirancang menggunakan bahasa pemrograman PHP dan *database* mariadb atau MYSQL yang dapat memberikan notifikasi kepada pengguna ketika hujan lebat terjadi di Kota Pagar Alam. Sistem aplikasi ini menggunakan teknologi *progressive web app* yang berjalan di perangkat *mobile*. |
| 4. | (Saputri & Halim, 2019) | Aplikasi Peta Titik Rawan Banjir Di Kota Palembang | Metode yang digunakan adalah metode *mobile-D* | Penelitian ini menghasilkan sistem aplikasi peta daerah rawan banjir di kota palembang berbasis Android. Aplikasi ini dapat digunakan sebagai sarana bagi masyarakat khususnya pengendara baik kendaraan roda dua maupun roda empat untuk dapat melihat lokasi yang memungkinkan terkena banjir, jadi pengendara bisa hindari area tersebut sehingga mengurangi dampak kerusakan dan kemacetan karena banjir. |
| 5. | (Ariyani & Putri, 2017) | Sistem Monitoring Banjir Pada Jalan Menggunakan Aplikasi Mobile Dan Modul Wi-Fi | Metode *eksperimental* | Penelitian ini menghasilkan alat monitoring banjir dengan mengimplementasikan sensor ultrasonik berbasis mikrokontroler yang diletakan di tiga titik ruas jalan dengan tampilan google maps untuk monitoring ketinggian permukaan air. Penelitian ini juga menghasilkan sebuah aplikasi *mobile* sebagai *interface* untuk menampilkan data ketinggian air menggunakan bahasa pemrograman HTML, PHP dan JavaScript dan App Inventor sebagai pembuat *user interface*. |
| 6. | (Lewi et al., 2017) | Sistem Monitoring Ketinggian Air Berbasis Internet Of Things Menggunakan Google *Firebase* | Metode eksperimental | Penelitian ini menghasilkan sebuah aplikasi Android yang dapat memonitoring ketinggian air tanpa keterbatasan jarak. Aplikasi mobile ini dapat membaca titik koordinat yang dikirimkan oleh alat sehingga aplikasi dapat membaca status ketinggian air. Aplikasi ini dirancang menggunakan Android studio menggunakan bahasa pemrograman Java dan *firebase* sebagai penyimpan data monitoring secara *real time*. |

# Metodologi Penelitian

Metodologi penelitian adalah serangkaian prosedur yang harus dilakukan dalam mengumpulkan data dan informasi untuk mendapatkan solusi dan menguji hipotesis penelitian. Dalam metode penelitian ini dilakukan dengan mengikuti metode ilmiah dengan tahapan perumusan masalah, studi literatur, pengembangan sistem, pengujian sistem, dan kesimpulan.

#### Lokasi Penelitian

Untuk proses pengembangan perangkat lunak dan uji coba dilakukan di Laboratorium Komputasi Universitas Mulawarman Fakultas Teknik, yang beralamat di JL.Sambaliung No.9 Sempaja Selatan, Samarinda Utara, Kecamatan Samarinda Ulu, Kota Samarinda, Kalimantan Timur.

#### Tahapan Penelitian

Tahapan penelitian adalah rangkaian secara sistematis yang direncanakan untuk dilalui terlebih dahulu sebelum mencapai hasil, dimana hasil tersebut berupa kesimpulan dari penelitian, dimulai dari studi pendahuluan, menentukan masalah, menentukan tujuan, mempelajari literatur, perancangan sistem, pengujian, analisis data, dan terakhir adalah memperoleh kesimpulan dari proses penelitian.

1. Studi Pendahuluan

Studi pendahuluan merupakan awal dari rangkaian kegiatan dalam penelitian, di tahap ini peneliti mempelajari materi yang berkaitan dengan pengembangan dan perancangan aplikasi berbasis Android, bahasa pemrograman aplikasi dan mencari lokasi dimana penelitian dapat dilakukan.

1. Menentukan Masalah

Pada tahap ini peneliti akan menemukan rumusan masalah dari sebuah penelitian yang akan dilakukan. Pada proses penentuan masalah biasanya didukung dengan pengumpulan data menggunakan jurnal, website, buku, skripsi ataupun dari hasil penelitian terdahulu.

1. Menentukan Tujuan

Mendefinisikan tujuan yaitu proses menemukan tujuan dari penelitian yang akan dilakukan, dimana tujuan tersebut meliputi cara-cara untuk memecahkan permasalahan yang ada di rumusan masalah.

1. Studi Literatur

Studi literatur adalah proses mencari dan mengumpulkan literatur (kajian pustaka) yang berkaitan dengan masalah yang terdapat dalam penelitian, yang bersumber dari buku, jurnal atau sumber lainnya yang bertujuan untuk menyusun dasar teori yang digunakan dalam melakukan penelitian.

1. Pengembangan Sistem

Metode pengembangan pada sistem aplikasi ini menggunakan metode *Rapid Application Development* (RAD) metode ini terdiri dari tahap perencanaan kebutuhan sistem, tahap desain sistem, tahap membangun sistem dan tahap implementasi atau penyelesaian produk.

1. Uji Coba Sistem Aplikasi

Dalam proses ini peneliti akan melakukan pengujian sistem menggunakan beberapa pengujian untuk memastikan tidak ada proses yang tidak sesuai dengan desain perancangan sistem.

1. Hasil dan Pembahasan

Pada tahap ini yaitu peneliti melakukan penyajian hasil dan pembahasan mengenai data yang diperoleh dari hasil uji aplikasi dengan beberapa metode pengujian pada sistem aplikasi.

1. Memperoleh Kesimpulan

Pada tahap akhir ini adalah penulis menarik kesimpulan atau rangkuman dari skripsi ini tentang program aplikasi yang sudah dikembangkan.

Adapun tahap penelitian digambarkan dalam bentuk diagram alir adalah sebagai berikut:



###### Gambar 8. Diagram Alur Penelitian

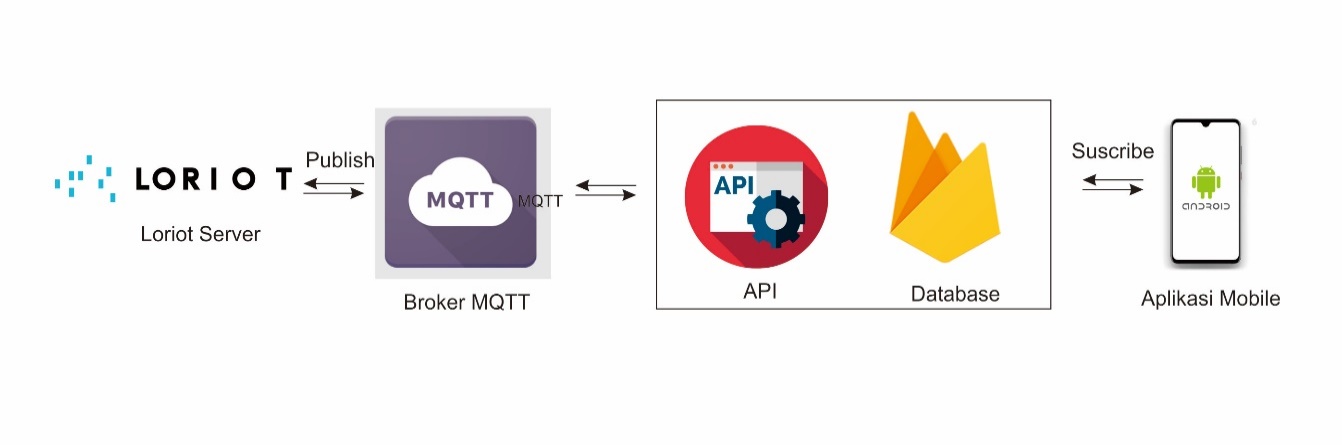
Penjelasan diagram alir:

1. Mulai.
2. Langkah selanjutnya adalah studi pendahuluan, hal ini dilakukan untuk memperoleh informasi awal terkait penelitian yang akan dilakukan serta dilakukannya pengembangan pada penelitian sebelumnya.
3. Rumusan masalah dibuat dengan tujuan agar ada kesesuaian mengenai masalah dengan konsep penelitian awal.
4. Langkah selanjutnya adalah menentukan tujuan penelitian, hal ini dilakukan guna didapatkan hasil yang diharapkan, pada penelitian ini terdapat 4 tujuan yang telah dituliskan diatas.
5. Langkah selanjutnya adalah studi literatur, dimana langkah ini berisi teori dasar mengenai aplikasi native, Android studio SDK dan JDK, bahasa pemrograman Java, *firebase*, protokol MQTT dan penjelasan terkait metode yang digunakan dalam pengembangan aplikasi.
6. Langkah selanjutnya adalah dilakukan pengembangan aplikasi menggunakan metode *Rapid Application Development* (RAD) dimana metode ini memiliki 4 tahapan yaitu tahap perencanaan kebutuhan sistem, tahap desain sistem, tahap membangun sistem dan tahap implementasi atau penyelesaian produk.
   1. Pada tahap perencanaan kebutuhan ini adalah tahap awal penulis untuk mengidentifikasi kebutuhan apa saja yang diperlukan untuk membangun sistem aplikasi baik kebutuhan fungsional maupun kebutuhan non fungsional seperti kebutuhan perangkat keras dan perangkat lunak.
   2. Tahap kedua yaitu penulis melakukan desain perancangan keseluruhan sistem aplikasi dan desain UML seperti desain *use case*, desain *activity*, *desain sequence*, dan desain *user interface.*
   3. Tahap ketiga adalah proses pengembangan sistem aplikasi setelah dua tahap diatas dilengkapi dan sesuai. Pada tahap ini peneliti melakukan coding aplikasi, *debugging* dan integrasi dengan bagian-bagian lainya. Jika proses berjalan lancar maka dapat dilanjutkan ke tahap berikutnya, Apabila terjadi kendala pada proses pengembangan aplikasi, peneliti akan kembali ke tahap sebelumnya.
   4. Tahap implementasi ini adalah tahap terakhir dimana semua perancangan aplikasi maupun *storyboard* diimplementasikan dalam bentuk kode program dengan menggunakan *platform* Android pada *software* Android Studio menggunakan bahasa pemrograman Java dan *extensible Markup Language* (XML) dan melakukan optimasi dan finalisasi aplikasi sebelum dilakukan uji coba.
7. Langkah selanjutnya adalah uji coba pada aplikasi dengan pengujian sebagai berikut:
   1. Pengujian aplikasi dengan metode *blackbox testing* pada semua fitur *fungsional* aplikasi.
   2. Uji performansi pada *device* aplikasi dengan menggunakan metode *Load Testing* menggunakan *tools* JMeter.
   3. Pengujian konektivitas antara MQTT Broker dan aplikasi *mobile*.
   4. Pengujian data *publish* dan *subscribe* dari LORIOT *server* ke aplikasi *mobile*.
8. Selanjutnya menyajikan data hasil dan pembahasan yang diperoleh dari uji coba keseluruhan sistem aplikasi sesuai dengan metode yang digunakan.
9. Pengambilan kesimpulan dan saran dari hasil penelitian yang telah dilakukan.
10. Selesai.

# Perancangan Sistem aplikasi

1. **Perancangan Arsitektur Sistem**

Dibawah ini merupakan perancangan sistem aplikasi secara keseluruhan, dimana sistem aplikasi ini melakukan pengambilan data ketinggian air dan GPS yang sudah ada di LORIOT dari *end node* sampai data monitoring tersebut ditampilkan pada aplikasi *mobile*.



**Gambar 9. Arsitektur Sistem Aplikasi**

Pada gambar 5. diatas merupakan alur komunikasi antar perangkat yang digunakan pada penelitian ini. Pertama LORIOT *server* melakukan *publish* data ketinggian banjir ke broker MQTT. API akan melakukan permintaan data monitoring ketinggian banjir pada broker MQTT dan data akan simpan pada *database*. *Database* memberikan respon dan data ketinggian banjir diberikan ke API dan diteruskan kembali ke aplikasi *mobile* MQTT. Data ketinggian banjir berhasil didapatkan oleh aplikasi *mobile* MQTT dan terhubung ke *server* LORIOT, selanjutnya aplikasi *mobile* MQTT melakukan *subscribe* data ketinggian banjir pada broker dan *topic* yang sama sehingga data akan ditampilkan pada pengguna.

##### Perancangan Diagram Alir Sistem

Diagram alir keseluruhan sistem monitoring ketinggian banjir dapat dilihat pada gambar dibawah ini :



**Gambar 10. Perancangan Diagram Alir Keseluruhan Sistem**

Penjelasan diagram alir :

* 1. Mulai
  2. LORIOT *server* menyimpan data ketinggian air dan GPS dari *node* sensor sesuai *node* diletakan. Data ketinggian air yang diterima adalah sebagai input yang akan ditampilkan pada pengguna*.*
  3. LORIOT *server* mempublish data ketinggian air dan GPS ke MQTT Broker. MQTT Broker berfungsi untuk mengatur data *publish* dan data *subscribe* dari berbagai perangkat untuk menyampaikan pesan dari pengirim ke penerima.
  4. Selanjutnya MQTT Broker akan memproses data ketinggian air dan GPS yang telah diterima untuk dikirim ke *database*.
  5. Data ketinggian air dan GPS tersimpan di *Firebase database*.
  6. API akan menghubungkan antara pengguna dan *database* untuk memproses *request* data pada aplikasi monitoring banjir.
  7. MQTT *client* mengsubscriber pada kategori topik data ketinggian banjir.
  8. Setalah itu data monitoring banjir akan ditampilkan pada aplikasi *mobile* pengguna.
  9. Selesai

##### Perancangan Kebutuhan Sistem

Dalam proses pengembangan sistem aplikasi *mobile* ini, diperlukan analisis kebutuhan sistem aplikasi agar pembuatan aplikasi bisa berjalan sesuai perancangan, Adapun kebutuhan pengembangan sistem aplikasi yaitu:

* 1. **Kebutuhan Fungsional**

Kebutuhan fungsional adalah gambaran tentang fitur apa saja yang dibutuhkan dalam pembuatan program aplikasi. Berikut ini adalah kebutuhan fungsional dari perancangan aplikasi yang akan dibuat adalah sebagai berikut:

1. Aplikasi ini menampilkan halaman *Onboarding* sebagai halaman awal bagi pengguna yang baru saja menginstal aplikasi dan memberikan beberapa informasi awal tentang aplikasi.
2. Pada tampilan *home page* terdapattampilan peta lokasi monitoring ketinggian banjir sesuai lokasi *node* diletakan.
3. Pada aplikasi ini menyediakan halaman Informasi banjir yang akan menampilkan informasi ketinggian banjir mulai dari status banjir, ketinggian air, tanggal, waktu dan lokasi.
4. Pada halaman menu terdapat dua halaman lainnya yaitu halaman daftar lokasi atau *history* lokasi perangkat dan halaman tentang aplikasi.
5. Ketika pengguna membuka aplikasi maka pengguna akan mendapatkan notifikasi.
   1. **Kebutuhan Non Fungsional**

Analisis kebutuhan non-fungsional adalah analisis yang diperlukan untuk menentukan spesifikasi kebutuhan untuk melakukan pengembangan sistem. Spesifikasi non-fungsional juga meliputi elemen atau komponen apa saja yang diperlukan mulai dari sistem dibangun sampai diimplementasikan. Dalam analisis kebutuhan non-fungsional terdapat dua kebutuhan yaitu:

1. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan aplikasi ini adalah sebagai berikut :

Table 2. Daftar Alat

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Alat | Jumlah | Spesifikasi |
| 1. | Laptop | 1 | Merk : HP  Processor : Intel®CoreTM i3 6th Gen  RAM : 8 GB DDR4 2133 MHz  Sistem Operasi : Windows 10  Storage : 1TB SATA HDD 5400 RPM  Display : 14” HD (1366x768) / FHD (1920x1080)  Graphics : Discrete graphics Nvidia GT 930mx 4GB / Nvidia 940MX 4GB |
| 2. | *Smartphone* | 1 | Processor : Snapdragon 680 2.4 GHz Octa-core  Memory : 128 Gb  Ram : 8 Gb  Sistem Operasi : Android 12 (Snow Cone) |
| 3 | Kabel Data | 1 | Kabel USB Type C |

1. Kebutuhan Perangkat Lunak

Adapun perangkat lunak yang dibutuhkan dalam pengembangan aplikasi ini adalah sebagai berikut:

1. Android Studio *Chipmunk version* 2021.2.1 *version plugin* 3.2-7.2 dan *Android Gradle version* 7.2.2
2. Android SDK *version* 33 dan Android JDK *version* 18.0.2.0
3. LORIOT
4. Firebase *Raltime database*
5. Broker MQTT (Eclipse Mosquitto *version* 5.0, 3.1.1)
6. Aplikasi Apache JMeter *version* 5.5

##### Perancangan Desain Aplikasi

Perancangan sistem aplikasi ini dibuat menggunakan pemodelan *Unified Modeling Language* (UML) yang merupakan suatu pemodelan untuk menggambarkan visualisasi perancangan aplikasi dengan sistem berorientasi objek. Pemodelan diagram yang digunakan pada penelitian ini antara lain *use case diagram*, *sequence diagram*, *activity diagram* dan perancangan *storyboard*.

* 1. ***Use Case Diagram***

*Use case diagram* merupakan diagram yang menggambarkan hubungan antara aktor dan aplikasi. *Use case diagram* dari aplikasi ini dapat dilihat pada gambar 11. di bawah ini:



**Gambar 11. Use Case Diagram**

* 1. ***Sequence Diagram***

*Sequence* *Diagram* adalah jenis interaksi diagram yang menggambarkan visualisasi dan melakukan validasi berbagai skenario teknis secara *runtime* secara berurutan yang menunjukkan interaksi yang diperlukan antara aktor dan sistem untuk melihat data monitoring pada aplikasi. Dibawah ini merupakan gambar *sequence diagram* dari aplikasi ini :



###### Gambar 12. *Sequence diagram*

Pada gambar 12. diatas merupakan gambaran interaksi antara pengguna dalam mengakses aplikasi *mobile* dengan menampilkan pesan atau perintah yang dikirim, seperti pada gambar diatas, mulai pengguna masuk ke aplikasi ditunjukan dengan garis putus-putus (*lifeline*) yang melambangkan aktifitas dari objek, dan terhubung dengan persegi panjang (*activation box)* yang mempresentasikan waktu yang dibutuhkan suatu sistem untuk menyelesaikan perintah atau pesan, dimana perintah atau pesan digambarkan sebagai anak panah dari satu lifeline ke lifeline yang lain sampai pada akhirnya pengguna dapat melihat data monitoring.

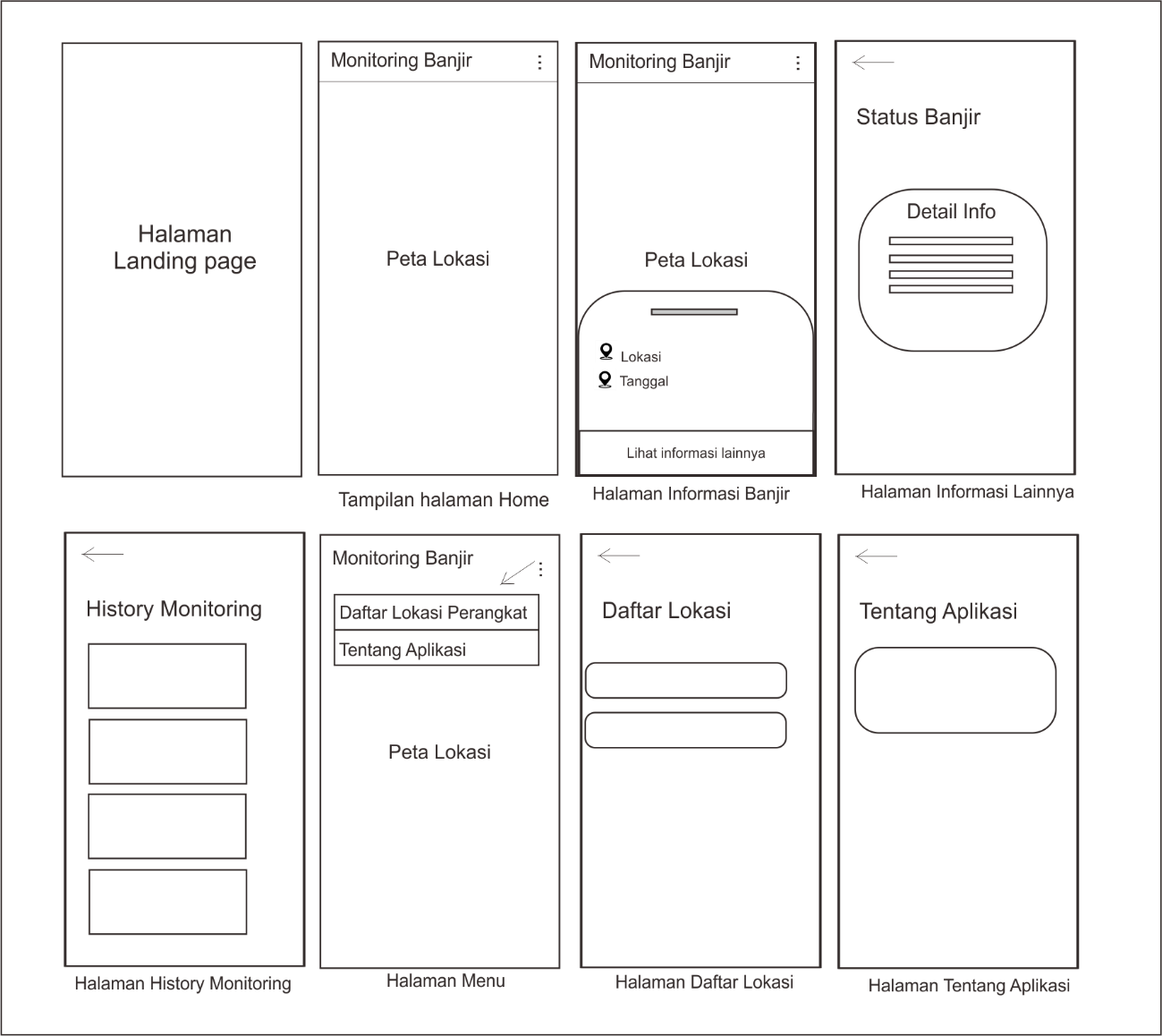
* 1. ***Activity Diagram***

Activity diagram yaitu diagram yang menunjukan data, aksi secara terstruktur yang dirancang dalam sebuah sistem. Activity diagram pada aplikasi monitoring ini dapat dilihat pada gambar dibawah ini:

***Gambar 13. Activity Diagram Pengguna***

* 1. **Perancangan Desain *Storyboard***

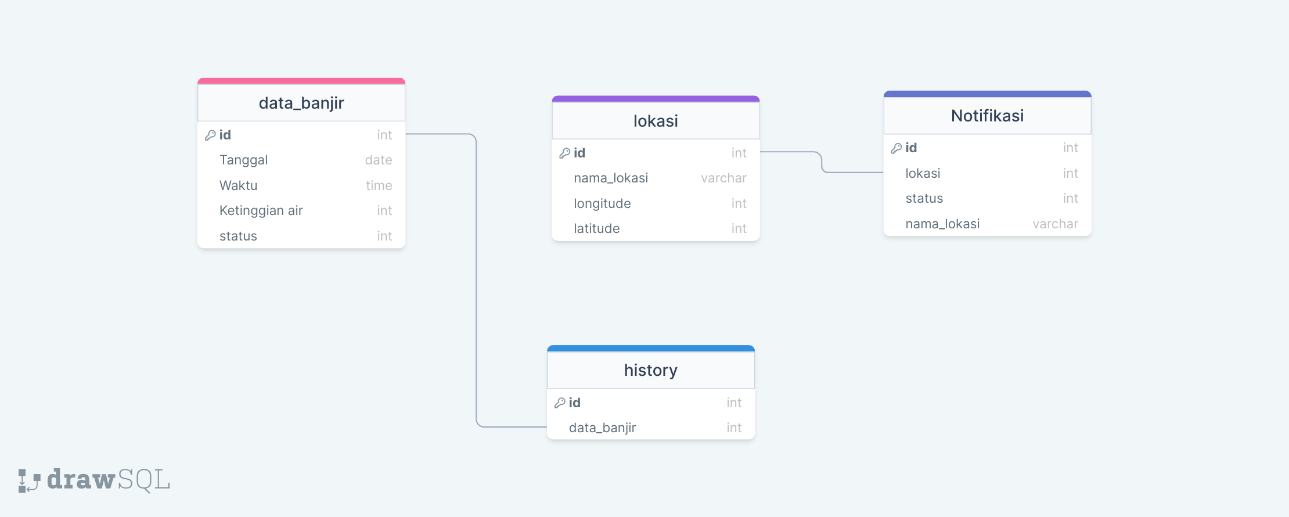
Perancangan *storyboard* aplikasi ini dibuat untuk memberikan gambaran fitur aplikasi yang akan dikembangkan. Adapun gambaran desain interface dalam pengembangan aplikasi adalah sebagai berikut:



**Gambar 14. Perancangan Storyboard Aplikasi**

* 1. **Perancangan *Database***

Desain *database* diimplementasikan dengan menggunakan skema *Javascript Object Notation* (JSON) dengan menuliskan format data JSON dengan *title*, *type*, *properties* sebagai atribut yang ada pada objek. Firebase digunakan untuk menyimpan data monitoring banjir untuk menampilkan data keseluruhan monitoring yang sudah ditampilkan pada aplikasi Android dari LORIOT *cloud* *server*. *Database* aplikasi ini terdiri dari 1 jenis *database* yaitu *database* ketinggian air dimana *database* ini berisi tanggal, status, ketinggian air dan lokasi. Gambar dibawah ini menunjukan perancangan *database* pada penelitian ini.



**Gambar 4. Perancangan Database**

# Uji Coba Sistem

1. **Pengujian *Black Box* *Testing***

Pengujian fungsional aplikasi merupakan proses pengujian fungsi sistem untuk memastikan bahwa sistem bekerja sesuai dengan perancangan. Pengujian ini menggunakan metode *balck box testing*. Jika pengujian fitur aplikasi berhasil maka akan diberikan tanda centang [✓] dan jika gagal maka akan diberi tanda silang [✕]. Hasil yang diharapkan dari setiap fitur aplikasi ditunjukkan pada tabel 3. dibawah ini :

Table 3. Pengujian *Black box Testing*

|  |  |  |
| --- | --- | --- |
| **No** | **Pengujian** | **Hasil** |
| 1 | Menekan tombol *next* pada halaman *onboarding* untuk menuju halaman *Home* | [] Berhasil  [] Tidak Berhasil |
| 2 | Menampilkan peta lokasi banjir | [] Berhasil  [] Tidak Berhasil |
| 3 | Menekan tombol informasi banjir | [] Berhasil  [] Tidak Berhasil |
| 4 | Melihat data tanggal, waktu , status, dan ketinggian banjir | [] Berhasil  [] Tidak Berhasil |
| 5 | Melihat *history* monitoring banjir | [] Berhasil  [] Tidak Berhasil |
| 6 | Menekan tombol halaman menu | [] Berhasil  [] Tidak Berhasil |
| 7 | Menekan tombol Melihat Lokasi Perangkat | [] Berhasil  [] Tidak Berhasil |
| 8 | Menekan tombol tentang aplikasi | [] Berhasil  [] Tidak Berhasil |
| 9 | Mendapatkan notifikasi banjir | [] Berhasil  [] Tidak Berhasil |

1. **Pengujian Kompatibilitas Aplikasi**

Pengujian kompatibilitas aplikasi merupakan pengujian pada berbagai macam *device* yang berbeda, dimana pengujian ini menggunakan delapan *virtual device* yang sudah mewakili beberapa versi Android dengan spesifikasi *device* yang berbeda, dibawah ini merupakan penjelasan spesifikasi *device* yang digunakan :

**Tabel 4*.* Tabel Spesifikasi *device***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Nama *Device*** | **Versi Android/API** | **Memory (RAM)** | **Resolusi Layar (Pixels)** | **Hasil** |
| 1. | Android *Virtual Device* | Android 12.0/31 | 2 GB | 1600 x 720 |  |
| 2 | Android *Virtual Device* | Android 11.0/30 | 2 GB | 1920 x 1080 |  |
| 3 | Android *Virtual Device* | Android 10.0/29 | 2 GB | 2960 x 1440 |  |
| 4 | Android *Virtual Device* | Android Pie 9.0/28 | 2 GB | 2880 x 1440 |  |
| 5 | Android *Virtual Device* | Android Oreo 8.1/27 | 2 GB | 1920 x 1080 |  |
| 6 | Android *Virtual Device* | Android Nougat 7.0/25 | 2 GB | 1280 x 720 |  |
| 7 | Android *Virtual Device* | Android Marshmallow 6.0 /23 | 2 GB | 1280 x 768 |  |
| 8 | Android *Virtual Device* | Android Lollipop 5.1/22 | 2 GB | 640 x 360 |  |

1. **Pengujian performansi**

Pengujian performansi merupakan pengujian yang dilakukan untuk mengetahui kinerja aplikasi dalam berbagai kondisi (*load condition*). Dalam pengujian ini menggunakan metode *Load testing* menggunakan *tools* JMeter.

Table 5. Pengujian performa pada *device*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Variabel** | **Hasil** | **Satuan** | **Deskripsi** |
| 1. | Pengujian CPU pada *device* |  | Persentase (%) | Penggunaan CPU pada *device* |
| 2. | Pengujian Memory *Usage* |  | Byte (B) | Penggunaan memory pada *device* |
| 3. | Pengujian Execution *Time* |  | Millisecond (ms) | Waktu eksekusi program |
| 4. | Pengujian jaringan |  | Byte/second (B/s) | Penggunaan Jaringan |

1. **Pengujian Data *Publish* dan *Subscribe***

Pengujian data *publish* dan *subscribe* dilakukan untuk mengetahui jumlah data, kesesuaian data serta kemungkinan kesalahan antara yang di *publish* dan di *subscribe* sesuai durasi waktu pengamatan.

Table 4. Pengujian Data *Publish* dan *Subscribe*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Objek Pengukuran** | **Topic** | **Jumlah data terkirim (Publisher)** | **Jumlah data diterima (Subscriber)** | **Selisih Data** | **Error** |
| 1. | Ketinggian Banjir |  |  |  |  |  |
| 2. | Data GPS |  |  |  |  |  |

# Jadwal Kegiatan

Adapun jadwal penelitian dapat ditunjukkan seperti pada tabel berikut:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Tahapan Penelitian** | **Tahun 2022** | | | | | | | | | | | | | | | | | | | | | | | |
| **Juni** | | | | **Agustus** | | | | **September** | | | | **oktober** | | | | **November** | | | | **Desember** | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1. | Pengajuan Judul proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2. | Verifikasi Judul skripsi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3. | Penyusunan Draft Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4. | Bimbingan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5. | Seminar Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Revisi Proposal Skripsi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6. | Penelitian |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7. | Penyusunan Laporan Hasil |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8. | Seminar Hasil |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9. | Proses Perbaikan Skripsi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10. | Sidang Skripsi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**DAFTAR PUSTAKA**

Ariyani, D. R., & Putri, R. E. (2017). Sistem Monitoring Banjir Pada Jalan Menggunakan Aplikasi Mobile Dan Modul Wi-Fi. *Seminar Nasional Sains Dan Teknologi 2017*, *November*, 1–8.

Barik, J., Rachmawan, A., Surabaya, U. N., Surabaya, U. N., & Banjir, B. (2022). *Perancangan Desain User Interface*. *3*(3), 252–266.

Burd, B. (2017). *Java Programming for Android Developers For Dummies* (2nd editio). John Wiley & Sons, Inc.

DiMarzio, J. F. (2016). *Android Programming With Android Studio* (C. (Shod) Darby (ed.)). John Wiley & Sons, Inc.

Dincer, A., & Uraz, B. (2013). *Google Maps JavaScript API Cookbook*.

Faqih, A. H., Laksana, T. G., & Febriati, A. (2018). Sistem informasi reporting curiculum vitae karyawan menggunakan metode rapid application development berbasis website di PT. PINS Indonesia. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *3*(1), 69–75.

JMeter, A. (2022). *Apache JMeter*. https://jmeter.apache.org/

Lewi, E. B., Sunarya, U., & Ramadan, D. N. (2017). Sistem Monitoring Ketinggian Air Berbasis Internet of Things Menggunakan Google Firebase. *Universitas Telkom, D3 Teknik Telekomunikasi*, *1*(1), 1–8.

Liu, F. (2013). *Android Native Development Kit Cookbook* (L. Purkait (ed.)). Packt.

Loriot.io. (2022). *LORIOT NETWORK SERVER*. https://www.loriot.io/index.html

Megawaty, M., Bakti, A., & Permana, H. (2017). Aplikasi Mobile Darurat Bencana Pada Kabupaten Musi Banyuasin. *Jurnal Ilmiah Matrik*, *3*, 257–266.

Moroney, L. (2017). The Definitive Guide to Firebase. In Welmoed Spahr (Ed.), *The Definitive Guide to Firebase*. Apress.

Mukti, Y. I., & Puspita, D. (2019). Sistem Informasi Peringatan Dini Bencana Pada Kota Pagar Alam Berbasis Mobile. *Jusikom : Jurnal Sistem Komputer Musirawas*, *4*(02), 65–74. https://doi.org/10.32767/jusikom.v4i2.608

Rumpe, B. (2016). Modeling with UML Language, Concepts, Methods. In *Modeling with UML*. RWTH Aachen University.

Saputri, N. A. O., & Halim, R. M. N. (2019). Aplikasi Peta Titik Rawan Banjir Di Kota Palembang. *Angewandte Chemie International Edition, 6(11), 951–952.*, *6*, 451–456.

Saragih, A. W., Farhanah, A., & Cahyana. (2020). Aplikasi Pemantauan Banjir Berbasis Android Menggunakan Komunikasi Lora. *E-Proceeding of Applied Science*, *6*(2), 4004–4010.

Statcounter. (2022). *Mobile & Tablet Android Version Market Share Indonesia*. https://gs.statcounter.com/os-version-market-share/android/mobile-tablet/indonesia#monthly-202201-202209

Tigli, J.-Y. (2016). *Middleware for Internet of Things - MQTT (Message Queuing Telemetry Transport)*. *33*(0), 1–12.