

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Terdapat dua musim di Indonesia, yaitu musim kemarau dan musim penghujan. Menurut Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) musim kemarau berlangsung antara bulan April sampai bulan September, sedangkan musim penghujan berlangsung antara bulan Oktober sampai bulan Maret atau dengan jangka waktu enam bulan.

Tingginya curah hujan menyebabkan terjadinya beberapa bencana alam seperti banjir dan tanah longsor yang mengakibatkan kerugian bagi masyarakat. Pencegahan bencana banjir dapat dilakukan dengan berbagai cara dengan lebih mencintai lingkungan seperti membuang sampah pada tempatnya, rajin untuk membersihkan lingkungan sekitar, terutama pada saluran air atau selokan yang terkadang tersumbat oleh sampah sehingga air tidak dapat mengalir dengan lancar dan akhirnya meluap dan menyebabkan terjadinya banjir.

Begitu juga pada sungai maupun bendungan harus dijaga kelestariannya dengan tidak membuang sampah ataupun limbah ke sungai atau bendungan. Akan tetapi, ada beberapa faktor yang tidak dapat dikendalikan oleh manusia seperti perubahan cuaca ekstrem yang menyebabkan tingginya curah hujan yang cenderung lama dan terus – menerus. Hal tersebut menjadi salah satu faktor terjadinya bencana banjir terutama di negara Indonesia.

Kota Parepare merupakan sebuah kota yang terletak pada provinsi Sulawesi Selatan. Selama tahun 2020 hingga 2022, terjadi kenaikan banjir yang signifikan pada wilayah tertentu di Kota Parepare. Wilayah seperti Perumahan Palem, Perumahan Tegal, Kelurahan Lemoe dan Perumnas Wekke'e mengalami banjir dengan ketinggian 1 sampai 2 meter bahkan lebih. Menurut (Detik 2023) pada awal tahun 2023 sebanyak 5,294 jiwa terdampak dan 2 meninggal akibat dampak banjir yang terjadi di Parepare. Pihak pemerintahan Kota Parepare masih memikirkan cara agar informasi banjir dan kenaikan air dapat segera disampaikan oleh masyarakat yang tinggal pada wilayah rawan banjir tersebut. Dengan begitu, solusi tepat yang dibutuhkan masyarakat yaitu peringatan dini dan juga monitoring ketinggian air sungai secara *real time*. Diharapkan masyarakat dapat tetap waspada ketika bencana banjir yang akan datang kapan saja, dan juga diharapkan dapat mengurangi kerugian masyarakat dan dampak yang diakibatkan oleh bencana banjir.

Puncak terjadinya banjir pada perumahan di Parepare adalah letaknya lokasi kawasan perumahan tersebut. Geografi pada Parepare yang mana tidak rata dan berbukitan menyebabkan beberapa perumahan berada pada kawasan bawah bukit. Ini menyebabkan air hujan yang mengalir turun ke kawasan perumahan tersebut menyebabkan banjir area tersebut. Selain itu, puncak kedua adalah pesatnya pembangunan yang mana menghilangkan tanaman pada area berbukitan sehingga serapan air hujan kurang dan menyebabkan banjir. Ini dapat dilihat dimana banjir mulai terjadi pada 2 tahun terakhir pada area seperti perumahan palem, perumahan tegal, kelurahan lemoe dan perumnas wekke'e.

Melalui latar belakang singkat diatas, penulis tertarik untuk mengangkat sebuah judul yaitu "Prototype Sistem Notifikasi Banjir Berbasis IOT" yang mana nantinya memberikan notifikasi banjir kepada masyarakat serta notifikasi pada *smartphone*.

### **B. Rumusan Masalah**

Berdasarkan uraian latar belakang yang telah diuraikan, maka akan dibuat suatu penelitian pada masalah yang sedang dihadapi. Adapun latar belakang masalah yang dapat diidentifikasi yaitu :

1. Bagaimana alat dapat mendeteksi ketinggian air pada sungai atau saluran air jika pada batas berbahaya?
2. Bagaimana alat dapat mendeteksi ketinggian air jika melewati batas sungai atau saluran air?
3. Bagaimana nantinya mengirim notifikasi kepada aplikasi mobile yang dibuat agar dapat menerima notifikasi ketinggian air secara *real-time*?

### C. Tujuan Penelitian

Tujuan yang hendak dicapai dalam penelitian ini adalah :

1. Alat dapat mendeteksi ketinggian air pada sungai atau saluran air jika pada batas berbahaya
2. Alat dapat mendeteksi ketinggian air jika melewati batas sungai atau saluran air
3. Mengirim notifikasi kepada aplikasi mobile yang dibuat agar dapat menerima notifikasi ketinggian air secara *real-time*

### D. Batasan Masalah

Agar pembahasan menjadi lebih terarah pada pokok permasalahan maka penulis membatasi masalah sebagai berikut :

1. Dibuat prototipe sungai dan saluran air pada daerah rawan banjir di Kota Parepare, Sulawesi Selatan.
2. *Controller* yang digunakan adalah *Esp 8266* yang mengambil nilai ketinggian air dan mengirimkan ke *Server*.
3. Sensor yang digunakan adalah sensor ultrasonik untuk mengukur ketinggian air pada sungai atau saluran air dan *water level sensor* untuk mengukur ketinggian air setelah melewati batas sungai dan saluran, dan juga penambahan *buzzer* sebagai tanda/sirine adanya kenaikan air.
4. *Server* nantinya dibuat menggunakan *Node JS* yang mana mengintegrasikan *web socket* untuk mengirimkan data ketinggian air secara *real time*.
5. Aplikasi *mobile* dibuat menggunakan *Framework Flutter* .

## E. Manfaat Penelitian

Adapun manfaat penelitian sebagai berikut :

### 1. Aspek Akademis

Sebagai langkah awal bagi peneliti untuk dapat mengembangkan teori-teori yang didapat selama ini di bangku perkuliahan khususnya mengenai *Internet of Things (IoT)*.

### 2. Aspek Praktis

- a. Memberikan informasi *real-time* tentang potensi banjir, sehingga masyarakat dapat bersiap lebih awal dan mengurangi risiko kerugian.
- b. Mempercepat tindakan darurat, seperti evakuasi atau pengamanan aset, dengan adanya data akurat yang langsung diterima oleh pihak berwenang.
- c. Dengan notifikasi yang tepat waktu, kerugian material dan korban jiwa dapat diminimalkan karena masyarakat memiliki waktu lebih banyak untuk menyelamatkan diri dan barang berharga.
- d. Masyarakat menjadi lebih sadar akan kondisi lingkungan sekitar dan lebih siap menghadapi kemungkinan bencana.
- e. Sistem ini memungkinkan pemantauan terus menerus tanpa perlu intervensi manusia, sehingga lebih andal dan konsisten.
- f. Sebagai tugas akhir untuk mendapatkan gelar sarjana strata 1 (S1) pada Fakultas Teknik Informatika

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. Kajian Teori**

##### **1. Banjir**

Banjir merupakan bencana alam yang paling sering terjadi di Indonesia. Definisi banjir adalah keadaan dimana suatu daerah tergenang oleh air dalam jumlah yang besar. Kedatangan banjir dapat diprediksi dengan memperhatikan curah hujan dan aliran air. Namun kadangkala banjir dapat datang tiba-tiba akibat dari angin badai atau kebocoran tanggul yang biasa disebut banjir bandang.

Penyebab banjir mencakup curah hujan yang tinggi; permukaan tanah lebih rendah dibandingkan muka air laut; wilayah terletak pada suatu cekungan yang dikelilingi perbukitan dengan sedikit resapan air; pendirian bangunan disepanjang bantaran sungai; aliran sungai tidak lancar akibat terhambat oleh sampah; serta kurangnya tutupan lahan di daerah hulu sungai. Meskipun berada di wilayah "bukan langganan banjir". Setiap orang harus tetap waspada dengan kemungkinan bencana alam ini.

##### **2. Evakuasi**

Makna kata dievakuasi adalah dipindahkan ke tempat yang aman. Evakuasi sangat penting untuk dipahami agar masyarakat bisa mengetahui apa yang dilakukan saat terjadi bencana atau peristiwa lain yang berbahaya.

Pengetahuan tentang evakuasi sudah seharusnya diajarkan ke masyarakat sedini mungkin.

Kata dasar dari dievakuasi adalah evakuasi. Dikutip dari [kbbi.kemdikbud.go.id](http://kbbi.kemdikbud.go.id), kata artinya adalah:

- a) pengungsian atau pemindahan penduduk dari daerah-daerah yang berbahaya, misalnya bahaya perang, bahaya banjir, meletusnya gunung api, ke daerah yang aman.
- b) pemindahan sesuatu (kendaraan, barang, dan sebagainya) ke tempat aman

Berdasarkan pengertian tersebut, bisa disimpulkan bahwa makna kata dievakuasi adalah dipindahkan ke tempat yang aman. Abdurrahman, dkk dalam *Pertolongan Pertama pada Kegawatdaruratan di Sekolah* (2022:11) menjelaskan bahwa evakuasi dibagi menjadi dua jenis, yaitu:

- a) Evakuasi berskala kecil adalah jenis evakuasi yang dilakukan secara darurat dan menunggu pertolongan atau penyelamatan lebih lanjut. Evakuasi jenis ini umumnya memanfaatkan kemampuan pergerakan individu untuk berpindah ke tempat yang aman dalam gedung dan tetap di tempat tersebut sambil menunggu bantuan datang.
- b) Evakuasi berskala besar bisa juga disebut sebagai evakuasi distrik. Evakuasi jenis ini adalah bagian dari manajemen sebuah bencana.

Berikut ini adalah beberapa contoh evakuasi yang bisa dipahami:

- a) Contohnya adalah banjir besar yang terjadi di Jakarta yang berdampak beberapa daerah menjadi terendam banjir. Warga Jakarta yang menjadi korban banjir dievakuasi ke lokasi yang lebih aman.
- b) Tsunami adalah yang menelan banyak korban. Beberapa negara yang pernah terdampak tsunami adalah Indonesia, Thailand, dan Jepang. Proses evakuasi korban tsunami dilakukan oleh relawan yang berasal dari berbagai negara.
- c) Selain bencana alam, perang juga bisa menimbulkan banyak korban. Contohnya adalah perang yang terjadi di Palestina. Seluruh WNI yang ada di Palestina dievakuasi ke lokasi yang lebih aman. Evakuasi dilakukan sebelum dan selama perang terjadi.

### 3. *Android*

*Android* adalah sistem operasi yang dikeluarkan oleh *Google*. *Android* dibuat khusus untuk smartphone dan tablet. Berbagai macam produsen telah menggunakan *Android* sebagai sistem operasi untuk peranti (*device*) yang mereka produksi. *Android* juga mempunyai *store* dengan lebih dari 2 miliar pengguna aktif per bulannya, per Januari 2018. Sejak dirilis tahun 2008, *Google* telah mengeluarkan beberapa versi, dengan "*Pie*" sebagai versi yang terbaru (Imaduddin dan Permana 2018).

Pada tahun 2013, *Android* menjadi *operation system (OS)* terlaris pada *tablet* dan *smartphone*. Kini market share *Android* sedikitnya 80 % dari total

penjualan smarphone di tingkat global (statista.com). Tercatat pada tahun 2016 *Android store* memiliki lebih dari 2.8 juta aplikasi.

*Android* menarik bagi perusahaan teknologi yang membutuhkan barang siap jadi, biaya rendah dan kustomisasi *OS* untuk perangkat teknologi tinggi mereka. Hal ini menjadi daya tarik bagi banyak perusahaan, sehingga mereka memilih *Android.Source code* dari *Android* bersifat *open source*. Ini adalah hal menarik bagi komunitas *developer*, karena lisensi *open source* sangat mendukung untuk mengembangkan produknya dengan aman (Imaduddin dan Permana 2018).

Berikut adalah rangkaian sejarah perkembangan *Android* yang resmi diluncurkan oleh *Google* dari waktu ke waktu.

### 1. *Android 1.0 Apple Pie*

*Apple Pie* pertama kali diperkenalkan pada 23 September 2008. Fitur yang dimiliki pada *Android* ini ialah *play store*, *web browser*, kamera, sinkronisasi antara *Gmail*, kontak dan juga *Google Agenda*. *Android Apple Pie* juga sudah dilengkapi fitur *Maps* dan *YouTube*. Terdapat dua versi *Apple Pie* yaitu yang dirilis oleh *Google* dan juga *OHA*.

### 2. *Android 1.1 Banana Bread*

Setelah *Apple Pie*, munculah *Android Banana Bread* pada Februari 2009. *Android Banana Bread* atau yang juga disebut *Android Beta* memiliki fitur yang tak jauh beda dari *Apple Pie*. *Android Banana Bread* mulai bisa digunakan pada *smartphone HTC*.

### 3. *Android 1.5 Cup Cake*

*Android Cup Cake* merupakan versi *Android* ketiga yang dikembangkan oleh *Google*. *Android* ini dirilis mulai April 2009 dan menjadi sistem operasi yang mulai dilirik oleh pengguna ponsel. *Android* 1.5 menyertakan fitur-fitur baru seperti *keyboard* di layar dan dukungan *Bluetooth* serta peningkatan fitur yang ada seperti perubahan *UI* untuk manajemen aplikasi dan beberapa aplikasi *Google*.

#### 4. *Android 1.6 Donut*

*Android Donut* dirilis pada September 2009. Jenis *Android* ini pun mulai bisa memperbaiki masalah atau *bug*. Selain itu, pada jenis *Android Donut* ini memiliki fitur tambahan berupa fitur navigasi *Turn by Turn*.

#### 5. *Android 2.0 - 2.1 Éclair*

Pada Oktober 2009, *Android* kembali merilis *OS* terbarunya yang diberi nama *Eclair*. Versi *Android Eclair* merupakan awal dari diterapkannya aplikasi *Bluetooth 2.1* pada *Android*. Tak berselang lama, versi 2.0 pun di update ke versi 2.1 yang tentu saja memiliki fitur tambahan lain. Fitur tambahan pada versi ini ialah *Multi Touch*, *Live Wallpaper* serta *flash* kamera.

#### 6. *Android 2.2 Froyo*

Setelah *Eclair*, *Android* kemudian merilis *Android Frozen Yogurt* atau *Froyo*. *Android Froyo* versi 2.2 ini mulai dirilis pada bulan Mei 2010. *OS Android* ini merupakan pembaruan fitur dari versi sebelumnya. Di *Android Froyo*, pada *Google Chrome* di *Android* terdapat tambahan *Script Chrome*. Fitur ini berguna untuk menambah kecepatan kinerja pada *Google Chrome*.

### 7. *Android 2.3 Gingerbread*

Masih ditahun 2010, pada bulan Desember, *Google* kembali merilis *Android* dengan nama *Gingerbread*. Pada *Android* versi 2.3 ini mulai disematkan fitur *NFC*, *internet calling* dan juga *download manager*.

### 8. *Android 3.0 - 3.2 Honeycomb*

*Android Honeycomb* ini resmi dirilis pada Februari 2011. Selain itu, jenis *Android* ini mulai banyak digunakan pada perangkat tablet. *Android Honeycomb* memang dirancang untuk perangkat dengan ukuran layar yang lebih besar, terutama tablet.

### 9. *Android 4.0 Ice Cream Sandwich*

*Android Ice Cream Sandwich* merupakan jenis *Android* yang mulai populer di kalangan pengguna *smartphone*. Pasalnya, *Android Ice Cream Sandwich* ini bisa digunakan pada semua jenis platform dan juga *smartphone*. Selain itu, fitur dari sistem operasi ini pun semakin bertambah dan juga menjadi lebih *multitasking*.

### 10. *Android 4.1.2 Jelly Bean*

*Jelly Bean* merupakan satu dari macam-macam *OS Android* yang mulai populer digunakan. *Android Jell Bean* dirilis pada 27 Juni 2012. *Android* ini merupakan versi dari *Android* yang mendapatkan *update* hingga 2 kali. Untuk versi terbaru dari *Android* ini pun menggunakan *User Interface* dan juga menambahkan *Google Search* sebagai fitur baru mereka.

### 11. *Android 4.4.2 KitKat*

Setelah kehadiran *Jelly Bean*, penggunaan smartphone ber-OS *Android* makin menjamur. *Google* kemudian kembali merilis versi *Android* terbarunya yang dinamakan *KitKat*. *Android KitKat* resmi dirilis pada 2013 lalu dan memiliki banyak pembaharuan pada fitur-fitur yang dimiliki.

#### 12. *Android 5.0 Lollipop*

*Android Lollipop* dirilis pada tahun 2014 dan pertama kali digunakan pada smartphone *Google Nexus 6*. OS ini memiliki fitur tambahan yang menyempurnakan fitur-fitur terdahulunya. Selain itu, sistem kerja pada *Android Lollipop* lebih sempurna dari versi *Android* sebelumnya.

#### 13. *Android 6.0 Marshmallow*

*Marshmallow* mulai digunakan pada tahun 2015. *Android* versi 6.0 ini memiliki fitur tambahan support *USB Type C*. *Android Marshmallow* juga memulai adanya fitur sensor sidik jari dan juga daya baterai yang meningkat.

#### 14. *Android 7.0 Nougat*

Setelah *Marshmallow*, pada 2016, *Google* merilis *Android Nougat*. Beberapa fitur pada *Android Nougat* ini pun cukup memuaskan, seperti support pada *Multi Window*.

#### 15. *Android 8.0 Oreo*

*Android Oreo* dirilis pada Agustus 2017 dengan fitur-fitur yang makin canggih. *Android Oreo* lebih mengutamakan pada kecepatan dan juga efisiensi. Bahkan kecepatan *Boot* pun mencapai 2 kali lipat. Selain itu baterai pada sistem *Android* ini lebih tahan lama.

#### 16. *Android 9.0 Pie*

Pada Oktober 2018, *Google* merilis *Android Pie* yang memiliki desain dan fitur yang berbeda dari versi sebelumnya. Salah satu fitur menarik dari *Android Pie* ini ialah *App Actions*. Fitur ini membuat *Android* bisa memprediksi tindakan yang akan dilakukan oleh penggunanya

#### 17. *Android 10*

Mulai pada *Android 10*, *Google* tidak lagi menyematkan nama makanan pada julukan *Androidnya*. Menurut pihak *Android*, pemakaian nama *dessert* untuk *Android* ternyata tidak selalu dimengerti oleh masyarakat global. *Android 10* dirilis pada September 2019. *Android 10* inilah yang kini banyak digunakan pengguna *smartphone Android* di seluruh dunia.

#### 18. *Android 11*

*Android 11* mulai dikembangkan pada 2020. Versi *previewnya* mulai dirilis pada awal Juni 2020. Sebelumnya *Google* juga berencana untuk merilis *Android 11* versi beta untuk publik pada awal minggu Juni 2020. Namun gara-gara aksi protes yang meluas di Amerika Serikat, *Google* menundanya.

#### 4. **ESP 8266**

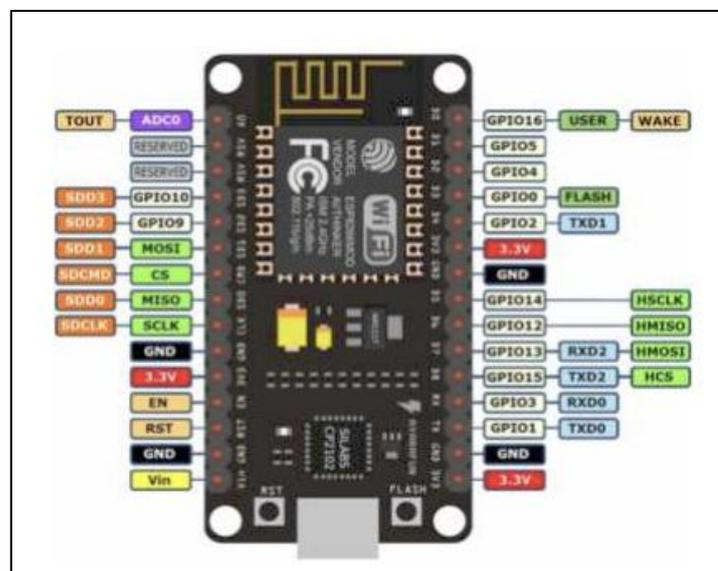
ESP8266 merupakan papan pengembangan produk *Internet of Things (IoT)* yang berbasis *Firmware eLua* dan *y (SoC) ESP8266-12E*. *ESP8266* sendiri merupakan chip *WiFi* dengan *protocol stack TCP/IP* yang lengkap.

*NodeMCU* dapat dianalogikan sebagai board *arduino-nya ESP8266*. Program *ESP8266* sedikit susah karena diperlukan beberapa teknik wiring

serta tambahan modul *USB* to serial untuk mengunduh program. Namun *NodeMCU* telah me-package *ESP8266* ke dalam sebuah board yang kompak dengan berbagai fitur layaknya mikrokontroler kapabilitas akses terhadap *Wifi* juga chip komunikasi *USB* to serial. Sehingga untuk memprogramnya hanya diperlukan ekstensi kabel data *USB* persis yang digunakan charging *smartphone*.

Spesifikasi dari *NodeMCU* sebagai berikut :

- a. 10 port pin *GPIO*
- b. Fungsionalitas *PWM*
- c. Antarmuka *I2C* dan *SPI*
- d. Antarmuka 1 *Wire*
- e. *ADC*



Gambar 2.1 *ESP 8266*

Berikut penjelasan dari pin – pin Gambar 2.1 NodeMCU tersebut.

- a. *ADC: Analog Digital Converter*. Rentang tegangan masukan 0- 1v,dengan skup nilai digital 0-1024.
- b. *RST* : berfungsi mereset modul
- c. *EN: Chip Enable, Active High*
- d. *IO16 :GPIO16*, dapat digunakan untuk membangunkan chipset dari mode *deep sleep*
- e. *IO14 : GPIO14; HSPI\_CLK*
- f. *IO12 : GPIO12; HSPI\_MISO*
- g. *IO13: GPIO13; HSPI\_MOSI; UART0\_CTS*
- h. *VCC*: Catu daya 3.3V (*VDD*)
- i. *CS0 :Chip selection*
- j. *MISO : Slave output, Main input.*
- k. *IO9 : GPIO9*
- l. *IO10 GBIO10*
- m. *MOSI: Main output slave input*
- n. *SCLK: Clock*
- o. *GND: Ground*
- p. *IO15: GPIO15; MTDO; HSPICS; UART0\_RTS*
- q. *IO2 : GPIO2;UART1\_TXD*
- r. *IO0 : GPIO0*
- s. *IO4 : GPIO4*
- t. *IO5 : GPIO5*

u. *RXD* : *UART0\_RXD*; *GPIO3*

v. *TXD* : *UART0\_TXD*; *GPIO1*

Untuk tegangan kerja *ESP8266* menggunakan standar tegangan *JEDEC* (tegangan 3.3V) untuk bisa berfungsi. Tidak seperti mikrokontroler *AVR* dan sebagian besar board *Arduino* yang memiliki tegangan *TTL 5 volt*. Meskipun begitu, *NodeMCU* masih bisa terhubung dengan 5V namun melalui *port micro USB* atau pin *Vin* yang disediakan oleh board-nya.

### 5. *Sensor Ultrasonik*

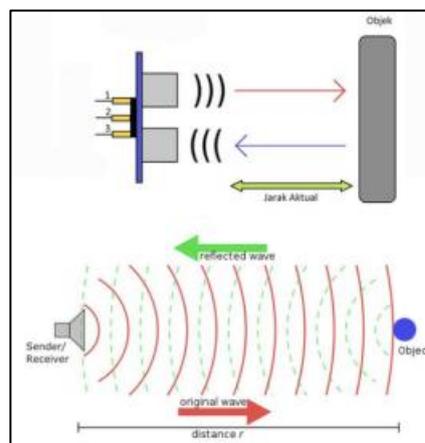


Gambar 2.2 Sensor Ultrasonik

Sensor ultrasonik adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik).

Gelombang ultrasonik adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi ultrasonik tidak dapat didengar oleh telinga manusia. Bunyi ultrasonik dapat didengar oleh anjing, kucing, kelelawar, dan lumba lumba. Bunyi ultrasonik bisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi ultrasonik di permukaan zat padat hampir sama dengan reflektivitas bunyi ultrasonik di permukaan zat cair. Akan tetapi, gelombang bunyi ultrasonik akan diserap oleh tekstil dan busa.

Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan *piezoelektrik* dengan frekuensi tertentu. *Piezoelektrik* ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima.

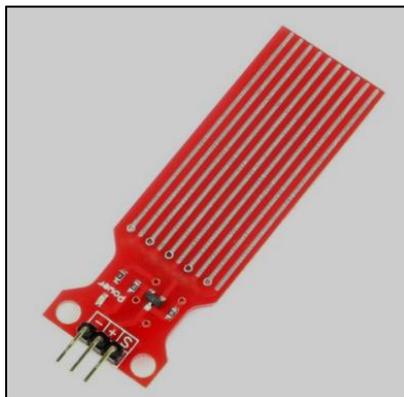


Gambar 2.3 Cara kerja sensor ultrasonik

Secara detail, cara kerja sensor ultrasonik adalah sebagai berikut:

- a. Sinyal dipancarkan oleh pemancar ultrasonik dengan frekuensi tertentu dan dengan durasi waktu tertentu. Sinyal tersebut berfrekuensi diatas 20 kHz. Untuk mengukur jarak benda (sensor jarak), frekuensi yang umum digunakan adalah 40 kHz.
- b. Sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan sekitar 340 m/s. Ketika menumbuk suatu benda, maka sinyal tersebut akan dipantulkan oleh benda tersebut.
- c. Setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut.

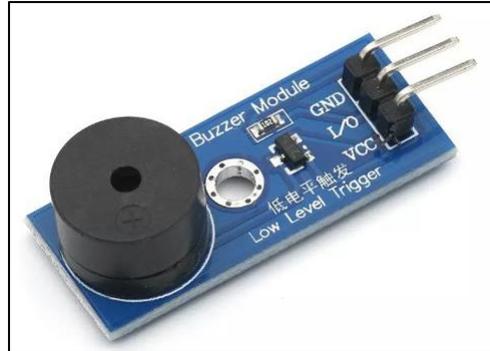
#### 6. *Water Level Sensor*



Gambar 2.4 Water Level Sensor

*Sensor Water Level* adalah sensor ketinggian air yang murah dan mudah digunakan. Sensor ini terdiri sejumlah garis yang disusun paralel untuk menentukan ketinggian permukaan air. Nilai konversi ketinggian air ke sinyal analog yang dihasilkan dapat langsung dibaca board *Arduino*.

## 7. *Buzzer*



Gambar 2.5 Buzzer

Buzzer Arduino adalah komponen yang menghasilkan suara atau bunyi ketika diaktifkan oleh mikrokontroler Arduino, digunakan untuk memberikan indikasi audio seperti alarm, notifikasi, atau peringatan dalam berbagai proyek elektronik. Buzzer ini dapat mengeluarkan nada tertentu atau sekadar bunyi sederhana, tergantung pada sinyal yang diberikan oleh Arduino.

## 8. *Flutter*

*Flutter* adalah platform yang digunakan para developer untuk membuat aplikasi *multiplatform* hanya dengan satu basis coding (*codebase*). Artinya, aplikasi yang dihasilkan dapat dipakai diberbagai *platform*, baik *mobile Android, iOS, web*, maupun *desktop*. *Flutter* memiliki dua komponen penting, yaitu, *Software Development Kit (SDK)* dan juga *framework user interface*.

- a. *Software Development Kit (SDK)* merupakan sekumpulan *tools* yang berfungsi untuk membuat aplikasi supaya bisa dijalankan di berbagai platform.

- b. *Framework UI* merupakan komponen *UI*, seperti teks, tombol, navigasi, dan lainnya, yang dapat Anda kustomisasi sesuai kebutuhan.

*Flutter* juga merupakan platform yang gratis dan *open source*. Jika Anda ingin menggunakan *Flutter*, Anda perlu mempelajari bahasa pemrograman *Dart*. Berbeda dengan *framework front-end* pada umumnya yang menggunakan *JavaScript* sebagai bahasa pemrogramannya. Sebelum resmi diluncurkan pada Desember 2018, *Flutter* sebenarnya telah dikembangkan oleh *Google* sejak 2015. Lalu, di tahun 2019, popularitas *Flutter* mulai meroket dan banyak *developer* berbondong-bondong menggunakan *Flutter*.

## 9. *Dart*

*Dart* adalah sebuah bahasa pemrograman yang dikembangkan oleh *Google* dan merupakan bahasa pemrograman resmi untuk *Flutter*, sebuah *UI toolkit* dan aplikasi *multiplatform* dari *Google*. *Flutter* sendiri telah digunakan oleh berbagai perusahaan besar seperti *Google*, *Alibaba.com*, dan *Tencent* karena dapat menghemat waktu dan tenaga dengan cara cukup membutuhkan satu *codebase* untuk mengembangkan aplikasi di berbagai platform, daripada harus menghabiskan waktu untuk membuat *codebase* terpisah untuk masing-masing *platform*.

- a. *Dart* merupakan bahasa yang wajib dikuasai untuk mengembangkan aplikasi *Flutter*.
- b. Dengan *Dart*, hanya perlu satu *codebase* untuk dapat mengembangkan aplikasi pada berbagai platform seperti *web*, *Android*, dan *iOS*.

- c. Bahasa *Dart* dirancang supaya familier dengan bahasa pemrograman lain sehingga mudah bagi yang sudah mengerti bahasa pemrograman lain maupun bagi yang baru memulai perjalanannya sebagai *developer*.
- d. Penggunaan bahasa *Dart* itu gratis (*open source*) dan dikembangkan oleh komunitas *developer* ahli yang aktif dan terbuka.
- e. *Dart* adalah bahasa yang dioptimalkan untuk pengembangan *UI* secara cepat dan produktif pada banyak *platform*.

## 10. *Node Js*

*Node.js* adalah *runtime environment* untuk *JavaScript* yang bersifat *open-source* dan *cross-platform*. Dengan *Node.js* kita dapat menjalankan kode *JavaScript* di mana pun, tidak hanya terbatas pada lingkungan *browser*.

*Node.js* menjalankan *V8 JavaScript engine* (yang juga merupakan inti dari *Google Chrome*) di luar *browser*. Ini memungkinkan *Node.js* memiliki performa yang tinggi.

*Node.js* juga menyediakan banyak *library/module JavaScript* yang membantu menyederhanakan pengembangan aplikasi web. Berikut ini adalah beberapa fitur penting dari *Node.js* yang menjadikannya pilihan utama dalam pengembangan aplikasi:

### a. *Asynchronous & Event-driven*

Semua *API* dari *Node.js* bersifat *asynchronous*, artinya tidak memblokir proses lain sembari menunggu satu proses selesai. *Server Node.js* akan melanjutkan ke ke pemanggilan *API* berikutnya lalu memanfaatkan

mekanisme event notification untuk mendapatkan respon dari panggilan *API* sebelumnya.

b. *Very Fast*

Eksekusi kode dengan *Node.js* sangat cepat karena berjalan pada *V8 JavaScript Engine* dari *Google Chrome*.

c. *Single Threaded but Highly Scalable*

*Node.js* menggunakan model *single thread* dengan *event looping*. Mekanisme ini membantu *server* untuk merespon secara *asynchronous* dan menjadikan *server* lebih scalable dibandingkan server tradisional yang menggunakan banyak thread untuk menangani permintaan.

*Node.js* dirancang untuk aplikasi dengan proses *I/O* yang intensif seperti *network server* atau *backend API*. Pemrograman dengan *multithreading* relatif lebih berat dan sulit untuk dilakukan. Jika kita ingin membuat web *server* yang bisa menangani ratusan *request* bersamaan, menggunakan ratusan *thread* akan membutuhkan memori yang besar. Oleh karena itu, karakteristik *Node* yang *asynchronous* dan *single thread* dirancang untuk memungkinkan implementasi *server* yang dapat menangani banyak *request* pada waktu yang sama.

Setelah berhasil menjadi *JavaScript Runtime* yang dapat mengeksekusi kode *JavaScript* di luar *browser*, *Node.js* seolah-olah menjadi gerbang bagi para *JavaScript Developer* untuk mengembangkan sistem di luar dari *browser*. Dengan adanya *Node.js*, *JavaScript* menjadi bahasa multiplatform yang banyak menggiring developer untuk menggunakannya.

Popularitas *JavaScript* pun meroket, *JavaScript* menjadi salah satu pilihan tepat dalam membangun *web server*, terlebih bila Anda adalah seorang *Front-End Web Developer*. Anda tentu tidak perlu menggunakan bahasa yang berbeda dalam membangun *Back-End*. Anda bisa menjadi *Full-Stack Developer* dengan mempelajari satu bahasa pemrograman saja.

#### 11. *Javascript*

*Javascript* merupakan salah satu bahasa pemrograman *web* yang bisa membuat halaman *website* anda lebih menarik serta lebih hidup. Bahasa pemrograman *javascript* itu dijalankan di perangkat pengunjung *website* atau situs, bukan di server jadi berbeda dengan *PHP*. *Javascript* juga dikatakan berbeda dari *HTML* dan *CSS*. *HTML* itu fungsinya untuk mengatur tampilan konten dan *CSS* mengelola *layoutnya* jadi ketiganya sangatlah berbeda. Banyak yang mengibaratkan perbandingan *HTML*, *CSS*, dan *javascript* seperti bangunan rumah, perabotan di dalamnya, serta lampu dan pintu-pintu. Namun sekarang, *javascript* juga mulai beralih kepada *crossplatform* yaitu aplikasi *multi platform* seperti ke sistem operasi *Android* dan *IOS* selain daripada *WEB* seperti sebelumnya. Perkembangan *javascript* juga memungkinkan programmer untuk membuat aplikasi desktop.

Seperti yang telah disinggung sebelumnya bahwa *javascript* yaitu bahasa pemrograman yang dikembangkan oleh *Netscape*. Proses desainnya memerlukan waktu selama sepuluh hari. Nama awal dari *javascript* yaitu *Mocha*, lalu *Mona*, *Live script*, baru kemudian *javascript*. Awalnya bahasa pemrograman *Javascript* yang dibuat bulan September 1995 ini hanya dipakai

oleh kalangan *Netscape*, fungsinya pun terbatas tidak seperti saat ini. Namun seiring berjalannya waktu, *javascript* terus dikembangkan hingga bisa seperti sekarang ini. Mulanya di tahun 1996 *javascript* disebut sebagai *ECMAScript*. Hal tersebut berlanjut hingga tahun 1998. Di tahun 1998 dimunculkan *ECMAScript 2* dan tahun 1999 muncul *ECMAScript 3*. *ECMAScript* terus dikembangkan hingga akhirnya menjadi *Javascript* yang ada saat ini. *Javascript* hampir digunakan oleh 92 persen *website* yang ada pada tahun 2016. Untuk saat ini, mungkin jumlahnya lebih dari itu. Namun terdapat *web browser* yang menawarkan pilihan bagi para pengguna untuk menentukan apakah ingin menonaktifkan atau tidak. Jika tidak diaktifkan maka beberapa fitur tidak akan bisa diproses ataupun tampil. Anda para awam pun pasti bisa memahami hal ini. Jika anda sering berselancar di internet, pasti ada *website* yang menggunakan *javascript* yang meminta anda untuk mengaktifkannya. Hal tersebut lah yang dimaksud dalam poin ini. Jadi memang ada *web browser* yang menawarkan pilihan aktif atau tidaknya *JavaScript*.

*Javascript* ialah salah satu bahasa pemrograman yang memiliki banyak kelebihan. Kelebihan yang dimaksud antara lain, lebih mudah dipelajari jika dibandingkan dengan bahasa pemrograman yang lainnya. Penanganan dan pencarian kesalahan ataupun error juga lebih mudah. Anda juga tidak membutuhkan *compiler* sebab *web browser* bisa menginterpretasikannya dengan *HTML*. Mengenai memori yang digunakan, *javascript* tergolong kecil, ringan, dan lebih cepat dibanding bahasa pemrograman yang lainnya. Kelebihan-kelebihan tersebut tentu saja

menjadikan *javascript* sebagai salah satu bahasa pemrograman favorit. Setiap hal tentu memiliki kekurangan, begitu juga dengan *javascript*.

## 12. **JSON (Javascript Object Notation)**

*JSON (JavaScript Object Notation)* adalah format pertukaran *data (lightweight data interchange format)*, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh *komputer*. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar *ECMA-262 Edisi ke-3 –Desember 1999*. *JSON* merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga *C* termasuk *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python* dll. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran-data (H 2014).

Berikut ini beberapa penerapan *JSON*:

1. *JSON* sebagai format untuk bertukar data client dan server atau antar aplikasi. Contoh: *RESTful API*
2. *JSON* sebagai tempat menyimpan data, contoh: *Database MongoDB*
3. *JSON* digunakan untuk menyimpan konfigurasi *project*, contoh: file *composer.json* pada *project JS* dan *package.json* pada *Nodejs*
4. *JSON* digunakan untuk menyimpan konfigurasi dan penyimpanan data pada *Hugo*
5. *JSON* digunakan untuk menyimpan konfigurasi *project* pada *Nodejs*
6. *JSON* digunakan untuk menyimpan data *manifest*

### 13. *UML (Unified Modelling Language)*

UML (*Unified Modelling Language*) menurut menurut Adi nugroho (2010:6), “*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek”. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Metode *Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu:

Sesuatu (*things*)

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

- a. *Structural things*, bagian yang relatif statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.
- b. *Behavioral things*, bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang waktu
- c. *Grouping things*, bagian pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsistem-subsistem.
- d. *Notational things*, merupakan bagian yang memperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap elemen.

### Relasi (*relationship*)

Ada empat *relationship* (hubungan) dalam *Unified Modelling Language* (UML):

- a. Ketergantungan (*dependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemen independent akan mempengaruhi elemen yang bergantung padanya.
- b. Asosiasi adalah apa dan bagaimana yang menghubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
- c. Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada di atasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisasi sedangkan arah sebaliknya dinamakan generalisasi..
- d. Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.

Diagram, *Unified Modelling Language* (UML) menyediakan Sembilan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya (statis dan dinamis).

#### 1. diagram *use case*

Diagram *use case* menyajikan interaksi antara *use case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang di bangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai. ( *Sholih, 2006*)

Adapun simbol-simbol *Use Case Diagram* antara lain :

**Tabel 2.1** *Simbol Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

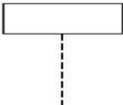
**Tabel 2.2** *Simbol Use Case Diagram*

<b>NO</b>	<b>GAMBAR</b>	<b>NAMA</b>	<b>KETERANGAN</b>
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

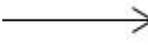
**Tabel 2. 3** *Simbol Class Diagram*

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

**Tabel 2. 4** *Simbol Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi

**Tabel 2. 5** *Simbol StateChart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

**Tabel 2. 6** *Simbol Activity Diagram*

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

#### 14. *Black Box Testing*

*Black Box Testing* adalah metode pengujian perangkat lunak di mana struktur internal/desain/implementasi item yang diuji tidak diketahui oleh penguji. Hanya desain dan struktur luar yang diuji.

Pengujian kotak hitam terutama difokuskan pada pengujian fungsionalitas perangkat lunak, memastikan bahwa perangkat lunak tersebut memenuhi persyaratan dan spesifikasi. Pengujian *black box* tidak memerlukan pengetahuan apa pun tentang cara kerja internal perangkat lunak, dan dapat dilakukan oleh penguji yang tidak terbiasa dengan bahasa pemrograman.

Pengujian *black box* menggunakan metode seperti partisi kesetaraan, analisis nilai batas, dan tebakan kesalahan untuk membuat kasus pengujian. Pengujian *black box* mudah digunakan, tidak memerlukan pengetahuan pemrograman dan efektif dalam mendeteksi masalah fungsional. Namun,

mungkin ada beberapa cacat internal penting yang tidak terkait dengan fungsionalitas.

**Tabel 2.3** Contoh Pengujian *Black Box*

<i>E-mail</i>	Benar	Benar
<i>Password</i>	Salah	Benar
<b>Hasil</b>	<i>E-mail</i> benar tapi <i>password</i> salah, tetap di halaman <i>log in</i>	<i>E-mail</i> dan <i>password</i> benar, diarahkan ke halaman Beranda

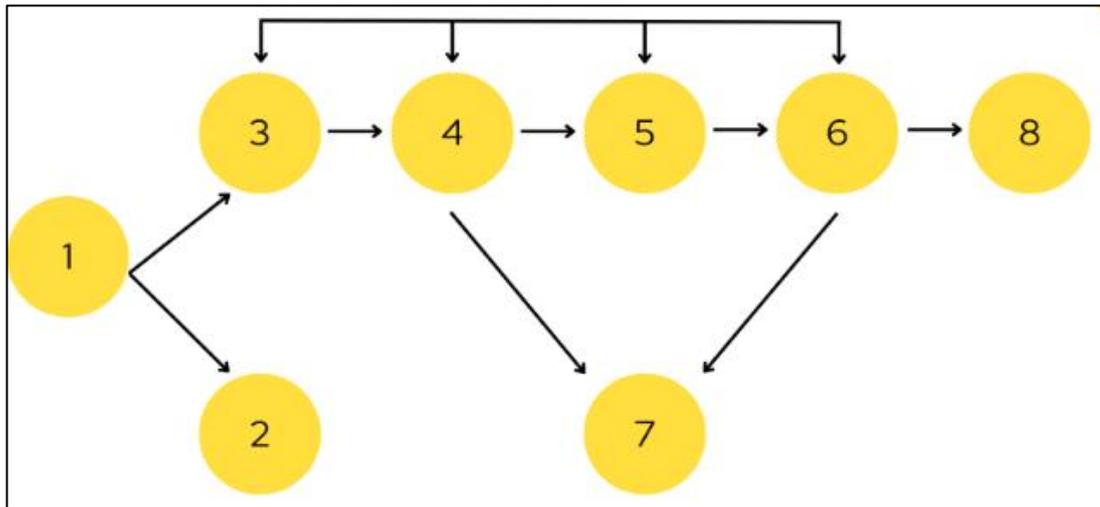
Diatas merupakan contoh pengujian *black box* dimana jika email dan password yang dimasukkan benar, maka diarahkan ke beranda, tapi jika email atau password yang dimasukkan salah, maka akan tetap di halaman login.

### 15. *White Box Testing*

*White Box Testing* adalah metode pengujian perangkat lunak di mana struktur internal/desain/implementasi item yang diuji diketahui oleh penguji. Implementasi dan dampak kode ini diuji. Tujuan *White Box* terutama difokuskan untuk memastikan bahwa kode internal perangkat lunak sudah benar dan efisien.

*White box* memerlukan pengetahuan tentang bahasa pemrograman, arsitektur perangkat lunak, dan pola desain. *White Box* digunakan untuk menguji perangkat lunak pada level unit, level integrasi, dan level sistem.

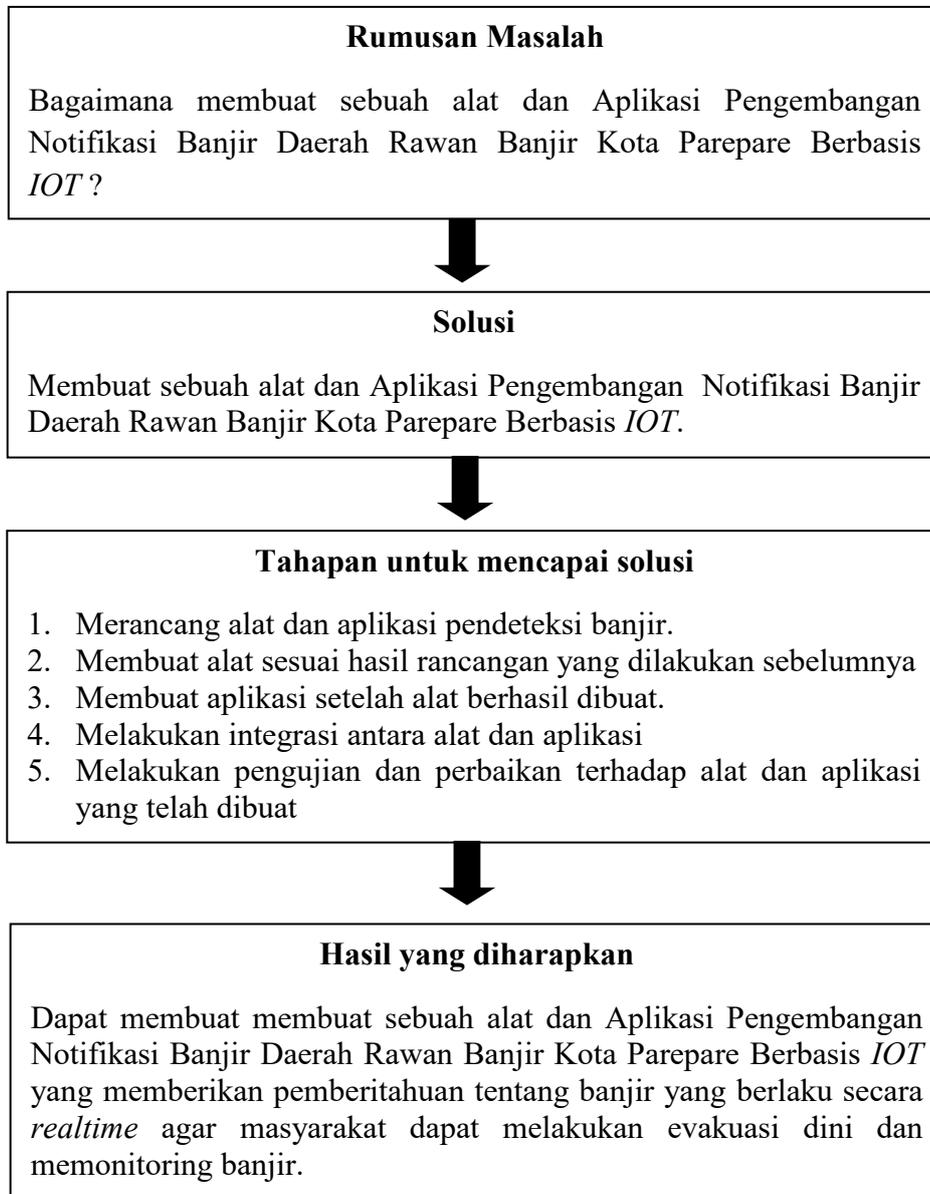
*White Box* efektif dalam mendeteksi cacat internal, dan memastikan bahwa kode tersebut efisien dan dapat dipelihara. Namun, hal ini memerlukan pengetahuan pemrograman dan dapat memakan waktu.



**Gambar 2.6** Contoh Pengujian *White Box*

Diatas merupakan contoh pengujian *white box* , yang mana berdasarkan diagram di atas, kemungkinan jalur yang dilewati antara lain:

- a. 1-2
- b. 1-3-4-5-6-8
- c. 1-3-4-5-6-7
- d. 1-3-4-5-6-8 dan sebagainya.

16. *Kerangka pikir*

Gambar 2.7 Kerangka Pikir

## B. Kajian Hasil Penelitian Terdahulu

1. **“Aplikasi Pemberitahuan Banjir pada Daerah Parepare Menggunakan *Google Maps* Berbasis Android)”** oleh Andi Adriani Paesal, Muhammad Basri dan Marlina, Universitas Muhammadiyah Parepare, Program Studi Teknik Informatika, Januari 2021 (Andi Adriani Paesal, Muhammad Basri and Marlina 2021), NIM : 215280110. Kesimpulan dari penelitian ini adalah :

Telah berhasil disusun sebuah Aplikasi pemberitahuan banjir pada daerah parepare menggunakan *google maps* berbasis *android*, yang selanjutnya dapat dilanjutkan pada proses pengembangan dan implementasi sistem deteksi banjir berbasis *IoT* dimana sistem yang telah dibangun akan diuji dengan menggunakan metode *Blackbox*. Sehingga diharapkan dengan adanya sistem ini akan dapat membantu masyarakat dalam menekan kerugian yang mungkin terjadi akibat bencana banjir.

Persamaan antara penelitian diatas dengan penulis adalah antisipasi kenaikan air pada wilayah tertentu.

Perbedaan antara penelitian diatas dengan penulis adalah penelitian diatas terfokus kepada deteksi banjir manakala penulis terfokus kepada notifikasi banjir pada daerah Parepare.

2. **“Sistem Monitoring Cuaca Dan Peringatan Banjir Berbasis *IoT* Dengan Menggunakan Aplikasi *MIT APP Inventor*”** oleh Muh Bahrul Ulum, dan Fawaidul Badri (Ulum and Badri 2023), Universitas Islam Malang, Jurnal Informatika dan Teknik Elektro Terapan, Volume 11, No 3, Agustus 2023, eISSN 2830-7062. Kesimpulan dari penelitian ini adalah:

Perangkat yang dibuat dapat berfungsi dengan baik setelah berbagai proses uji coba, sehingga memungkinkan untuk diimplementasi di lapangan. Sensor ultrasonik digunakan untuk mengukur ketinggian air pada titik banjir, sedangkan Led dan buzzer digunakan sebagai indikator tingkat bahaya sesuai dengan ketinggian air pada titik banjir, Sensor hujan digunakan untuk memonitoring intensitas hujan di sekitar titik banjir. Terdapat tiga tahap intensitas ketinggian air (Aman, Siaga, Bahaya), dan hujan (Cerah, Hujan Gerimis, Hujan Lebat).

Persamaan antara penelitian diatas dengan penulis adalah antisipasi kenaikan air pada wilayah tertentu.

Perbedaan antara penelitian diatas dengan penulis adalah penelitian diatas terfokus kepada deteksi banjir manakala penulis terfokus kepada notifikasi banjir pada daerah Parepare.

3. **“Rancang Bangun Sistem Monitoring Peringatan Dini Bencana Banjir Berbasis *IOT* Menggunakan Protokol *MQTT* Dengan Notifikasi *Bot Telegram*”** (Bustomi 2021). Kesimpulan dari penelitian ini adalah :

Penulis telah melakukan uji coba alat tersebut sebanyak 5 kali percobaan agar dapat dipastikan bahwa alat berfungsi dengan baik, masyarakat dapat melihat ketinggian air melalui website pada halaman grafik air serta dapat melihat lokasi alat tersebut dipasang. Alat peringatan dini bencana banjir juga dapat memberikan informasi peringatan melalui bot Telegram secara otomatis dalam jangka waktu 1 jam, masyarakat dapat mengirim perintah chat (Status) ke bot Telegram, maka sistem akan

memberikan informasi langsung kepada masyarakat ketinggian air pada saat itu. Alat tersebut juga dilengkapi dengan sistem peringatan buzzer atau sirine serta lampu kondisi level ketinggian air agar masyarakat dapat terbantu dengan peringatan tersebut.

Sistem informasi peringatan dini bencana banjir ini semoga dapat memitigasi bencana banjir sehingga dapat membantu masyarakat serta pemerintah untuk meminimalisir kerugian materi maupun korban jiwa akibat bencana banjir.

Persamaan antara penelitian diatas dengan penulis adalah antisipasi kenaikan air pada wilayah tertentu.

Perbedaan antara penelitian diatas dengan penulis adalah penelitian diatas terfokus kepada deteksi banju manakala penelis terfokus kepada notifikasi banjir pada daerah Parepare.

## **BAB III**

### **METODE PENELITIAN**

#### **A. Jenis Penelitian**

Adapun jenis dan sumber data yang dibutuhkan dalam penelitian ini adalah sebagai berikut:

1. Penelitian Kepustakaan

Yaitu Pengumpulan data dengan membaca buku mengenai literature dan buku lain yang bersifat ilmiah yang berkaitan dengan materi pembahasan seperti :

- a) *IOT*
- b) *ESP 8266*
- c) *Sensor Ultrasonik*
- d) *Sensor Water Level*
- e) *Flutter*

2. Studi Literatur

Pengumpulan data dengan cara mengumpulkan literatur, jurnal paper dan bacaan-bacaan yang ada kaitannya dengan judul penelitian seperti *IoT, Evakuasi, Banjir, Open CV..*

## B. Lokasi Dan Waktu

### 1. Lokasi

Kegiatan penelitian ini dilakukan di Kota Parepare, Kabupaten/Kota Parepare, Provinsi Sulawesi Selatan.

### 2. Waktu

Waktu yang dipergunakan untuk pelaksanaan penelitian ini adalah kurang lebih dua (2) bulan dengan perencanaan sebagai berikut.

**Tabel 3.1** Waktu Penelitian

Pengumpulan data , studi literatur	2 (dua) Minggu
Perancangan dan konstruksi alat	2 (dua) Minggu
Pembuatan Alat	2 (dua) Minggu
Penulisan <i>syntax</i>	1 (dua) Minggu
Pengujian dan finalisasi alat	1 (dua) Minggu

### C. Alat Dan Bahan

Dalam melakukan penelitian, penulis memerlukan alat dan bahan penelitian yang mendukung kegiatan penelitian tersebut.

#### 1. Alat

**Tabel 3.2** Spesifikasi Perangkat Keras Komputer

Sistem Operasi	<i>Windows 10 Pro 64 Bit</i>
Processor	<i>Intel Core 2 Duo E8500</i>
RAM	<i>8 GB</i>
Harddisk	<i>SSD 240 GB</i>
Display	<i>LCD CMV 745 A &amp; LG 22 Inch</i>
Other Peripheral	<i>Mouse dan Keyboard External</i>

**Tabel 3.3** Spesifikasi Perangkat Lunak Komputer

1	<i>Visual Studio Code 1.63.0</i>
2	<i>Diagram Net Desktop</i>

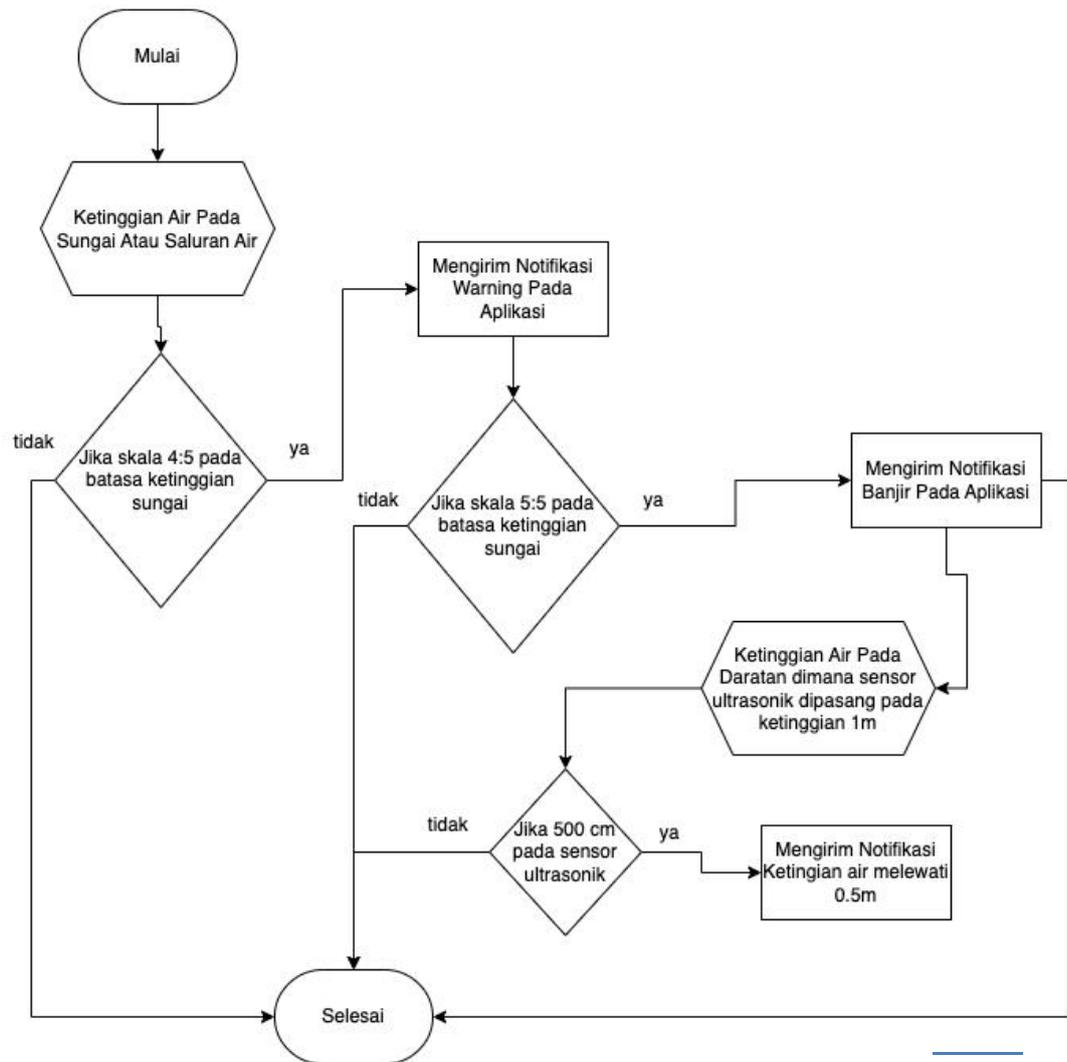
#### 2. Bahan

**Tabel 3.4** Bahan yang digunakan

No	Uraian	Jumlah
1	<i>ESP 8266</i>	1 buah
2	<i>Sensor Ultrasonik HC-SRC04</i>	1 buah
3	<i>Water Level Sensor</i>	2 buah
4	<i>Buzzer</i>	1 buah

### D. Rancangan Penelitian

Dibawah merupakan sistem yang diusulkan oleh penulis.

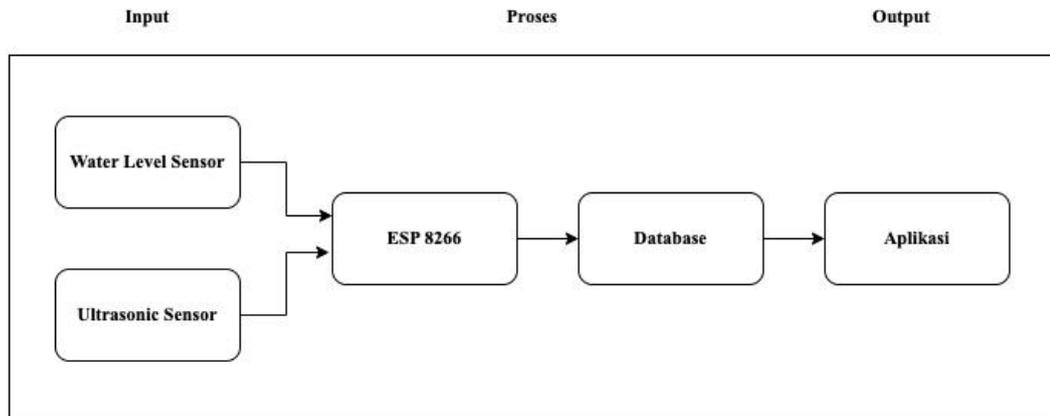


Gambar 3.1 Rancangan Penelitian

**Tabel 3.5** Penjelasan Rancangan Sistem

<b>Flow chart</b>	<b>Penjelasan</b>
Mulai	Sistem dimulai
Ketinggian air pada sungai atau saluran air	Merupakan pendeklarasian variable dimana nilai yang akan diambil adalah ketinggian air pada sungai atau saluran air
Jika 500 cm pada batas ketinggian sungai	Menjelaskan Jika 500 cm pada batas ketinggian sungai
Mengirim notifikasi pada aplikasi	Menjelaskan sistem akan mengirim notifikasi warning pada aplikasi
Jika melewati batas ketinggian sungai atau saluran air	Menjelaskan jika melewati batas ketinggian sungai atau saluran air
Mengirim notifikasi banjir pada aplikasi	Menjelaskan sistem akan mengirim notifikasi banjir pada aplikasi
Ketinggian air pada daratan dimana sensor ultrasonik dipasang pada ketinggian 1m	Pendeklarasian variable dimana nilai yang diambil adalah ketinggian air pada daratan dimana sensor ultrasonik dipasang pada ketinggian 1m
Jika 500 cm pada sensor ultasonik	Menjelaskan jika air banjir 500 cm pada sensor ultrasonik
Mengirim Notifikasi ketinggian air melewati 0.5m	Menjelaskan sistem akan mengirim Notifikasi ketinggian air melewati 0.5m
Selesai	Sistem Selesai

### E. Blok Diagram Sistem



**Gambar 3.2** Blok Diagram Sistem

Diatas merupakan blok diagram sistem, dimana nilai yang dimasukkan adalah ketinggian air pada sungai yang didapatkan oleh sensor water level dan sensor ultrasonik, kemudian nilai ini akan dikirim ke server yang akan diproses, jika nilai mencapai nilai tertentu, maka sistem akan mengirim notifikasi kepada aplikasi yang mana merupakan output daripada sistem.

### F. Teknik Pengumpulan Data

#### 1. Studi Literatur

Pengumpulan data dengan cara mengumpulkan literatur, jurnal *paper* dan bacaan-bacaan yang ada kaitannya dengan judul penelitian.

### G. Teknik Analisis Data

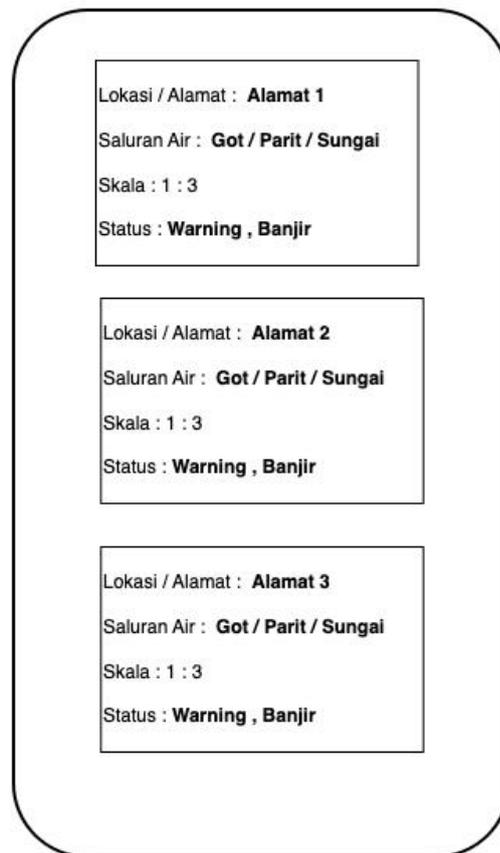
Tahap-tahap analisis data yang dimaksudkan dalam penelitian ini adalah berkenaan dengan proses pelaksanaan penelitian.

1. Tahap pengumpulan data. Pada tahap ini dilakukan pengumpulan data-data yang akan diperlukan pada perancangan alat pendeteksi banjir berbasis *IoT*.

2. Tahap analisis data. Pada tahap ini dilakukan analisis terhadap data-data yang sudah dikumpulkan pada tahap sebelumnya, menganalisis cara kerja aplikasi yang akan dirancang, mengidentifikasi masalah, dan menganalisa kebutuhan alat.
3. Tahap perancangan. Pada Tahap ini dilakukan perancangan diagram *UML* dan perancangan output dari aplikasi yang akan dibuat.
4. Tahap pembuatan aplikasi. Pada tahap ini aplikasi dibangun dengan menggunakan *text editor Visual Studio Code*. Pengimplementasian *IoT* dan *OpenCV* pada kontroller dan sensor.
5. Tahap Pengujian. Pada tahap ini alat yang telah dibangun diuji tingkat keberhasilannya.
6. Tahap Implementasi. Pada tahap ini dilakukan setelah seluruh tahap selesai dilakukan.

## H. Desain Antaramuka Aplikasi

### Tampilan Utama Aplikasi



Gambar 3.3 Tampilan Utama Aplikasi

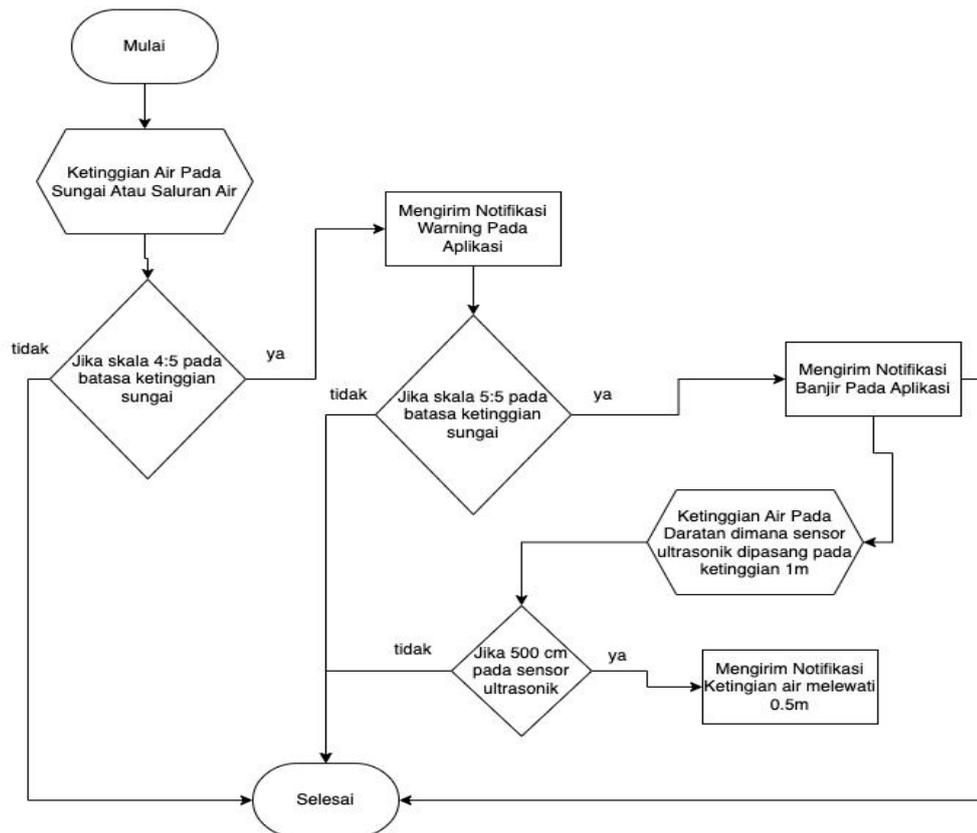
Gambar 3.3 diatas merupakan tampilan utama aplikasi dimana *main menu* merupakan menu utama dimana info-info penting ditampilkan manakala *navigation bar* merupakan tampilan navigasi aplikasi.

## BAB IV

### HASIL DAN PEMBAHASAN

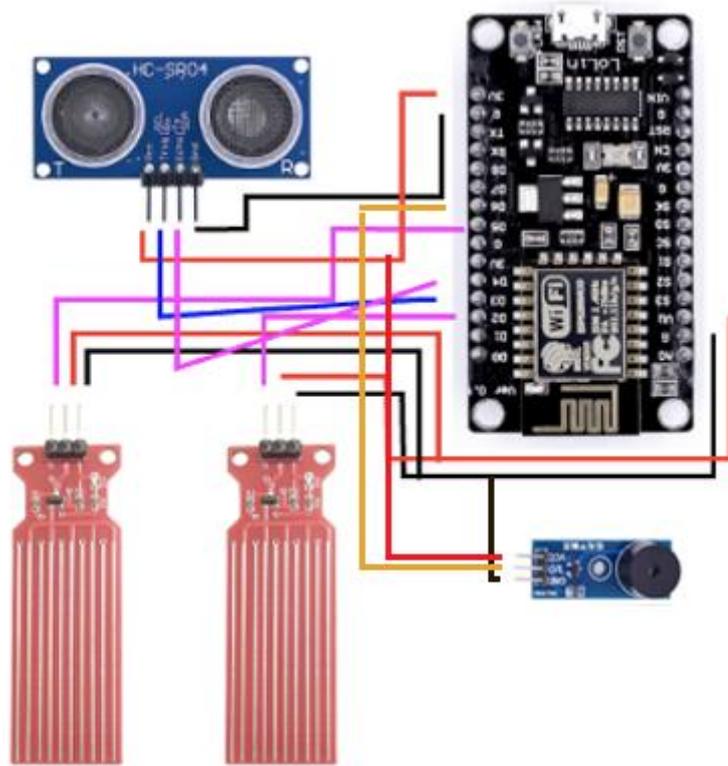
#### A. Flowchart Sistem

*Flowchart* merupakan diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah. Berikut dibawah merupakan *flowchart* sistem penelitian :



**Gambar 4.1** *Flowchart Sistem*

## B. Desain Konstruksi



**Gambar 4.2** Desain Konstruksi

Diatas merupakan konstruksi alat dimana terdapat dua buah *sensor water level* yang mana *water level* yang pertama diletakkan pada skala 4:5 (Ketinggian air 6 cm) saluran air manakala *water level* kedua diletakkan pada batas atau skala 5:5 (Ketinggian air 9 cm) saluran air. *Sensor ultrasonic* pula diletakan pada ketinggian 21 cm dari batas permukaan tanah menghitung ketinggian banjir.

Dibawah merupakan tabel koneksi kabel antara *esp8266* dan *module sensore*

**Tabel 4.1** *Koneksi Esp8266 ke Ultrasonic*

<b>ESP8266</b>	<b>Sensor Ultrasonic</b>
<i>Gnd</i>	<i>Gnd</i>
<i>3v3</i>	<i>Vcc</i>
<i>D3</i>	<i>Trig</i>
<i>D4</i>	<i>Echo</i>

**Tabel 4.2** *Koneksi Esp8266 ke Water Level*

<b>ESP8266</b>	<b>Water Level 1</b>	<b>Water Level 2</b>
<i>Gnd</i>	-	-
<i>VV</i>	+	+
<i>D1</i>	<i>S</i>	
<i>D2</i>		<i>S</i>

**Tabel 4.3** *Koneksi Esp8266 ke Buzzer*

<b>ESP8266</b>	<b>Buzzer</b>
<i>Gnd</i>	<i>Gnd</i>
<i>D6</i>	<i>Vcc</i>
<i>3v3</i>	<i>Trig</i>
<i>D4</i>	<i>Echo</i>

### C. Tampilan Alat

Dibawah ini merupakan tampilan alat dan aplikasi yang selesai dirancang dan dibuat oleh penulis :



**Gambar 4.3** Tampilan Alat

## D. Tampilan Aplikasi

### 1. Halaman *Splashscreen*



**Gambar 4.4** *Tampilan Splashscreen*

Diatas merupakan halaman *splashscreen* yang mana menampilkan *logo* dan juga nama aplikasi

## 2. Halaman Monitoring



**Gambar 4.5** *Halaman Monitoring*

Diatas merupakan halaman monitoring, disini pengguna dapat melihat kondisi air secara *realtime*. Aplikasi menampilkan status, *warning level*, *danger level* dan ketinggian air.

### 3. Halaman Log Data



Waktu	Status	Ketinggian Air
2024-08-10 05:46:50	Air Dalam	-
2024-08-10 05:46:50	Tahap Normal	-
2024-08-10 05:46:20	Air Dalam	-
2024-08-10 05:46:20	Tahap Normal	-
2024-08-10 05:45:50	Air Dalam	-
2024-08-10 05:45:50	Tahap Normal	-
2024-08-10 05:43:53	Air Dalam	-
2024-08-10 05:43:53	Tahap Normal	-
2024-08-10 05:43:23	Air Dalam	-
2024-08-10 05:43:23	Tahap Normal	-
2024-08-10 05:42:38	Air Dalam	-
2024-08-10 05:42:38	Tahap Normal	-
2024-08-10 05:42:07	Air Dalam	-
2024-08-10 05:42:07	Tahap Normal	-
2024-08-10 05:41:37	Air Dalam	-
2024-08-10 05:41:37	Tahap Normal	-
2024-08-09 00:37:35	Air Dalam	-
2024-08-09 00:37:35	Tahap Normal	-
2024-08-09 00:37:04	Air Dalam	-
2024-08-09 00:37:04	Tahap Normal	-
2024-08-09 00:36:34	Air Dalam	-
2024-08-09 00:36:34	Tahap Normal	-
2024-08-09 00:36:04	Air Dalam	-
2024-08-09 00:36:04	Tahap Normal	-
2024-08-09 00:35:34	Air Dalam	-
2024-08-09 00:35:34	Tahap Normal	-
2024-08-09 00:35:03	Air Dalam	-
2024-08-09 00:35:03	Tahap Normal	-
2024-08-09 00:34:33	Air Dalam	-
2024-08-09 00:34:33	Tahap Normal	-
2024-08-09 00:34:03	Air Dalam	-
2024-08-09 00:34:03	Tahap Normal	-
2024-08-09 00:33:32	Air Dalam	-
2024-08-09 00:33:32	Tahap Normal	-
2024-08-09 00:33:02	Air Dalam	-
2024-08-09 00:33:02	Tahap Normal	-
2024-08-09 00:32:32	Air Dalam	-
2024-08-09 00:32:32	Tahap Normal	-
2024-08-09 00:32:01	Air Dalam	-
2024-08-09 00:32:01	Tahap Normal	-

**Gambar 4.6** *Halaman Log Data*

Diatas merupakan halaman log data. Pada halaman ini ditampilkan data-data sebelumnya yang disimpan oleh sistem selama dua menit jika air pada tahap normal. Jika tahap *warning* maka sistem akan menyimpan data selama satu menit. Jika dalam tahap *danger* maka sistem akan menyimpan data selama 30 menit secara berkala.

### E. Kamus Data

Kamus data adalah kumpulan yang berisi informasi atau disimpan dalam bentuk basis data. Disini peneliti menamakan basis data “*db\_flood*” dengan tabel “*tb\_data*” yang berisi informasi tentang ketinggian air, waktu dan status. Berikut di bawah merupakan deskripsi tabel.

**Tabel 4.4** *tb\_data*

Field	Type	Length	Attribute
<i>no</i>	<i>int</i>	4	<i>primary key auto increment</i>
<i>water_height</i>	<i>varchar</i>	5	
<i>status</i>	<i>int</i>	1	
<i>created_at</i>	<i>datetime</i>		

### F. Proses Deteksi Banjir

Disini peneliti akan menjelaskan proses bagaimana deteksi banjir terjadi pada alat dan aplikasi.

1. Ketinggian air secara terus menerus dipantau dengan menggunakan sensor *ultrasonic*.
2. Jika air mengenai atau menenggelamkan sensor *water level* yang berada pada skala 4:5 (ketinggian 6 cm) saluran air, maka sensor akan mengeluarkan *beep* dan berstatus *warning*. Dan *esp8266* akan mengirimkan data ke server yang nantinya akan ditampilkan di aplikasi.
3. Jika air mengenai atau menenggelamkan sensor *water level* yang berada pada skala 5:5 (ketinggian 9 cm) saluran air, maka sensor akan mengeluarkan *beep* dan berstatus *danger*. Dan *esp8266* akan mengirimkan data ke server yang nantinya akan ditampilkan di aplikasi dan juga ketinggian air secara *realtime*.

## G. Pengujian Alat

### 1. Air Ketinggian Normal

Alat	Aplikasi
	 <p>The screenshot shows a mobile application window titled 'redroid11_x86_64' with a green header 'Real Time'. Below the header is a large green box with the text 'Notif Banjir' and an icon of a house with rain. Underneath, a green box displays the following information:</p> <ul style="list-style-type: none"> <li>Status : Normal</li> <li>Warning Level : Air Belum Mencapai Tahap Warning</li> <li>Danger Level : Air Belum Mencapai Tahap Danger</li> <li>Water Height : -</li> </ul> <p>The bottom of the screen shows a green navigation bar with a home icon, the text 'Real Time', and a list icon, along with the standard Android navigation bar.</p>
<p>Diatas merupakan pengujian dengan ketinggian air normal atau belum ada kenaikan air. Disini terlihat bahwa aplikasi menunjukkan status normal.</p>	

## 2. Ketinggian Air status *Warning* (ketinggian 6 cm)

Alat	Aplikasi
	 <p>The screenshot shows the application interface with a green header 'Real Time'. Below it, a notification titled 'Notif Banjir' is displayed. The notification includes a house icon and the following text: 'Status : Peringatan Banjir, Air Dalam Skala 4:5 atau lebih', 'Warning Level : Air Mencapai Tahap Warning', 'Danger Level : Air Belum Mencapai Tahap Danger', and 'Water Height : -'. At the bottom, there is a navigation bar with a home icon, the text 'Real Time', and a list icon.</p>
<p>Diatas merupakan pengujian dengan ketinggian air <i>warning</i> atau berada di skala 4:5 (Ketinggian air 6 cm), disini terlihat bahwa aplikasi menunjukkan status <i>warning</i> dan peringatan banjir diinformasikan oleh aplikasi.</p>	

### 3. Ketinggian Air *Danger* (ketinggian 9 cm)

Alat	Aplikasi
	
<p>Diatas merupakan pengujian dengan ketinggian air <i>danger</i> atau berada pada skala 5:5 (Ketinggian air 9 cm). Disini terlihat bahwa aplikasi menunjukkan status <i>danger</i> dan peringatan bahaya banjir, serta ketinggian air diinformasikan oleh aplikasi.</p>	

## BAB V

### PENUTUP

#### A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan pada bab-bab sebelumnya, dihasilkan sebuah alat dan aplikasi Notifikasi Rawan Banjir Berbasis *IOT* yang sudah bebas dari kesalahan. Maka dapat ditarik kesimpulan seperti berikut :

1. Penelitian ini berhasil merancang dan membangun sebuah prototipe Alat dan Aplikasi Notifikasi Rawan Banjir Berbasis *IoT*.
2. Kontroller yang digunakan adalah *esp8266* yang mana terhubung kepada dua *sensor water level* dan satu *sensor ultrasonic*. *Esp8266* berfungsi untuk mengirimkan data kepada *server*.
3. Sensor yang digunakan adalah *sensor water* yang mana berfungsi untuk mengetahui ketinggian air pada saluran air dan *sensor ultrasonic* untuk mengetahui ketinggian air setelah melewati batasan saluran air..
4. Aplikasi menampilkan status notifikasi banjir dan juga ketinggian air secara *realtime*.
5. Alat akan berbunyi *beeb* dengan nada tidak beraturan ketika mengenai *sensor water level* pertama dengan ketinggian air 6 cm yang berstatus *warning*.
6. Alat akan berbunyi *beeb* panjang ketika mengenai *sensor water level* kedua dengan ketinggian air 9 cm yang berstatus *danger*.

## **B. Saran**

Berdasarkan kesimpulan yang telah dibuat, maka penulis memberikan saran pengembangan lebih lanjut yang mencakup penambahan sensor tambahan dan peningkatan daya tahan perangkat terhadap kondisi lingkungan ekstrem yang dapat lebih meningkatkan keandalan dan akurasi sistem dalam berbagai situasi banjir

## DAFTAR PUSTAKA

- Adi, A. T., & Tohir, T. (2022, August). Rancang Bangun Sistem Proteksi Rumah Tinggal Saat Terjadi Banjir Menggunakan Mikrokontroler Berbasis IoT. In *Prosiding Industrial Research Workshop and National Seminar* (Vol. 13, No. 01, pp. 633-637).
- Bustomi, Abi Yazid. 2021. "Rancang Bangun Sistem Monitoring Peringatan Dini Bencana Banjir Berbasis IOT Menggunakan Protokol MQTT Dengan Detik. 2023. *Detik SulSel*. Februari 02. <https://www.detik.com/sulsel/berita/d-6548595/data-terkini-dampak-banjir-di-parepare-5-294-jiwa-terdampak-2-meninggal#:~:text=Data%20Terkini%20Dampak%20Banjir%20di%20Parepare%3A%205.294%20Jiwa%20Terdampak%2C%202%20Meninggal,-Muhclis%20Abduh%20%2D%20detikS>.
- Dharwiyanti, Sri. 2003. *Pemodelan Visual dengan UML*. Yogyakarta: Graha Ilmu.
- H, Nazruddin Safaat. 2014. *Android: Pemrograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android (Edisi 2)*. Bandung: Informatika.
- Imaduddin, Ahmad, and Sidiq Permana. 2018. *Menjadi Android Developer Expert*. Bandung: PT. Presentologics.
- Munawar. 2005. *Permodelan Visual dengan UM*. Yogyakarta: Pustaka Pelajar.4
- Notifikasi Bot Telegram." *Rancang Bangun Sistem Monitoring Peringatan Dini Bencana Banjir Berbasis IOT Menggunakan Protokol MQTT Dengan Notifikasi Bot Telegram*.
- Paesal, Andi Adriani. 2021. "Aplikasi Pemberitahuan Banjir pada Daerah Parepare Menggunakan Google Maps Berbasis Android." *Aplikasi Pemberitahuan Banjir pada Daerah Parepare Menggunakan Google Maps Berbasis Android* (Program Studi Teknik Informatika).
- Pratama, V. S. (2023). *Sistem Monitoring Ketinggian Air Sebagai Notifikasi Peringatan Banjir Berbasis IOT* (Doctoral dissertation, Politeknik Negeri Jember).
- Wikipedia. 2021. *Google Maps*. 12 3. Accessed 12 17, 2021. [https://id.wikipedia.org/wiki/Google\\_Maps](https://id.wikipedia.org/wiki/Google_Maps).
- Windiaстик, Shania Putri, Elsha Novia Ardhana, and Joko Triono. 2019. "Perancangan Sistem Pendeteksi Banjir Berbasis IOT (Internet Of

Things)". " *Perancangan Sistem Pendeteksi Banjir Berbasis IOT (Internet Of Things)* " (Fakultas Teknik Informasi).

Ulum, Muh Bahrul, and Fawaidul Badri. 2023. "Sistem Monitoring Cuaca Dan Peringatan Banjir Berbasis IOT Dengan Menggunakan Aplikasi MIT APP Inventor." *Sistem Monitoring Cuaca Dan Peringatan Banjir Berbasis IOT Dengan Menggunakan Aplikasi MIT APP Inventor* (Jurnal Informatika Dan Teknik Elektro) 11 (3).

2021. *Flutter*. March 5. Accessed May 21, 2021. [https://id.wikipedia.org/wiki/Flutter\\_\(perangkat\\_lunak\)](https://id.wikipedia.org/wiki/Flutter_(perangkat_lunak)).