

PROPOSAL

SISTEM *MONITORING* KETINGGIAN AIR (BANJIR) MENGUNAKAN *WIRELESS SENSOR NETWORK* (WSN) DAN WEB *GIS*

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Sarjana Teknik



GAGAS ALIANSI CAKRAWALA

E1E117056

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HALU OLEO

KENDARI

2021

LEMBAR PENGESAHAN

ABSTRAK

Gagas Aliansi Cakrawala, E1E117056

Sistem *Monitoring* Ketinggian Air (Banjir) Menggunakan *Wireless Sensor Network* (WSN) Dan Web *GIS*

Proposal, Fakultas Teknik, 2021

Kata kunci: *Monitoring*, Air sungai, Sensor Jarak Ultrasonik.

Banjir merupakan bencana yang paling sering dialami masyarakat ketika terjadi musim hujan khususnya masyarakat yang tinggal di sekitaran sungai atau dataran rendah. Berdasarkan analisis kejadian banjir di kota Kendari pada tahun 2017 tingkat kesigapan rumah tangga dalam menghadapi bencana banjir dari hasil perhitungan indeks yang telah dilakukan secara umum memiliki 4 parameter dimana 2 dari 4 parameter tersebut masih menunjukkan indeks kesigapan yang cukup rendah yang dapat dilihat pada parameter sistem peringatan bencana masih mencapai indeks kesigapan 43,3 dengan tingkat kesigapan kurang siap dan parameter mobilisasi sumber daya yang memiliki indeks kesigapan mencapai 55,8 dengan tingkat kesigapan hampir siap dengan begitu indeks gabungan dari empat parameter mencapai angka 69,43. Oleh karena itu perlu adanya alat dan sistem yang dapat membantu petugas untuk dapat mengakses informasi yaitu berupa sebuah bentuk sistem *Monitoring* yang dapat melakukan pengukuran terhadap ketinggian air sungai secara *real time*, dan menampilkannya pada *website* yang dapat diakses oleh pengawas sebagai peringatan dini bahaya banjir (*Early Warning System*) sehingga dapat melakukan evakuasi dengan segera agar dapat meminimalisir kerugian bagi masyarakat. Alat yang digunakan adalah sensor ultrasonik HC-SR04, rain sensor, buzzer, node MCU, *access point*.

DAFTAR ISI

| | |
|--|-------------|
| PROPOSAL..... | i |
| LEMBAR PENGESAHAN | ii |
| ABSTRAK | iii |
| DAFTAR ISI..... | iv |
| DAFTAR GAMBAR..... | viii |
| BAB I PENDAHULUAN..... | 1 |
| 1.1. Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 2 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan Penelitian | 3 |
| 1.5 Manfaat Penelitian | 3 |
| 1.6 Sistematika Penulisan | 3 |
| 1.7 Tinjauan Pustaka..... | 4 |
| BAB II LANDASAN TEORI | 7 |
| 2.1 Sistem <i>Monitoring</i> | 7 |
| 2.2 Banjir | 7 |
| 2.3 Air | 7 |
| 2.4 <i>Wireless Sensor Network</i> (WSN)..... | 8 |
| 2.5 <i>Geographic Information System</i> (GIS) | 9 |
| 2.6 Arduino IDE | 9 |
| 2.7 Node MCU..... | 11 |
| 2.8 Sensor Ultrasonik HC-SR04..... | 12 |
| 2.9 <i>Rain sensor</i> | 12 |
| 2.10 <i>Buzzer</i> | 13 |
| 2.11 <i>Website</i> | 14 |
| 2.12 Xampp..... | 14 |

| | | |
|--------|--|----|
| 2.13 | PHP | 15 |
| 2.14 | <i>Casding Style Sheet (CSS)</i> | 16 |
| 2.15 | <i>Hypertext Markup Language (HTML)</i> | 17 |
| 2.16 | <i>Codeigniter</i> | 17 |
| 2.17 | <i>Firebase</i> | 17 |
| 2.20 | <i>Unified Modeling Language (UML)</i> | 20 |
| 2.20.1 | <i>Usecase Diagram</i> | 21 |
| 2.20.2 | <i>Activity Diagram</i> | 22 |
| 2.20.3 | <i>Class Diagram</i> | 23 |
| 2.21 | <i>Flowchart</i> | 24 |
| 3.1 | Pengumpulan Data | 27 |
| 3.2 | Metode Pengembangan Perangkat Lunak | 27 |
| 3.3 | Waktu dan Tempat Penelitian | 28 |
| 3.3.1 | Waktu Penelitian | 28 |
| 3.3.2 | Tempat Penelitian | 29 |
| 3.4 | Analisis Sistem <i>Monitoring</i> yang akan dibangun | 29 |
| 3.4.1 | Kebutuhan Fungsional | 29 |
| 3.4.2 | Analisis Kebutuhan Non Fungsional | 30 |
| 3.5 | Arsitektur Sistem | 31 |
| 3.5.1 | <i>Hardware</i> | 31 |
| 3.5.4 | <i>Output System</i> | 32 |
| 3.5.4 | <i>Software</i> | 33 |
| 3.5.5 | <i>Software Sistem</i> | 33 |
| 3.6 | Blok Diagram | 33 |
| 3.7 | Perancangan Sistem | 34 |
| 3.7.1 | <i>Flowchart</i> | 35 |
| 3.7.2 | Perancangan <i>use case diagram</i> | 35 |
| 3.7.3 | Skenario Sistem | 36 |

| | | |
|-----------------------------|---|-----------|
| 3.7.4 | Perancangan <i>Activity Diagram</i> | 37 |
| 3.7.6 | Rancangan Antar Muka Sistem (<i>Interface</i>)..... | 42 |
| 3.7.7 | Rencana Pengujian Sistem | 47 |
| DAFTAR PUSTAKA | | 50 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 2. 1 Komponen-komponen <i>use case diagram</i> | 22 |
| Tabel 2. 2 Komponen-komponen <i>activity diagram</i> | 23 |
| Tabel 2. 3 Komponen-komponen <i>class diagram</i> | 24 |
| Tabel 2. 4 komponen-komponen multiplicity | 24 |
| Tabel 2. 5 Simbol - Simbol <i>Flowchart</i> | 25 |
| Tabel 3. 1 <i>Gantt Chart</i> waktu penelitian..... | 29 |
| Tabel 3. 2 Spesifikasi Perangkat Keras | 30 |
| Tabel 3. 3 Spesifikasi Perangkat Lunak | 31 |
| Tabel 3. 4 Skenario Sistem <i>Monitoring</i> | 37 |
| Tabel 3. 5 Contoh tabel pengujian keakuratan sensor Ultrasonik HC-SR04..... | 48 |
| Tabel 3. 6 Contoh pengujian Rain Sensor..... | 49 |
| Tabel 3. 7 Contoh pengujian <i>Blackbox</i> | 49 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 3. 1 Ilustrasi <i>Hardware</i> Arsitektur Sistem | 32 |
| Gambar 3. 2 Blok Diagram | 33 |
| Gambar 3. 3 <i>Flowchart</i> | 35 |
| Gambar 3. 4 <i>Use case Diagram</i> | 36 |
| Gambar 3. 5 <i>Activity Diagram Menu Login</i> | 38 |
| Gambar 3. 6 <i>Activity Diagram Menu Monitoring</i> | 39 |
| Gambar 3. 7 <i>Activity Diagram Menu Data Sungai</i> | 40 |
| Gambar 3. 8 <i>Activity Diagram Menu Tambah Data</i> | 41 |
| Gambar 3. 9 <i>Diagram Class</i> | 42 |
| Gambar 3. 10 Halaman <i>Login</i> | 43 |
| Gambar 3. 11 Halaman <i>Dashboard</i> | 43 |
| Gambar 3. 12 Halaman <i>Monitoring</i> | 44 |
| Gambar 3. 13 Halaman <i>Monitoring Sungai</i> | 44 |
| Gambar 3.14 Halaman <i>sungai</i> | 45 |
| Gambar 3. 15 Halaman <i>Tambah data sungai</i> | 45 |
| Gambar 3. 16 <i>Tampilan Monitoring</i> | 46 |
| Gambar 3. 17 <i>Alat Monitoring</i> | 46 |
| Gambar 3. 18 <i>Access Point</i> | 47 |

BAB I

PENDAHULUAN

1.1. Latar Belakang

Selain gempa dan tsunami, sebagai daerah rawan bencana (*ring of fire*), Indonesia juga sering dilanda bencana alam lainnya, seperti banjir bandang dan tanah longsor. Bencana banjir selain kerap menimbulkan banyak korban jiwa dan harta, juga turut merusak lingkungan. Banyaknya korban pada bencana tersebut menggambarkan kurangnya kesiapan dan antisipasi masyarakat khususnya masyarakat yang tinggal di daerah rawan bencana. Hal ini mencerminkan kurangnya pengetahuan dan minimnya informasi mengenai fenomena bencana alam yang terjadi di daerah tersebut (Hidayati, 2018).

Bencana yang selalu mengancam Kota Kendari hingga saat ini adalah banjir dan tanah longsor. Banjir dan tanah longsor yang terjadi pada 2013 silam misalnya, masih berbekas dalam ingatan warga kota. Banjir merendam 10 kecamatan di Kota Kendari yaitu Kecamatan Poasia, Abeli, Kambu, Baruga, Wua-Wua, Kadia, Mandonga, Puuwatu, Kendari Barat dan Kendari. Menurut Badan Penanggulangan Bencana Daerah (BPBD) Kota Kendari (2014), banjir setinggi 30 cm hingga 3 m akibat meluapnya air sungai telah menimbulkan dampak negatif di Kota Kendari. Banjir dan tanah longsor pada 2013 kembali terulang pada 2017. Sama halnya dengan banjir sebelumnya, bencana ini juga menimbulkan banyak dampak negatif. Dampak negatif dimaksud adalah rusaknya infrastruktur, terganggunya aktifitas warga, terputusnya jalan, dan kerugian materil lain bagi warga metro Kendari (Murbawan et al., 2017).

Banjir merupakan bencana yang kerap terjadi di Indonesia hingga saat ini, kurangnya kesadaran dari masyarakat merupakan salah satu pemicu utama terjadinya banjir. Dampak dari banjir ini sangatlah besar bagi masyarakat terutama penduduk yang tinggal di dataran renda atau di sekitar sungai yang rentan meluap. Ketinggian permukaan air sungai merupakan parameter yang biasanya dijadikan data untuk diteliti

perubahannya sebagai alarm atau peringatan akan bencana banjir. Pengawasan terhadap ketinggian air sungai merupakan salahsatu pengawasan yang perlu diperketat karena dapat menyebabkan kerugian serta bahaya yang cukup besar bagi masyarakat karena berhubungan dengan keselamatan penduduk.

Berdasarkan analisis kejadian banjir di kota Kendari pada tahun 2017 tingkat kesigapan rumah tangga dalam menghadapi bencana banjir dari hasil perhitungan indeks yang telah dilakukan secara umum memiliki 4 parameter diantaranya 2 dari 4 parameter tersebut masih menunjukkan indeks kesigapan yang cukup rendah yang dapat dilihat pada parameter sistem peringatan bencana masih mencapai indeks kesigapan 43,3 dengan tingkat kesigapan kurang siap dan parameter mobilisasi sumber daya yang memiliki indeks kesigapan mencapai 55,8 dengan tingkat kesigapan hampir siap dengan begitu indeks gabungan dari empat parameter mencapai angka 69,43 (Murbawan et al., 2017).

Berdasarkan uraian permasalahan diatas, peneliti akan membuat alat dan sistem yang dapat membantu petugas untuk melakukan *monitoring* ketinggian air (banjir) dengan mengangkat judul tugas akhir “Sistem *monitoring* ketinggian air (banjir) menggunakan *Wireless Sensor Network (WSN)* dan web *GIS*”. Alat yang di gunakan yaitu sensor ultrasonik HC-SR04, rain sensor, buzzer, node MCU, *access point*. Alat ini akan melakukan pengukuran terhadap ketinggian air sungai secara *real time*, dan menampilkannya pada *website* yang dapat diakses sebagai *command center* sebagai peringatan dini bahaya banjir (*Early Warning System*) sehingga dapat melakukan evakuasi dengan segera agar dapat meminimalisir kerugian bagi masyarakat.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan maka rumusan masalah dalam penelitian ini adalah bagaimana membangun sistem *Monitoring* ketinggian air (banjir) menggunakan *Wireless Sensor Network (WSN)* dan web *GIS*

1.3 Batasan Masalah

Adapun hal yang menjadi batasan masalah pada penelitian ini adalah sebagai berikut:

1. Perangkat mikrokontroler harus terkoneksi dengan jaringan internet
2. Sistem *Monitoring* dapat berfungsi pada lokasi rawan banjir yang telah di pasangkan komponen-komponen *Monitoring*.
3. Sistem ini merupakan sistem yang berbasis web.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, tujuan dari penelitian ini adalah membangun sistem *Monitoring* ketinggian air (banjir) menggunakan *Wireless Sensor Network (WSN)* dan web *GIS*.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Dapat menerapkan *Wireless Sensor Network (WSN)* dan web *GIS* ke dalam alat atau sistem *monitoring* ketinggian air (banjir).
2. Dapat membantu pihak badan penanggulangan bencana daerah melakukan mitigasi bencana lebih awal.
3. Dapat memberi peringatan dini kepada masyarakat bila air sungai sudah mulai meluap akibat hujan yang turun secara berkepanjangan.

1.6 Sistematika Penulisan

Adapun sistematika penulisan proposal penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini menguraikan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, sistematika penulisan dan tinjauan pustaka.

BAB II LANDASAN TEORI

Pada bab ini landasan teori, sekilas menjelaskan parameter yang akan digunakan dalam penelitian.

BAB III METODOLOGI PENELITIAN

Pada bab ini berisi tentang metode pengumpulan data yang terdiri dari kajian pustaka dan wawancara. Bab ini juga berisi tentang metode pengembangan sistem RUP, serta waktu dan tempat penelitian.

1.7 Tinjauan Pustaka

(Gunawan et al., 2020) Dalam penelitian ini terdiri dari beberapa objek baik itu *software* maupun *hardware* yakni sensor ultrasonik, NodeMCU ESP 8266, *smartphone* dan *relay* sebagai *hardware* nya. Sedangkan aplikasi blynk sebagai *software* sekaligus tempat *Monitoring* dan kontrol manual bagi *relay*. Perancangan Prototipe Perangkat keras model penelitian menggunakan rancang bangun prototipe yang merupakan *design* dari perancangan prototipe alat pengontrol ketinggian air. Sistem diimplementasikan melalui sebuah perangkat lunak. Aplikasi yang dibangun merupakan aplikasi *cloud* berbasis android.

(Anna et al., 2017) Metode pengembangan sistem yang digunakan dalam penelitian ini mengacu pada model waterfall (Demers, 1997), antara lain:

- 1) *System requirements*,
- 2) *Software requirements*,
- 3) *Analysis*,
- 4) *Program design*,
- 5) *Coding*,
- 6) *Testing*, dan
- 7) *Operations*.

Tujuan utama penelitian ini adalah untuk mengembangkan perangkat lunak SIG berbasis web yang dapat digunakan untuk pemantauan kualitas air sungai. Perangkat

lunak tersebut dikembangkan dengan menggunakan metode model waterfall. Hasil penelitian ini menunjukkan bahwa (a) proses penyusunan prosedur sistem pemantauan pencemaran sungai dilakukan bersama-sama dengan melakukan *Forum Group Discussion* (FGD) agar perangkat lunak mudah digunakan (*user friendly*), dan (b) berdasarkan hasil evaluasi dan uji coba yang dilakukan bersama pemangku kepentingan, penggunaan perangkat lunak SIG berbasis web yang dihasilkan, efektif untuk kegiatan pemantauan kualitas air sungai.

(Ulumuddin et al., 2017) Metode penelitian yang digunakan yaitu Studi Literatur, Pembuatan *Prototype Monitoring* ketinggian air di *project board*. Serta membuat sistem yang mampu menampilkan ketinggian air serta kondisi penGISian air melalui web secara akurat dan *real-time*. Pengujian Aplikasi dilakukan dengan menggunakan bahan uji:

- 1) Ketinggian air dalam tangki air.
- 2) Status penGISian air.

Cara pengujian pada penelitian meliputi: Pengujian ini dilakukan dengan cara mengakses aplikasi pada beberapa *browser* seperti Google Chrome, Mozilla Firefox, Opera Mini, dan Microsoft Edge. Sistem *Monitoring* tangki air berbasis *Internet of Things* dengan menggunakan NodeMCU ESP8266 dan Sensor Ultrasonik. Sistem *monitoring* ini mampu *memonitoring* ketinggian pada tangki air secara akurat dan *real-time*. Pada sistem ini ditambahkan indikator penGISian air menggunakan *rain detector* sensor pada saat *jet pump* diaktifkan, sehingga memudahkan dalam pemantauan air menGISi atau hanya aktifnya *jet pump* tanpa naiknya air dari penampungan ke tangki air.

(Abdulrachman et al., 2017) Metode penelitian dimulai dari studi literatur dengan pengumpulan bahan-bahan referensi dari sumber terpercaya yang berkaitan dengan penelitian ini. Selanjutnya menganalisa kebutuhan sistem mulai dari perangkat keras dan perangkat lunak. Berdasarkan hasil analisa kebutuhan sistem, selanjutnya dilakukan proses perancangan sistem terhadap perangkat keras dan perangkat lunak.

Setelah selesai, maka dilakukan tahap implementasi, tahap pembuatan sistem keseluruhan. Pada penelitian ini proses perancangan sistem *monitoring* dan kontrol jumlah pemakaian air pada kamar kos dengan menggunakan teknologi *Wireless Sensor Network* dilakukan dengan cara menggunakan dua node sensor. Pada masing-masing node sensor terdapat beberapa komponen didalamnya yang bekerja berdasarkan fungsinya, yang akan menghasilkan data yang telah disensing. Data yang dihasilkan oleh kedua node sensor akan dikirim ke *server* melalui jaringan *internet*. Dan pada penelitian ini data-data *monitoring* setiap node sensor telah berhasil ditampilkan pada *website* dengan menggunakan teknologi API (*Application Programming Interface*) yang merupakan teknologi bertukar informasi atau data antara node sensor dan *website*.

BAB II

LANDASAN TEORI

2.1 Sistem *Monitoring*

Menurut Harijono Djojodihardjo, sistem merupakan gabungan obyek yang memiliki hubungan secara fungsi dan hubungan antara setiap ciri objek, secara keseluruhan menjadi satu kesatuan yang berfungsi. Sistem adalah suatu kumpulan objek atau bagian-bagian yang saling memiliki hubungan, dan saling mempengaruhi satu sama lain serta memiliki keterkaitan pada rencana yang sama dalam mencapai suatu tujuan tertentu, sedangkan *monitoring* adalah suatu aktifitas yang bertujuan untuk memantau atau mengamati sesuatu. Sehingga dapat disimpulkan bahwa sistem *Monitoring* adalah gabungan objek yang saling berhubungan dengan tujuan memantau atau mengamati sesuatu.

2.2 Banjir

Banjir adalah kejadian alam dimana suatu daerah atau daratan yang biasanya kering menjadi terendam air. Secara sederhana, banjir dapat didefinisikan sebagai luapan air dalam jumlah besar ke daratan yang biasanya kering. Banjir terjadi karena banyak hal seperti hujan yang berlebihan, meluapnya aliran sungai, sungai, danau atau lautan. Banjir sangat berbahaya dan berpotensi menyapu bersih seluruh kota, garis pantai atau daerah dan menyebabkan kerusakan luas pada kehidupan dan properti.

2.3 Air

Air adalah zat yang paling penting dalam kehidupan setelah udara. Sekitar tiga per empat bagian dari tubuh kita terdiri dari air dan tidak seorangpun dapat bertahan hidup lebih dari 4 – 5 hari tanpa minum air. Selain itu, air juga dipergunakan untuk memasak, mencuci, mandi, dan membersihkan kotoran yang ada di sekitar rumah. Air

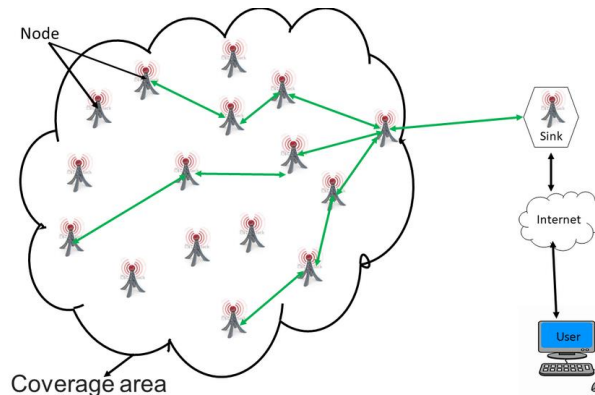
juga digunakan untuk keperluan industri, pertanian, pemadam kebakaran, tempat rekreasi, transportasi, dan lain-lain.

2.4 *Wireless Sensor Network (WSN)*

Wireless Sensor Network (WSN) merupakan jaringan *wireless* alat yang menggunakan sensor untuk memonitor fisik atau kondisi lingkungan sekitar, seperti suhu, suara, getaran, gelombang elektromagnetik, tekanan, gerakan, dan lain-lain. WSN merupakan suatu kesatuan dari proses pengukuran, komputasi, dan komunikasi yang memberikan kemampuan administratif kepada sebuah perangkat, observasi, dan melakukan penanganan terhadap setiap kejadian dan fenomena yang terjadi di lingkungan yang menggunakan teknologi *wireless*. Sistem ini jauh lebih efisien dibandingkan dengan penggunaan kabel.

Komponen WSN meliputi sensor, modul *wireless*, dan PC. Seluruh komponen akan membentuk suatu sistem *Monitoring* yang mampu menampilkan data yang berupa karakteristik sensor yang digunakan dengan memanfaatkan media *wireless* (Hariyawan, M.Y, 2014).

Perkembangan dari WSN sebenarnya sudah dimulai dari kebutuhan dalam bidang militer seperti pemantauan pada saat perang di medan perang. Tetapi sekarang WSN sudah digunakan dalam bidang industri dan penggunaan untuk kemudahan masyarakat sipil, melingkupi pengawasan dan pengontrolan proses dalam industri, mesin pengawasan kesehatan, pemantau kondisi lingkungan, aplikasi untuk kesehatan, otomatisasi pada rumah, dan pengaturan pada lalu lintas. Adapun bentuk arsitektur WSN dapat dilihat Gambar 2.1.



Gambar 2. 1 Arsitektur WSN

2.5 *Geographic Information System (GIS)*

Sistem Informasi Georafis atau *Georaphic Information Sistem (GIS)* merupakan suatu sistem informasi yang berbasis komputer, dirancang untuk bekerja dengan menggunakan data yang memiliki informasi spasial (bereferensi keruangan). Sistem ini *mengcapture*, mengecek, mengintegrasikan, memanipulasi, menganalisa, dan menampilkan data yang secara spasial mereferensikan kepada kondisi bumi.

Pada konfigurasi Web *GIS* ada *server* yang berfungsi sebagai *map server* yang bertugas memproses permintaan peta dari *client* dan kemudian mengirimkannya kembali ke *client*. Dalam hal ini pengguna / *client* tidak perlu mempunyai *software GIS*, hanya menggunakan internet *browser* seperti Internet Explorer, Mozilla Fire Fox, atau Google Chrome untuk mengakses informasi *GIS* yang ada di *server* (Margareth, n.d.).

2.6 **Arduino IDE**

IDE itu merupakan kependekan dari *Integrated Development Enviroenment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software*

inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Bahasa pemrograman Arduino (*Sketch*) sudah di lakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler. Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library* C/C++ yang biasa disebut *Wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah.

1. Arduino IDE ini dikembangkan dari *software Processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino. *Software* IDE terdiri dari tiga bagian *Editor program*, untuk menulis dan mengedit program dalam bahasa *processing*. *Listing* program pada Arduino disebut *sketch*.
2. *Compiler*, modul yang berfungsi mengubah bahasa *processing* (kode program) kedalam kode biner karena kode biner adalah satu-satunya bahasa program yang dipahami oleh mikrokontroller.
3. *Uploader*, modul yang berfungsi memasukkan kode biner kedalam memori mikrokontroller. Struktur perintah pada arduino secara garis besar terdiri dari dua bagian yaitu *void setup* dan *void loop*. *Void setup* berisi perintah yang akan dieksekusi hanya satu kali sejak arduino dihidupkan sedangkan *void loop* berisi perintah yang akan dieksekusi berulang-ulang selama arduino dinyalakan. Adapun tampilan Arduino IDE dapat dilihat pada gambar 2.2.



Gambar 2. 2 Arduino

2.7 Node MCU

NodeMCU merupakan sebuah *open source* platform IoT dan pengembangan kit yang menggunakan bahasa pemrograman Lua untuk membantu *makers* dalam membuat prototype produk IoT atau bisa dengan memakai sketch dengan arduino IDE. Pengembangan Kit ini didasarkan pada modul ESP8266, yang mengintegrasikan *GPIO*, *PWM* (*Pulse Width Modulation*), *IIC*, *1-Wire* dan *ADC* (*Analog to Digital Converter*) semua dalam satu *board*.

Keunikan dari *NodeMCU* sendiri yaitu papannya yang berukuran sangat kecil yaitu panjang 4.83cm, lebar 2.54cm, dan berat 7 gram. Walaupun ukurannya yang kecil, *NodeMCU* sudah dilengkapi dengan fitur wifi dan *firmware* yang bersifat *opensource*. Adapun gamabar dari *NodeMCU* dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Node MCU

2.8 Sensor Ultrasonik HC-SR04

HC-SR04 merupakan Sensor Ultrasonik yang dapat digunakan untuk mengukur jarak antara penghalang dan sensor. Sensor ini mirip dengan sensor PING namun berbeda dalam jumlah pin serta spesifikasinya. Sensor jenis ini adalah modul elektronik yang mendeteksi sebuah objek menggunakan suara. Sensor Ultrasonik HC-SR04 terdiri dari sebuah *transmitter* (Pemancar) dan sebuah *Receiver* (penerima). *Transmitter* berfungsi untuk memancarkan sebuah gelombang suara kearah depan. Jika ada sebuah objek di depan *transmitter* maka sinyal tersebut akan memantul kembali ke *Receiver*. Kegunaan sensor ultrasonik HC-SR04 pada alat ini yaitu sebagai pendeteksi utama ketinggian air. Adapun bentuk dari sensor ultrasonik HC-SR04 dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Sensor Ultrasonik HC-SR04

2.9 Rain sensor

Rain sensor merupakan sensor yang berfungsi untuk mendeteksi hujan turun atau tidak. Intinya sensor ini jika terkena air pada papan sensornya maka resistansinya akan berubah, semakin banyak semakin kecil dan sebaliknya. Untuk pengaplikasiannya sensor ini dapat digunakan untuk jemuran otomatis jadi ketika hujan turun sensor mendeteksi dan akan memberikan peringatan atau untuk tambahan dapat digunakan penutup yang dapat melindungi jemuran pada saat hujan. Kegunaan *Rain*

sensor pada alat ini adalah sebagai pendeteksi hujan. Adapun bentuknya dapat dilihat pada gambar 2.5.



Gambar 2. 5 Rain Sensor

2.10 Buzzer

Buzzer Listrik adalah sebuah komponen elektronika yang dapat mengubah sinyal listrik menjadi getaran suara. Pada umumnya, *Buzzer* yang merupakan sebuah perangkat *audio* ini sering digunakan pada rangkaian anti-maling, Alarm pada Jam Tangan, Bel Rumah, peringatan mundur pada Truk dan perangkat peringatan bahaya lainnya. Kegunaan *Buzzer* pada alat ini adalah sebagai pemberi alarm utama bila ketinggian air telah melebihi batas maksimal. Adapun bentuk dari *Buzzer* dapat dilihat pada gambar 2.6.



Gambar 2. 6 Buzzer

2.11 Website

Website adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol HTTP (*hyper text transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser. Beberapa jenis browser yang populer saat ini diantaranya: Internet Explorer yang diproduksi oleh Microsoft, Mozilla Firefox, Opera dan Safari yang diproduksi oleh Apple. Browser (peramban) adalah aplikasi yang mampu menjalankan dokumen-dokumen web dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat di dalam aplikasi browser yang biasa disebut *web engine*. Semua dokumen web ditampilkan dengan cara diterjemahkan (Suhartanto, 2016).

2.12 Xampp

XAMPP adalah sebuah paket perangkat lunak (*software*) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris. Program aplikasi XAMPP berfungsi sebagai *server* lokal untuk mengampu berbagai jenis data *website* yang sedang dalam proses pengembangan.

Xampp adalah *software web server apache* yang di dalamnya tertanam *server* MySQL yang didukung dengan bahasa pemrograman PHP untuk membuat *website* yang dinamis. XAMPP sendiri mendukung dua *system* operasi yaitu windows dan Linux. Untuk linux dalam proses penginstalannya menggunakan *command line* sedangkan untuk windows dalam proses penginstalannya menggunakan *interface* grafis sehingga lebih mudah dalam penggunaan XAMPP di Windows dibanding dengan Linux (Ardianto & Nst, 2016) .

2.13 PHP

PHP adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server* tidak dijalankan pada *client*. PHP merupakan suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *Open Source*, yaitu pengguna dapat mengembangkan kode fungsi PHP dengan kebutuhannya.

PHP sendiri berasal dari kata *Hypertext Preprocessor*, sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (situs personal), PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web, pada tahun 1997 sebuah perusahaan bernama Zend menulis ulang *interpreter* PHP menjadi lebih baik dan merilis PHP 2.0 kemudian pada tahun 1998 Zend merilis PHP 3.0, pada pertengahan tahun 1999 Zend merilis *interpreter* PHP baru yang dikenal dengan PHP 4.0, versi ini adalah versi PHP yang paling banyak dipakai pada abad ke-21, beberapa tahun kemudian tepatnya pada juni 2004, Zend merilis PHP 5.0 dimana versi ini memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Versi terbaru dari bahasa pemrograman PHP saat ini adalah versi 7.1.2.

Bahasa pemrograman ini menggunakan *server-side programming* yang artinya program tersebut akan dijalankan atau diproses oleh *server*. Kelebihan PHP adalah bahasa *open source* yang dapat digunakan di Linux, Unix, Macintosh, Windows dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem (Hidayat et al., 2017).

2.14 *Casding Style Sheet (CSS)*

Cascading Style Sheets (CSS) adalah salah satu bahasa pemrograman desain web (*style sheet language*) yang mengontrol format tampilan sebuah halaman web yang ditulis dengan menggunakan bahasa penanda (*markup language*). Biasanya CSS digunakan untuk mendesain sebuah malam HTML dan XHTML, tetapi sekarang bahasa pemrograman CSS bisa diaplikasikan untuk segala dokumen XML, termasuk SVG dan XUL. CSS dibuat untuk memisahkan kontek utama (biasanya dibuat dengan menggunakan bahasa HTML dan sejenisnya) dengan tampilan dokumen yang meliputi layout, warna dan font. Pemisahan ini dapat meningkatkan daya akses konten pada web, menyediakan lebih banyak fleksibilitas dan control dalam spesifikasi dari sebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi banyak halaman untuk sebuah *forming* dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik *tables* pada *layout* desain web (*layout* tanpa tabel) *Style Sheet* adalah sebuah *text file* yang sederhana (dimana berekstensi *.css), ditulis menurut aturan bahasa yang dipaparkan pada rekomendasi CSS1 atau CSS2. Cara kerja CSS dengan menggunakan dua buah elemen penting untuk pemformatan tampilannya, diantaranya selektor dan deklarator. Dua buah elemen ini berfungsi sebagai penentu format tampilan dan lainnya menempatkan format tampilan tersebut. Deklarator berisi beberapa perintah-perintah CSS untuk menentukan format dari sebuah elemen pada halaman web. Sedangkan selektor adalah sebuah perintah lanjut dari deklarator dan berfungsi menempatkan format tampilan dari deklarator. Dalam *Cascading Style Sheets* ada dua cara menghubungkan sebuah dokumen HTML dengan CSS.

1. *Selector class.*
2. *Selector ID.*
3. *Selector Descendant (turunan).*
4. *Selector Descendant (turunan).*
5. *Selector Pseudo elemen.*
6. *Selector dynamic pseudo class.*

7. *Selector language.*
8. *Selector child.*
9. *Selector first-child.*
10. *Selector adjacent* (berdekatan).

2.15 Hypertext Markup Language (HTML)

HTML adalah sebuah bahasa pemrograman yang berbentuk skrip-skrip yang berguna untuk membuat sebuah halaman web. HTML dapat dibaca oleh berbagai platform seperti: Windows, Linux, Macintosh. Kata Markup Language pada HTML menunjukkan fasilitas yang berupa tanda tertentu dalam skrip HTML dimana pengguna bisa mengatur judul, garis, tabel, gambar, dan lain-lain dengan perintah yang telah ditentukan pada elemen HTML. HTML sendiri dikeluarkan oleh W3C (*Word Wide Web Consortium*), setiap terjadi perkembangan level HTML harus dievaluasi ketat dan disetujui oleh W3C. Contoh tag HTML antara lain: `<head>`, `<body>` dan `<table>`.

2.16 Codeigniter

Codeigniter merupakan aplikasi sumber terbuka yang berupa kerangka kerja PHP dengan model MVC untuk membangun situs web dinamis dengan menggunakan PHP. *Codeigniter* memudahkan pengembang web untuk membuat aplikasi web dengan cepat dan mudah. *Codeigniter* adalah sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun aplikasi web. Tujuan utama pengembangan *codeigniter* adalah untuk membantu developer dalam pengerjaan aplikasi yang lebih cepat daripada menulis semua *code* dari awal dan *codeigniter* merupakan salah satu *framework php* tercepat yang ada saat ini (Novianto, 2016).

2.17 Firebase

Firebase adalah penyedia layanan *realtime database* dan *backend* sebagai layanan. Suatu aplikasi yang memungkinkan pengembang membuat API untuk

disinkronisasikan untuk *client* yang berbeda-beda dan disimpan pada *cloud*-nya *Firebase*. *Firebase* memiliki banyak *library* yang memungkinkan untuk mengintegrasikan layanan ini dengan Android, Ios, Javacript, Java, Objective-C dan Node.JS. *Database Firebase* juga bersifat bisa diakses lewat REST API. REST API tersebut menggunakan protokol *Server-Sent Event* dengan membuat koneksi HTTP untuk menerima *push notification* dari *server*. Pengembang menggunakan REST API untuk *post* data yang selanjutnya *Firebase client library* yang sudah diterapkan pada aplikasi yang dibangun yang akan mengambil data secara *realtime*.

Produk utama dari *Firebase* yaitu suatu *database* yang menyediakan API untuk memungkinkan pengembang menyimpan dan mensinkronisasi data lewat *multiple client*. Bagi *developer web* yang membangun aplikasi dengan *HTML, CSS dan JS*, Selain dari sisi *server* dan *database*, *Firebase* juga menyediakan *hosting* untuk *static file* yang dilengkapi dengan fasilitas *CDN*. Beberapa contoh aplikasi yang *real time* saat ini seperti *bbm, whatsapp, facebook* dan lain-lain dapat diaplikasikan menggunakan *Firebase* untuk *database* maupun *tools* lainnya yang ada pada *Firebase*. Dalam pembuatan suatu aplikasi yang memerlukan notifikasi secara *realtime* maka *Firebase* dapat membantu dalam mengatasi masalah tersebut melalui *tools* pada *Firebase* dimana *Firebase* menyediakan *library* untuk berbagai *client platform*. Untuk *browser* menggunakan Javascript dan untuk *mobile* menggunakan Objective-C atau Android API.

Firebase memiliki banyak *library* yang memungkinkan untuk mengintegrasikan layanan ini dengan *Android, iOS, Javascript, Java, Objective-C dan Node.JS*. *Database Firebase* juga bisa diakses lewat REST API dan data *binding* untuk beberapa *framework* Javascript seperti halnya *AngularJS, ReactJS, Ember.JS, dan Backbone.JS*. REST API tersebut menggunakan protokol *Server-Sent Event* dengan membuat koneksi HTTP untuk menerima *push notification* dari *server*. Pengembang juga bisa menggunakan *database* ini untuk mengamankan data menggunakan *server Firebase* dengan *rules* yang ada. *Firebase* mempunyai suatu cara untuk mengirim

pesan dengan menggunakan *push notification* sebagai solusinya. Salah satu cara untuk bisa menggunakan *push notification* yaitu dengan *Firebase Cloud Messaging* (Sonita & Fardianitama, 2018).

2.18 *Rational Unified process (RUP)*

Rational Unified process (RUP) adalah suatu kerangka kerja proses pengembangan perangkat lunak interaktif yang dibuat oleh *Rational Software*. *Rational Software* diakuisisi oleh IBM pada februari 2003. Produk ini memuat pengetahuan yang bertautan dengan artefak sederhana disertai deskripsi detail dari beragam aktivitas (Mubarok et al., 2015).

RUP dimasukkan dalam produk IBM *Rational Method Composer* (RMC) yang memungkinkan untuk kustomisasi proses, RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML).

Dalam RUP ada beberapa fase yaitu sebagai berikut:

1. *Inception* (insepsi)

Pada tahap ini pengembang mendefinisikan batasan kegiatan, melakukan analisis kebutuhan *user*, dan melakukan perancangan awal perangkat lunak (perancangan arsitektural dan *use case*).

2. *Elaboration* (elaborasi)

Pada tahap ini dilakukan perancangan perangkat lunak mulai dari analisis kebutuhan, design hingga *coding*.

3. *Construction* (konstruksi)

Pada tahap ini dilakukan pengimplementasian dari rancangan perangkat lunak.

4. *Transition* (transisi)

Pada tahap ini dilakukan instalasi dan sosialisasi perangkat lunak yang telah dibuat.

2.19 *Black Box testing*

Pengujian *Black Box* adalah pengujian perangkat lunak yang fokus pada spesifikasi fungsional dari perangkat tersebut, serta mengevaluasi berdasarkan tampilan dari aplikasi tersebut tanpa mengetahui proses detail yang terjadi pada pengujian aplikasi tersebut pengujian aplikasi yang dilakukan menggunakan algoritma ini akan memungkinkan pengembang *software* membuat kondisi input yang akan melatih syarat-syarat fungsional suatu program. Pada pengujian ini peneliti menggunakan pengujian *Black Box* (Mustaqbal et al., 2015).

2.20 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem. *Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (PBO) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem PBO dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut, UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar PBO sejak tahun 1980-an.

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh *Grady Booch*, *Jim Rumbaugh*, dan *Ivar Jacobson*, namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat.

UML terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu, yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model.

Adapun kegunaan dari UML adalah sebagai berikut:

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan memberikan informasi secara detail tentang *coding* program dan menginterpretasikan kembali ke dalam bentuk diagram (*reverse engineering*).









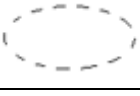

2.20.1 Usecase Diagram

Use case diagram adalah kegiatan atau urutan interaksi yang saling berkaitan antara sistem dan *actor*, *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

Use case juga digunakan untuk membentuk perilaku (*behaviour*) sistem yang akan dibuat, *use case* menggambarkan sebuah interaksi antara pengguna (*actor*) dengan sistem yang ada dan menggambarkan sekelompok *use case* dan *actor* yang disertai dengan hubungan diantaranya. Diagram *use case* ini menjelaskan dan menerangkan kebutuhan yang diinginkan pengguna, serta sangat berguna dalam menentukan struktur organisasi dan model dari pada sebuah sistem. Proses desain mengubah kebutuhan-kebutuhan menjadi bentuk karakteristik yang dimengerti

perangkat lunak sebelum dimulai penulisan program. Adapun komponen-komponen *use case diagram* dapat dilihat pada tabel 2.1.

Tabel 2. 1 Komponen-komponen *use case diagram*






| Simbol | Nama | Keterangan |
|---|-----------------------|--|
|  | <i>Actor</i> | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> . |
|  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>dependent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>). |
|  | <i>Generalization</i> | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk (<i>ancestor</i>). |
|  | <i>Include</i> | Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit. |
|  | <i>Extend</i> | Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan. |
|  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
|  | <i>System</i> | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
|  | <i>Use case</i> | Deskripsi dari uraian aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> . |
|  | <i>Collaboration</i> | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
|  | <i>Note</i> | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

2.20.2 Activity Diagram

Activity diagram ini menggambarkan tentang aktifitas yang terjadi pada sistem. Dari mulai sampai selesai, diagram ini menunjukkan langkah-langkah dalam proses

kerja sistem yang dibuat. Adapun komponen-komponen yang terdapat pada diagram ini dapat dilihat pada tabel 2.2.

Tabel 2. 2 Komponen-komponen *activity diagram*






| Simbol | Nama | Keterangan |
|---|----------------------------|--|
|  | <i>Activity</i> | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain. |
|  | <i>Action</i> | <i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi. |
|  | <i>Initial Node</i> | Bagaimana objek dibentuk atau diawali (<i>start flow</i>). |
|  | <i>Activity Final Node</i> | Bagaimana objek dibentuk dan diakhiri (<i>end flow</i>). |
|  | <i>Fork Node</i> | Satu aliran yang pada tahap tertentu berubah menjadi beberapa alur. |

Activity diagram merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar transisi dipanggil oleh *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

2.20.3 *Class Diagram*

Class diagram adalah model statis yang menggambarkan struktur dan deskripsi *class* serta hubungan antar *class*. *Class diagram* mirip ER-Diagram pada perancangan *database*, bedanya pada ER-diagram tidak terdapat operasi atau metode tapi hanya atribut. *Class* terdiri dari nama kelas, atribut dan operasi atau metode. Adapun komponen-komponen yang terdapat pada diagram ini dapat dilihat pada tabel 2.3.

Tabel 2. 3 Komponen-komponen *class diagram*

| Simbol | Nama | Keterangan |
|---|-----------------------------|--|
|  | <i>Association</i> | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
|  | <i>Directed Association</i> | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
|  | <i>Generalization</i> | Relasi antar kelas dengan makna generalisasi-spesialisasi. |
|  | <i>Dependency</i> | Relasi antar kelas dengan makna kebergantungan antar kelas. |
|  | <i>Aggregation</i> | Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>). |

Relasi antar kelas biasanya diikuti dengan *Multiplicity* atau multiplisitas, multiplisitas menunjukkan jumlah banyaknya obyek sebuah *class* yang berelasi dengan sebuah obyek lain pada *class* lain yang berasosiasi dengan *class* tersebut. Adapun komponen-komponen *multiplicity* dapat dilihat pada tabel 2.4.

Tabel 2. 4 komponen-komponen *multiplicity*


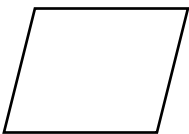
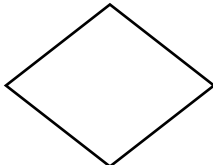

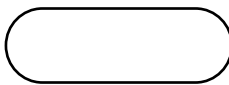


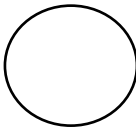
| Nilai Kardinalitas | Keterangan |
|--------------------|---------------------------------|
| 0..1 | Nol atau satu |
| 1 | Hanya satu |
| 0..* | Nol atau lebih |
| 1..* | Satu atau lebih |
| N | Hanya n (dengan $n > 1$) |
| 0..n | Nol sampai n (dengan $n > 1$) |
| 1..n | Satu sampai n (dengan $n > 1$) |

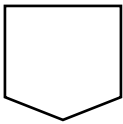

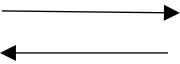
2.21 *Flowchart*

Flowchart merupakan penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara

logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Adapun simbol – simbol flowchart dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Simbol - Simbol *Flowchart*

| No | Gambar | Nama | Keterangan |
|----|---|------------------------|--|
| 1. |  | Proses | Mempresentasikan operasi. |
| 2. |  | <i>Input/ Output</i> | Mempresentasikan Input data atau Output data yang diproses atau informasi. |
| 3. |  | Keputusan | Keputusan dalam program. |
| 4. |  | Dokumen | Dokumen I / O dalam format cetak |
| 5. |  | <i>Terminal points</i> | Awal / akhir <i>flowchart</i> |
| 6. |  | <i>Preparation</i> | Pemberian harga awal |
| 7. |  | <i>Manual input</i> | Input yang dimasukkan secara manual dari keyboard |
| 8. |  | Penghubung | Keluar atau masuk dari bagian lain <i>flowchart</i> khususnya |

| | | | |
|-----|---|------------|--|
| 9. |  | Penghubung | Keluar atau masuknya dari bagian lain <i>flowchart</i> khususnya halaman lain. |
| 10. |  | Display | Output yang ditampilkan pada terminal |
| 11. |  | Anak panah | Mempresentasikan alur kerja |

BAB III

METODOLOGI PENELITIAN

3.1 Pengumpulan Data

Adapun metode pengumpulan data yang digunakan dalam penelitian ini yaitu:

1. Kajian Pustaka

Pada metode pengumpulan data yang dilakukan yaitu dengan cara melakukan pengumpulan data pendukung penelitian yang akan dijadikan referensi, data dapat berupa buku, paper, jurnal, skripsi dan sebagainya, yang berkaitan dengan sistem *monitoring* yang akan dibangun.

2. Wawancara (*interview*)

Pengumpulan data dengan wawancara ini dilakukan untuk mengumpulkan informasi-informasi yang dibutuhkan dalam membangun sistem *monitoring* seperti data sungai yang akan *dimonitoring* yang bertempat di badan penanggulangan bencana daerah Kota Kendari.

3.2 Metode Pengembangan Perangkat Lunak

Dalam merancang atau membuat perangkat lunak dibutuhkan suatu pemodelan dari keseluruhan proses-proses yang akan dilakukan selama pembuatan perangkat lunak tersebut. Model yang digunakan dalam perangkat lunak ini adalah proses *Rational Unified Proses* (RUP). Adapun proses dari metode/model RUP ini meliputi:

1. *Inception* (Pemulaan)

Pada fase ini dilakukan proses pengidentifikasian sistem, dilakukan dengan menganalisis kebutuhan sistem, melakukan kajian terhadap penelitian yang terkait dengan sistem *monitoring* menggunakan WSN dan Web *GIS*.

2. *Elaboration* (Perluasan/perencanaan)

Setelah menentukan ruang lingkup penelitian, selanjutnya pada fase ini akan dilakukan perancangan dan analisis sistem menggunakan *Unified Modeling Language* (UML) yang meliputi *use case*, *Activity Diagram* dan *class diagram*.

3. *Construction* (Konstruksi)

Proses yang dilakukan pada fase ini adalah mengimplementasikan kode sesuai perancangan yang telah dilakukan sebelumnya sehingga menjadi sistem yang dapat digunakan. Bahasa pemrograman yang akan digunakan dalam penelitian ini adalah HTML, PHP, bahasa C dan *framework* yang di gunakan yaitu *Codeigniter*. Adapun database yang akan peneliti gunakan yaitu *Firebase* dan *MYSQL*.

4. *Transition* (Transisi)

Pada fase ini akan dilakukan pengujian dan perpindahan pada sistem yang telah dibangun. Adapun bentuk pengujian yang akan dilakukan dengan menampilkan data perbandingan keakuratan sensor ultrasonik HC-SR04 dan *rain* sensor serta pengujian *blackbox*.

3.3 Waktu dan Tempat Penelitian

3.3.1 Waktu Penelitian

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Mei 2021 sampai dengan Agustus 2021, dengan rincian kegiatan dapat dilihat pada tabel 3.1.

Tabel 3. 1 Gannt Chart waktu penelitian

| No | Uraian | Waktu (2021) | | | | | | | | | | | | | | | |
|----|---------------------|--------------|---|---|---|------|---|---|---|---------|---|---|---|-----------|---|---|---|
| | | Juni | | | | Juli | | | | Agustus | | | | September | | | |
| | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | <i>Inception</i> | | | | | | | | | | | | | | | | |
| 2 | <i>Elaboration</i> | | | | | | | | | | | | | | | | |
| 3 | <i>Construction</i> | | | | | | | | | | | | | | | | |
| 4 | <i>Transition</i> | | | | | | | | | | | | | | | | |

3.3.2 Tempat Penelitian

Adapun tempat penelitian tugas akhir yang akan dilakukan mengacu pada tempat *Laboratorium Software Engineering* Universitas Halu Oleo.

3.4 Analisis Sistem *Monitoring* yang akan dibangun

Analisis sistem *monitoring* yang akan dibangun merupakan tahapan yang bertujuan untuk mengetahui apa saja yang dibutuhkan dalam sistem yang akan dibangun.

Dalam analisis sistem ini dilakukan analisis kebutuhan fungsional dan analisis kebutuhan non fungsional.

3.4.1 Kebutuhan Fungsional

Analisis kebutuhan fungsional data yang dibutuhkan agar sistem dapat berjalan sesuai dengan prosedur. Setelah melalui tahapan analisis kemudian akan ditetapkan kebutuhan-kebutuhan untuk membangun sistem. Pada analisis kebutuhan fungsional terdapat beberapa kebutuhan-kebutuhan untuk membangun sistem yaitu analisis kebutuhan *input*, analisis kebutuhan proses, dan analisis kebutuhan *output*.

1. Perancangan diagram sistem menggunakan pemodelan *Unified Modeling Language* (UML) seperti *usecase diagram*, *Activity diagram* dan *class diagram*.
2. Menganalisis kebutuhan data pada sistem melalui penginputan data seperti data sungai yang akan diteliti.

3.4.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional bertujuan untuk menganalisis sumber daya yang dibutuhkan untuk membangun sistem. Analisis kebutuhan nonfungsional dibagi menjadi dua tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak yang bertujuan untuk merincikan komponen-komponen yang dibutuhkan dalam membangun sistem. Perangkat keras yang dibutuhkan untuk membangun sistem *Monitoring* ketinggian air (banjir) dapat dilihat pada tabel 3.2.

Tabel 3. 2 Spesifikasi Perangkat Keras

| No. | Nama Perangkat | Jumlah | Jenis Komponen | Deskripsi |
|-----|---------------------------|--------|----------------|---|
| 1. | NodeMCU ESP 8266 | 1 | Proses | Digunakan sebagai pengontrol sistem dari semua komponen dan alat penyimpanan perangkat lunak sehingga instruksi - instruksi pada perangkat lunak dapat berjalan serta untuk komunikasi data |
| 2. | Sensor Ultrasonik HC-SR04 | 1 | <i>Input</i> | HC-SR04 merupakan Sensor Ultrasonik yang dapat digunakan untuk mengukur jarak antara penghalang dan sensor. |
| 3. | <i>Rain sensor</i> | 1 | <i>Input</i> | <i>Rain sensor</i> merupakan sensor yang berfungsi untuk mendeteksi hujan turun atau tidak. |
| 4. | <i>Buzzer</i> | 1 | <i>Output</i> | <i>Buzzer</i> Listrik adalah sebuah komponen elektronika yang dapat mengubah sinyal listrik menjadi getaran suara. |

Adapun spesifikasi perangkat lunak yang dibutuhkan untuk membangun sistem *monitoring* dapat dilihat pada tabel 3.3.

Tabel 3. 3 Spesifikasi Perangkat Lunak

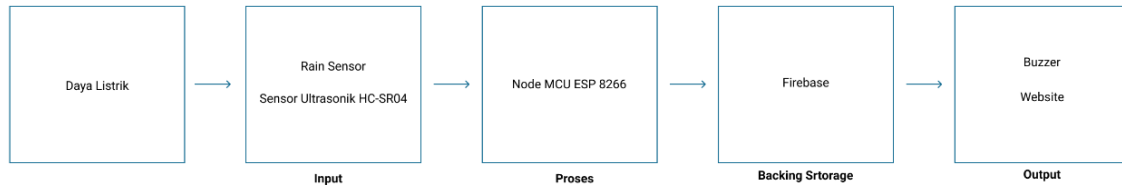
| No. | Nama Perangkat | Deskripsi |
|-----|---------------------------|---|
| 1. | Sistem operasi windows 10 | Digunakan untuk menjalankan aplikasi-aplikasi yang dibutuhkan dalam pembuatan sistem |
| 2. | Visual Studio Code | Digunakan untuk <i>teks editor</i> dalam pembuatan sistem ini |
| 3. | Xampp | Program aplikasi XAMPP berfungsi sebagai <i>server</i> lokal untuk mengampu berbagai jenis data <i>website</i> yang sedang dalam proses pengembangan. |
| 4. | <i>Firebase</i> | Digunakan <i>developer</i> untuk menyimpan data dan synchronize ke banyak <i>user</i> . |
| 5. | Arduino IDE | Sebagai <i>text editor</i> untuk membuat, mengedit dan juga memvalidasi kode program |

3.5 Arsitektur Sistem

Arsitektur sistem digunakan untuk mendefinisikan komponen – komponen *Hardware* dan *software* yang lebih spesifik secara terstruktur. *Hardware* merupakan kumpulan elemen atau komponen fisik yang menyusun sistem, sedangkan *software* merupakan sebuah data yang diprogram dan disimpan secara digital yang tidak terlihat secara fisik.

3.5.1 Hardware

Pada *Hardware* arsitektur sistem yang diilustrasikan pada Gambar 3.1 menunjukkan komponen – komponen *Hardware* secara spesifik dan terstruktur.



Gambar 3. 1 Ilustrasi *Hardware* Arsitektur Sistem

3.5.2 Input Device

Input Device pada sistem yang dirancang antara lain yaitu *Rain* sensor dimana bertugas untuk mendeteksi bila terjadi hujan. dan Sensor Ultrasonik HC-SR04 berfungsi untuk mendeteksi jarak ketinggian air. Setelah sensor - sensor tersebut membaca data yang dihasilkan, sensor kemudian mengirimkan data tersebut ke *process device*. Yang nantinya data yang dikirim akan diolah untuk kemudian dijadikan informasi yang diperlukan.

3.5.2 Process Device

Process device pada sistem yang dirancang adalah NodeMCU yang bertugas untuk mengolah data yang telah dikirim sensor. Data yang diolah kemudian dikirim ke *Output system* agar data yang telah diolah dapat di tampilkan.

3.5.3 Backing Storage

Backing storage pada sistem yang dirancang adalah firebase, firebase digunakan untuk menyimpan data yang dikirim oleh process device secara realtime, data yang diperbaharui secara real-time tersebut digunakan user untuk melakukan *Monitoring*

3.5.4 Output System

Output System pada sistem yang dirancang adalah *website website* digunakan untuk menampilkan data yang dikirim oleh *process device*, data yang yang di

tampilkan secara *real-time* tersebut akan digunakan *user* untuk melakukan *Monitoring* ketinggian air dengan menampilkan lokasi bencana banjir.

3.5.4 Software

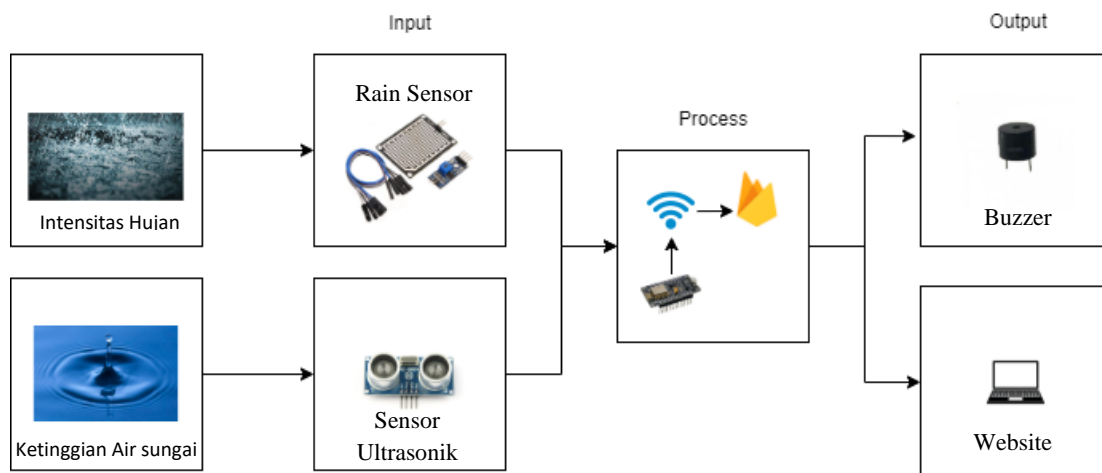
Pada *software* arsitektur sistem yang diilustrasikan pada Gambar 3.2 menunjukkan komponen – komponen *software* secara spesifik dan terstruktur.

3.5.5 Software Sistem

Software sistem pada sistem yang dirancang menggunakan *website* untuk menampilkan data yang diolah oleh *Hardware*. Data yang diolah akan ditampilkan pada *dashboard* sistem yang dibuat sehingga *user* dapat dengan leluasa untuk melakukan *Monitoring* ketinggian air.

3.6 Blok Diagram

Pada perancangan sistem ini disediakan blok diagram untuk menjadi ilustrasi alur kerja alat. Adapun komponen – komponen utama dari alat ini dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Blok Diagram

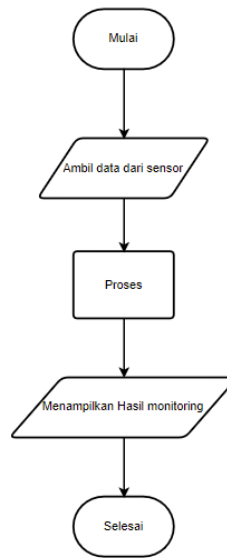
Cara kerja sistem *monitoring* ketinggian air yaitu ketika mikrokontroler dialiri arus listrik maka Sensor Ultrasonik HC-SR04 dan *Rain* sensor akan aktif lalu *Rain* sensor akan bekerja ketika hujan terkena sensor. Selanjutnya sensor akan mengirim data ke mikrokontroler dan dari mikrokontroler akan mengirim data ke *Firebase* lalu diteruskan ke dalam sistem bahwa di sungai A sedang diguyur hujan, dan Sensor Ultrasonik HC-SR04 akan bekerja bila ketinggian air sungai akan semakin meningkat dengan membaca jarak yang ada pada pelampung yang telah dipasang pada alat ini, maka sensor akan mengirim data ke mikrokontroler untuk dikirim ke *Firebase* dan diteruskan ke *website*. Di dalam *website* terdapat menu *monitoring* dimana pada menu *monitoring* ada tiga komponen utama yaitu *water level* yang berfungsi sebagai pengukur *realtime* ketinggian air sungai, *rain level* yang akan menampilkan data curah hujan yang terjadi di lokasi alat *monitoring*, dan Web *GIS* yang berfungsi untuk menampilkan titik lokasi kejadian banjir, sehingga dengan adanya sistem ini dapat membantu pihak badan penanggulangan bencana daerah untuk melakukan mitigasi bencana lebih awal. Dan alat selanjutnya adalah buzzer akan bekerja sebagai output pada alat apabila ketika ketinggian air telah mencapai level tertentu maka *buzzer* akan mengeluarkan suara sebagai tanda alarm untuk penduduk setempat agar segera waspada akan banjir. Alat ini didesain untuk bekerja di sungai yang rawan meluap akibat tingginya curah hujan.

3.7 Perancangan Sistem

Perancangan sistem yang akan dibangun meliputi, perancangan *use case diagram*, skenario sistem, perancangan *Activity Diagram*, perancangan *class diagram*, perancangan *sequence diagram*, perancangan *interface* aplikasi berbasis *website*, dan perancangan komponen alat *Monitoring*.

3.7.1 Flowchart

Flowchart digunakan untuk mengelola alur kerja karena *flowchart* berfungsi dalam penetapan karakter, yakni dapat membentuk hasil yang bermutu berdasarkan strategi. Adapun contoh *flowchart* untuk sistem ini dapat dilihat pada Gambar 3.3.

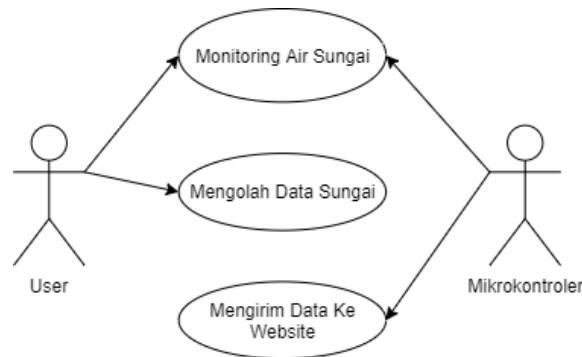


Gambar 3. 3 Flowchart

Gambar 3.3 menjelaskan alur kerja sistem mulai dari mengambil data yang dilakukan oleh kedua sensor yaitu sensor ultrasonik HC-SR04 dan rain sensor, kemudian akan di proses lalu hasil dari pengambilan sensor akan ditampilkan kedalam sistem dengan memperlihatkan *water level*, *Rain level*, dan Web *GIS* sebagai tiga komponen utama pada sistem ini.

3.7.2 Perancangan *use case diagram*

Use case diagram digunakan untuk memodelkan fungsionalitas fungsionalitas sistem yang dilihat dari pengguna yang ada diluar sistem (aktor). Adapun bentuk *use case diagram* dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Use case Diagram

Gambar 3.3 merupakan tampilan *use case diagram* dimana dijelaskan bahwa *user* akan melakukan *Monitoring* air sungai dan mengolah data sungai dan mikrokontroler akan melakukan *Monitoring* air sungai menengirim data dari hasil *Monitoring* kedalam *website*.

3.7.3 Skenario Sistem

- Skenario Sistem *Monitoring*

Use Case : Sistem

Aktor : *User*, Sensor Ultrasonik HC-SR04, *Rain* sensor, Buzzer, sistem

Deskripsi : *User* membuka sistem *Monitoring*, sistem secara otomatis akan memeriksa apakah semua alat telah terkoneksi dengan internet lalu sistem akan menampilkan *water level* dan lokasi dari berbagai titik sungai yang telah dipasang alat *Monitoring* ke dalam *dashboard*. Ketika *rain sensor* mendeteksi terjadinya hujan maka alat akan mengirim notifikasi bahwa sungai A sedang terjadi hujan lalu Sensor Ultrasonik HC-SR04 akan mendeteksi ketinggian air dengan menampilkan data hasil *Monitoring* kedalam *website* yang berupa *water level*, *rain level* dan *map* yang nantinya akan dikelola sebagai *Comand center* serta *buzzer* akan berfungsi sebagai alarm untuk masyarakat yang tinggal di sekitar lokasi rawan banjir. Adapun skenario sistem *monitoring* dapat dilihat pada tabel 3.4.

Tabel 3. 4 Skenario Sistem *Monitoring*

| Tindakan Aktor | Reaksi Sistem |
|--|---|
| 1. <i>User</i> membuka aplikasi | |
| | 2. Menampilkan halaman <i>Login</i> |
| 3. <i>User</i> memasukan <i>username</i> dan <i>password</i> | |
| | 4. Jika benar akan menampilkan halaman beranda. Jika salah akan kembali ke halaman <i>Login</i> |
| | 5. Mengecek status sensor |
| | 6. Mengirim data ke <i>Firebase</i> |
| | 7. Mengambil data dari <i>Firebase</i> |
| | 8. menampilkan data <i>Monitoring</i> pada sistem |
| 9. <i>User</i> melakukan <i>Monitoring</i> | |
| 10. <i>User</i> menutup aplikasi | |

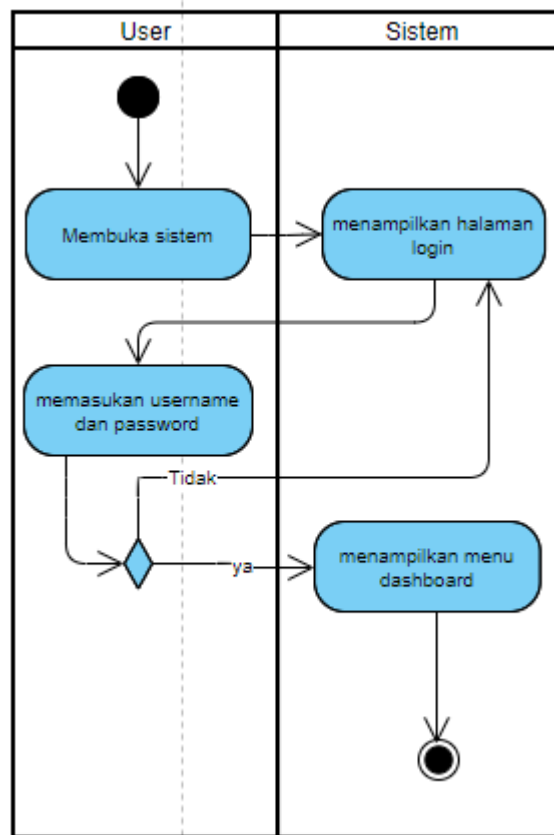
3.7.4 Perancangan *Activity Diagram*

1. *Activity Diagram*

Activity Diagram adalah *activity* yang menggambarkan alir sebuah sistem yang sedang dibuat, bagai mana alir sistem berawal. Ada beberapa *activity* yang menggambarkan alur aktivitas sistem.

a. *Activity Diagram Login*

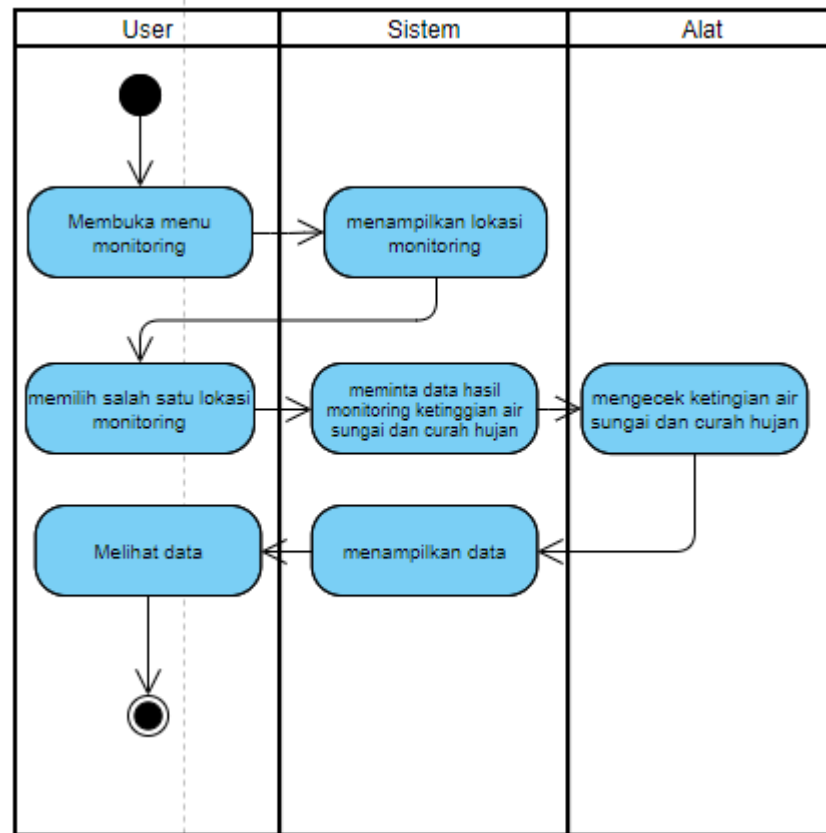
Pada tampilan *activity user* memasukan *Username* dan *Password*, kemudian sistem akan melakukan validasi. Adapun *activity login* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Activity Diagram Menu Login

b. *Activity Diagram Menu Monitoring*

Pada *Activity Diagram* menu *Monitoring*, *user* memilih menu tersebut sehingga sistem akan menampilkan beberapa lokasi daerah yang sedang dimonitoring ketika *user* memilih salah satu lokasi yang dipilih maka sistem akan menampilkan *water level*, *rain level* dan *map* ketinggian air sungai. Adapun *Activity diagram* menu *monitoring* dapat dilihat pada Gambar 3.6.



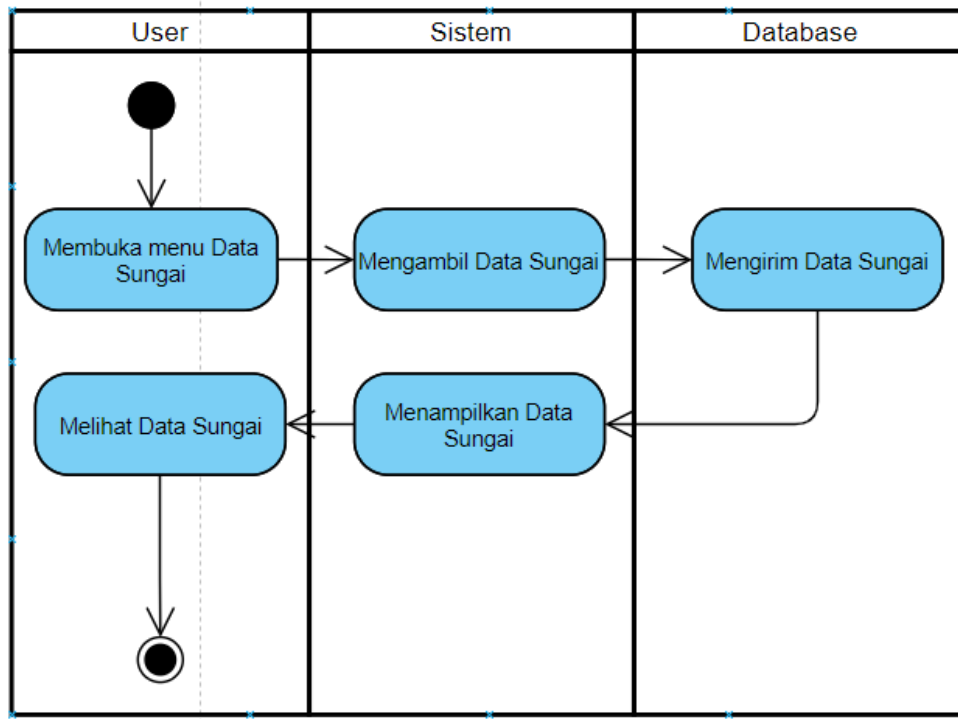
Gambar 3. 6 Activity Diagram Menu Monitoring

Adapun alur kerja saat melakukan *Monitoring* adalah sebagai berikut:

1. *User* membuka aplikasi.
2. Sistem akan menampilkan halaman *Login*.
3. *User* memasukkan *username* dan *password*.
4. Ketika *user* tidak menGISi sesuai *username* dan *password* maka akan kembali ke halaman *Login*, tetapi jika *user* menGISi sesuai *username* dan *password* maka akan ditampilkan halaman *dashboard*.
5. *User* memilih menu *Monitoring*.
6. *User* melakukan *Monitoring*.
7. *User* menutup aplikasi.

c. *Activity Diagram Menu Data Sungai*

Pada Activity Diagram Menu Data Sungai *user* memilih menu tersebut lalu sistem akan menampilkan menu data sungai yang di*Monitoring*. Adapun gambar dari Activity diagram data sungai dapat dilihat pada gambar 3.7.



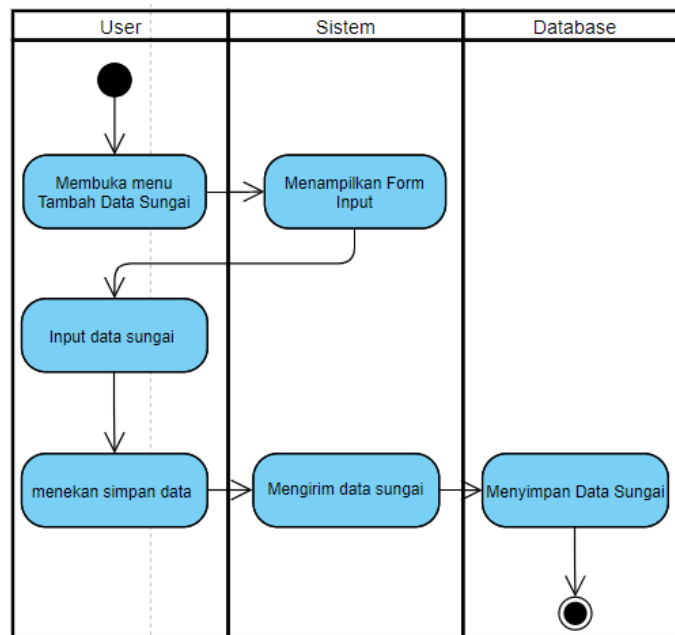
Gambar 3. 7 Activity Diagram Menu Data Sungai

Adapun alur kerja saat melakukan lihat data sungai adalah sebagai berikut:

1. *User* membuka aplikasi.
2. Sistem akan menampilkan halaman *Login*.
3. *User* memasukkan *username* dan *password*.
4. Ketika *user* tidak men*GISi* sesuai *username* dan *password* maka akan kembali ke halaman *Login*, tetapi jika *user* men*GISi* sesuai *username* dan *password* maka akan ditampilkan halaman *dashboard*.
5. *Ueser* memilih menu Data sungai.
6. *User* melihat Data sungai.
7. *User* menutup aplikasi.

d. *Activity Diagram Menu Tambah Data*

Pada *Activity Diagram Menu* tambah data Sungai *user* memilih menu tersebut lalu sistem akan menampilkan *form input* untuk men~~GIS~~Si data sungai setelah melakukan pen~~GIS~~Sisan data, *user* menekan *button* tambah dan sistem melakukan proses penyimpanan data. Adapun *activity diagram* menu tambah data dapat dilihat pada Gambar 3.8.



Gambar 3. 8 Activity Diagram Menu Tambah Data

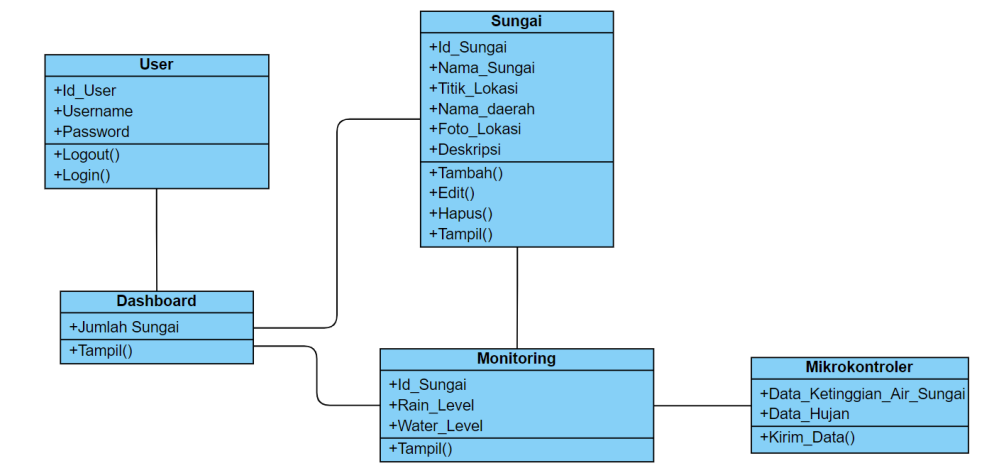
Adapun alur kerja saat melakukan tambah data sungai adalah sebagai berikut:

1. *User* membuka aplikasi.
2. Sistem akan menampilkan halaman *Login*.
3. *User* memasukkan *username* dan *password*.
4. Ketika *user* tidak men~~GIS~~Si sesuai *username* dan *password* maka akan kembali ke halaman *Login*, tetapi jika *user* men~~GIS~~Si sesuai *username* dan *password* maka akan ditampilkan halaman *dashboard*.

5. *User* memilih menu tambah data sungai.
6. *User* menambah data sungai.
7. *User* tekan simpan data.
8. Sistem menyimpan data.
9. *User* menutup aplikasi.

3.7.5 Perancangan *class*

Diagram Class diagram adalah diagram yang menjelaskan tentang pemodelan sistem berorientasi objek. *Class diagram* juga menunjukan hubungan *Class* dengan sistem yang sedang dibangun dan bagaimana mereka saling berinteraksi untuk mencapai suatu tujuan. Adapun contoh diagram *Class* yang akan dibangun dapat dilihat pada Gambar 3.9.



Gambar 3. 9 Diagram Class

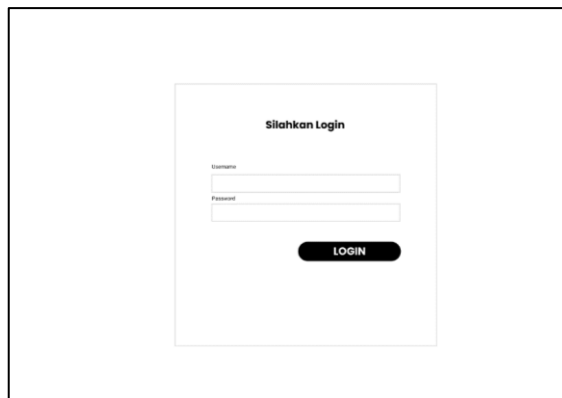
3.7.6 Rancangan Antar Muka Sistem (*Interface*)

Pada rancangan antar muka sistem, terdiri atas rancangan interface sistem berbasis android. Rancangan interface sistem berbasis android digunakan untuk *Monitoring*.

- Rancangan Antar Muka Sistem Berbasis Web

1. Halaman *Login*.

Pada halaman *user* memasukkan *username* dan *password* sesuai agar dapat masuk ke halaman *dashboard* sistem. Adapun contoh tampilan halaman login dapat dilihat pada Gambar 3.10.

The image shows a login form within a light gray border. At the top center, the text "Silahkan Login" is displayed. Below this, there are two input fields: the first is labeled "Username" and the second is labeled "Password". Both fields are empty. Below the password field is a black button with the word "LOGIN" in white capital letters.

Gambar 3. 10 Halaman *Login*

2. Halaman *Dashboard*

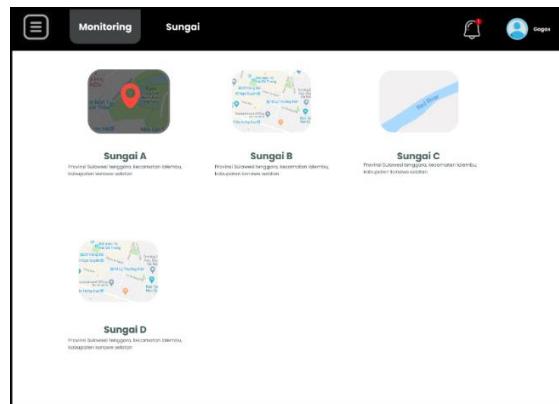
Pada halaman *dashboard* ditampilkan beranda pilihan menu untuk melakukan salahsatunya yaitu menu *Monitoring* ketinggian air sungai dan menu sungai untuk melihat data sungai. Adapun contoh tampilan *dashboard* dapat dilihat pada Gambar 3.11.

The image shows a dashboard interface. At the top, there is a dark header bar with a hamburger menu icon on the left, followed by the text "Monitoring" and "Sungai". On the right side of the header, there is a bell icon and a user profile icon labeled "Gagas". Below the header, the main content area has a white background. It starts with the text "Selamat datang Gagas" in a large font. Below this, there is a dark horizontal bar containing the text "Total sungai : 4" in yellow. The rest of the dashboard area is empty.

Gambar 3. 11 Halaman *Dashboard*

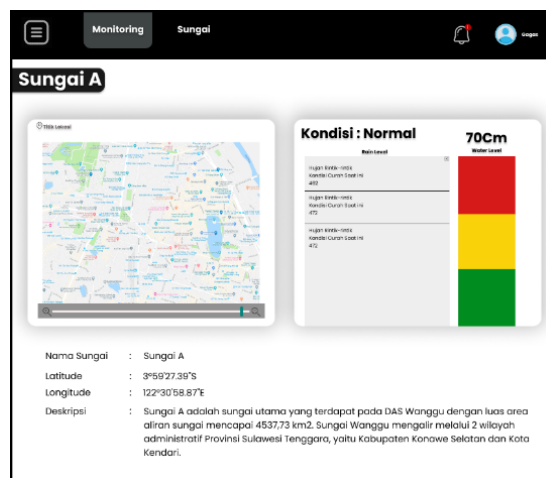
3. Halaman *Monitoring*

Pada halaman ini *user* dapat melihat beberapa sungai yang sedang *dimonitoring* lalu ketika *user* memilih salah satu sungai maka *user* akan melakukan *Monitoring* pada sungai. Adapun contoh tampilan halaman *monitoring* dapat dilihat pada Gambar 3.12.



Gambar 3. 12 Halaman *Monitoring*

Adapun contoh halaman *monitoring* sungai dapat dilihat pada Gambar 3.13.

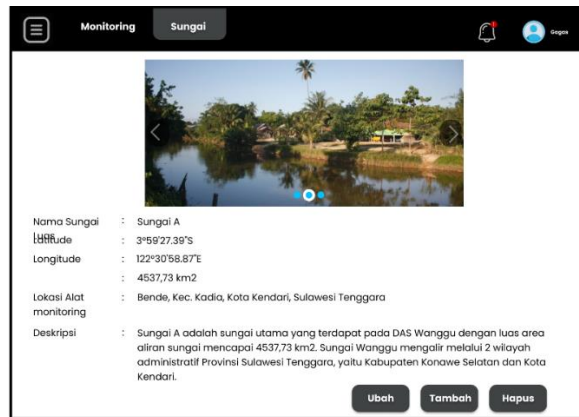


Gambar 3. 13 Halaman *Monitoring* Sungai

4. Halaman Sungai

Pada halaman sungai menampilkan informasi tentang data-data sungai yang telah dipasang alat *Monitoring* dan pada halaman ini terdapat beberapa *button*

ubah, tambah dan hapus. Adapun tampilan halaman sungai dapat dilihat pada Gambar 3.14.



Gambar 3.14 Halaman sungai

5. Halaman Tambah data sungai

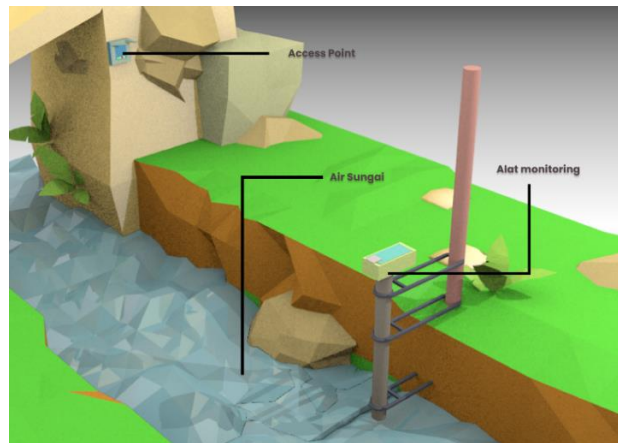
Pada halaman ini sistem memberi akses kepada *user* untuk dapat menambah data sungai yang akan *dimonitoring* lalu data tersebut akan disimpan lalu ditampilkan datanya pada menu sungai. Adapun tampilan halaman data sungai dapat dilihat pada Gambar 3.15.

Gambar 3. 15 Halaman Tambah data sungai

- Rancangan Alat *Monitoring*

1. Tampilan *Monitoring*

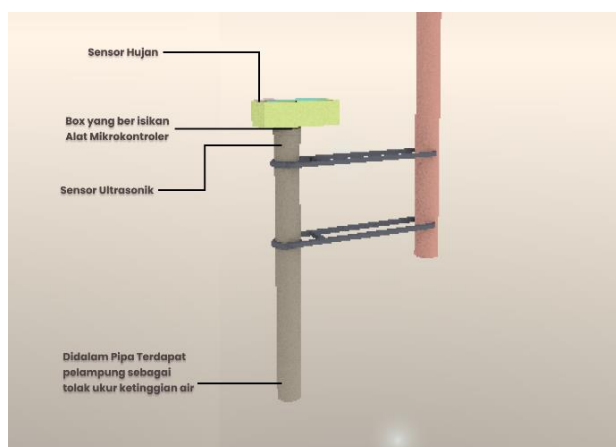
Pada Gambar 3.16 menjelaskan bahwa letak alat saat melakukan *Monitoring* ketinggian air di sungai.



Gambar 3. 16 Tampilan *Monitoring*

2. Alat *Monitoring*

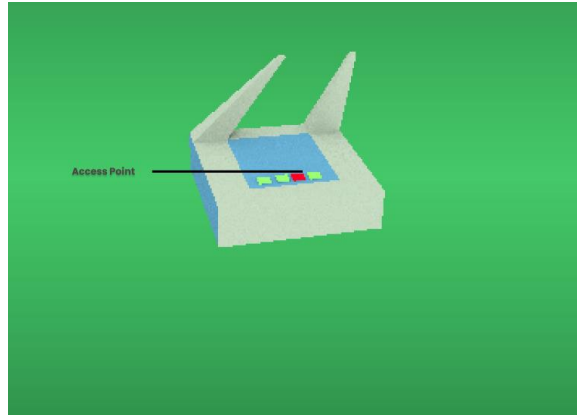
Gambar 3.17 menampilkan struktur yang ada pada alat *Monitoring* ketinggian air yang terdiri dari 2 sensor yaitu *rain* sensor yang berfungsi untuk mendeteksi hujan sensor ultrasonik yang berada di bawah *box* mendeteksi ketinggian air dengan menggunakan pelampung yang berada di dalam pipa yang berfungsi sebagai tolak ukur ketinggian air.



Gambar 3. 17 Alat *Monitoring*

3. Access Point

Pada Gambar 3.18 menampilkan gambar Access Point untuk mengirimkan notifikasi dan informasi pada sistem dari hasil *Monitoring*.



Gambar 3. 18 Access Point

3.7.7 Rencana Pengujian Sistem

1. Sebelum Penerapan Alat

Sebelum sistem diterapkan, proses efakuasi dilakukan setelah air membanjiri rumah warga yang berada dekat sungai yang rentan meluap akibat diguyur hujan sehingga banyak menyebabkan kerugian yang dialami oleh sebagian besar warga yang tinggal di sekitar daerah rawan banjir. Hal ini terjadi akibat tidak adanya pengawasan dari badan penanggulangan bencana daerah yang dilakukan secara langsung.

2. Setelah Penerapan Alat

Pada dasarnya sebelum terjadi hujan ketinggian maksimal air sungai mencapai sekitar 70 cm. Pengujian sensor ultrasonik HC-SR04 dilakukan dengan cara mendeteksi ketinggian air sungai apakah air sudah dalam ketinggian maksimal atau minimum. Tahap pengujian dilakukan dengan menguji tingkat keakuratan Sensor Ultrasonik HC-SR04, *Rain Sensor* dan pengujian blackbox dengan pengujian hasil sistem.

- a) Sensor Ultrasonik HC-SR04. Adapun contoh pengujian keakuratan sensor Ultrasonik HC-SR04 dapat dilihat pada tabel 3.5.

Tabel 3. 5 Contoh tabel pengujian keakuratan sensor Ultrasonik HC-SR04

| No | Level Air | Ketinggian Air Sebenarnya (cm) | Pembacaan Sernsor Ultrasonik HC-SR04 (cm) | Rentan Waktu | Buzzer | Selisih (Ketinggian Air Sebenarnya – pembacaan sensor) |
|------------------------|-----------|--------------------------------|---|--------------|--------|--|
| 1. | Level 1 | 50 - 70 | 51 - 71 | 2 menit | Diam | 1 |
| 2. | Level 2 | 70 - 90 | 71 - 91 | 2 Menit | Diam | 1 |
| 3. | Level 3 | 90 – 110 | 91 -111 | 1 Menit | Bunyi | 1 |
| 4. | Level 4 | 110 – 130 | 111 - 131 | 1 Menit | Bunyi | 1 |
| Rata-rata selisih data | | | | | | 1 |

Pada tabel 3.5 pengujian sensor dilakukan melalui beberapa *level*, pada *level* 1 dan 2 dimana merupakan ketinggian normal air sungai dengan kondisi *buzzer* diam, sehingga untuk ketinggian air sebenarnya dan hasil pembacaan sensor memiliki selisih 1 angka dengan rentan waktu sekitar 2 menit. Pada *level* 3 dan 4 merupakan ketinggian maksimal pada *level* ini kondisi *buzzer* berubah menjadi berbunyi, dengan rata – rata rentan waktu sekitar 1 menit.

Dan pada *level* 2 untuk mencapai ketinggian air dari 70 cm ke ketinggian 90 cm yang memiliki selisih 1 angka dengan pembacaan sensor mempunyai rentan waktu sekitar 1 menit untuk mencapai ketinggian 90 cm. Rentan waktu bertujuan untuk mengetahui seberapa cepat air sungai mencapai titik level tertinggi pada alat monitoring.

b) *Rain Sensor*. Adapun contoh pengujian *rain* sensor dapat dilihat pada tabel 3.6.

Tabel 3. 6 Contoh pengujian *Rain Sensor*

| No | Pembacaan sensor | Kondisi hujan | Rentan Waktu hujan | Keterangan |
|----|------------------|-----------------------|--------------------|------------|
| 1 | 462 | Hujan Rintik - Rintik | 20 Menit | Aman |
| 2 | 480 | Hujan Sedang | 15 Menit | Aman |
| 3 | 501 | Hujan Deras | 30 menit | Bahaya |

Pada tabel 3.6 menjelaskan proses pengujian *rain* sensor untuk kondisi hujan yang rintik-rintik akan mencapai angka pembacaan sensor yaitu sekitar 462 yang terjadi dalam kurun waktu 20 menit sehingga masih memberikan keterangan aman, dan untuk kondisi hujan deras akan mencapai angka pembacaan sensor sekitar 501 yang terjadi dalam kurun waktu 30 menit sehingga memberikan keterangan bahaya dan berpotensi mempercepat meluapnya air sungai.

Pengujian *blackbox*. Adapun contoh pengujian *blackbox* dapat dilihat pada tabel 3.7.

Tabel 3. 7 Contoh pengujian *Blackbox*

| Input/Event | Output | Hasil Uji |
|--|------------------------------------|-----------|
| Melakukan login dengan menginputkan <i>username</i> dan <i>password</i> yang benar | Menampilkan halaman dashboard | Berhasil |
| Memilih menu <i>monitoring</i> | Menampilkan menu <i>monitoring</i> | Berhasil |
| Memilih menu sungai | Menampilkan data sungai | Berhasil |

Pada tabel 3.7 dapat dilihat proses pengujian sistem dengan menggunakan metode pengujian *blackbox* pada tabel pengujian ini memiliki 3 komponen utama yaitu: *input/event*, *output* dan hasil uji dimana dari ketiga komponen ini masing – masing memiliki isi yang merupakan bentuk representasi dari alur atau tatacara menjalankan sistem yang telah dibuat.

DAFTAR PUSTAKA

- Abdulrachman, I., Trianto, B., & Kurniawan, D. (2017). Implementasi Internet of Things (IoT) pada Sistem Penanganan Banjir. *Seminar Nasional Informatika Dan Aplikasinya (SNIA) 2017, 1*(September), 67–70.
- Anna, A. N., Darnoto, S., Astuti, D., Studilingkungan, P., Surakarta, U. M., & Geografis, S. I. (2017). *Monitoring Kualitas Air Sungai Di Kota Surakarta*. 226–233.
- Ardianto, I., & Nst, M. R. K. (2016). Perancangan aplikasi video streaming web menggunakan xampp di universitas al-washliyah labuhanbatu. *Infotek, 1*(3), 1–6.
- Berbasis Android Tarmidi, L., taqwa, A., Silvia Handayani, A., Teknik Elektro, J., Studi Teknik Telekomunikasi Politeknik Negeri Sriwijaya Jl Sriwijaya Negara, P., & Besar Palembang, B. (2019). *Penerapan Wireless Sensor Network Sebagai Monitoring*. 224–230.
- Gunawan, I., Akbar, T., & Giyandhi Ilham, M. (2020). Prototipe Penerapan Internet Of Things (Iot) Pada *Monitoring* Level Air Tandon Menggunakan Nodemcu Esp8266 Dan Blynk. *Infotek : Jurnal Informatika Dan Teknologi, 3*(1), 1–7. <https://doi.org/10.29408/jit.v3i1.1789>
- Hidayat, H., Hartono, & Sukiman. (2017). Pengembangan Learning Management System (LMS) untuk Bahasa Pemrograman PHP. *Jurnal Ilmiah Core IT: Community Research Information Technology, 5*(1), 20–29.
- Hidayati, S. N. (2018). Sistem *Monitoring* Tinggi Muka Air Pada Sungai Berbasis Mikrokontroler dan *Website* Geographic Information System (WebGIS). *Jurnal Teknologi Informasi Dan Elektro*.
- Margareth, T. (n.d.). *PENGERTIAN DAN APLIKASINYA*.
- Mubarok, F., Harliana, H., & Hadijah, I. (2015). Perbandingan Antara Metode RUP dan Prototype Dalam Aplikasi Penerimaan Siswa Baru Berbasis Web. *Creative Information Technology Journal, 2*(2), 114. <https://doi.org/10.24076/citec.2015v2i2.42>
- Murbawan, I., Ma'ruf, A., & Manan, A. (2017). *Kesiapsiagaan Rumah Tangga Dalam Mengantisipasi Bencana Banjir Di Daerah Aliran Sungai (Das) Wanggu (Studi*

- Bencana Banjir Di Kelurahan Lepo-Lepo Kota Kendari*). 3(2), 59–69.
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). *Pengujian aplikasi menggunakan black box testing boundary value analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)*. I(3), 31–36.
- Novianto, D. (2016). Implementasi Sistem Informasi Pegawai (Simpeg) Berbasis Web Menggunakanframework Codeigniter Dan Bootstrap. *Ilmiah Informatika Global*, 7(1), 10–16.
- Sonita, A., & Fardianitama, R. F. (2018). Aplikasi E-Order Menggunakan Firebase Dan Algoritme Knuth. *Jurnal Pseudocode*, 5(2), 38–45.
- Suhartanto, M. (2016). Pembuatan *Website* Sekolah Menengah Pertama Negeri 3 Delanggu Dengan Menggunakan Php Dan Mysql. *Sentra Penelitian Engineering Dan Edukasi*, 4(1), 1–8.
- Ulumuddin, U., Sudrajat, M., Rachmildha, T. D., Ismail, N., & Hamidi, E. A. Z. (2017). Prototipe Sistem *Monitoring* Air Pada Tangki Berbasis Internet of Things Menggunakan Nodemcu Esp8266 Sensor dan Ultrasonik. *Seminar Nasional Teknik Elektro 2017, 2016*, 100–105. <https://doi.org/978-602-512-810-3>