

**VISUALISASI BANJIR SECARA SPASIAL BERDASARKAN
DATA SENSOR KETINGGIAN BERBASIS
MIKROKONTROLER ARDUINO(*PROTOTYPE*)**

SKRIPSI

Program Studi Sarjana Informatika
Jurusan Informatika

Oleh:
Rizky Aditya Saputra
NIM D1041151077



FAKULTAS TEKNIK
UNIVERSITAS TANJUNGPURA
PONTIANAK
2022

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Rizky Aditya Saputra

NIM : D1041151077

Menyatakan bahwa dalam skripsi yang berjudul “Visualisasi Banjir Secara Spasial Berdasarkan Data Sensor Ketinggian Berbasis Mikrokontroler Arduino(*Prototype*)” tidak terdapat karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu perguruan tinggi manapun. Sepanjang pengetahuan Saya, tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam Daftar Pustaka.

Demikian pernyataan ini dibuat dengan sebenar-benarnya. Saya sanggup menerima konsekuensi akademis dan hukum di kemudian hari apabila pernyataan yang dibuat ini tidak benar.

Pontianak, 11 Maret 2022

Rizky Aditya Saputra
NIM D1041151077



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS TANJUNGPURA
FAKULTAS TEKNIK

Jl. Prof. Dr. H. Hadari Nawawi Pontianak 78124 Telp./Fax. 0561 740186

HALAMAN PENGESAHAN

VISUALISASI BANJIR SECARA SPASIAL BERDASARKAN DATA SENSOR
KETINGGIAN BERBASIS MIKROKONTROLER ARDUINO(*PROTOTYPE*)

Program Studi Sarjana Informatika
Jurusan Informatika

Oleh:

Rizky Aditiya Saputra
NIM D1041151077

Susunan Penguji Skripsi:

Ketua,

Penguji Utama,

Dr. Yus Sholva, S.T.,M.T.
NIP. 19741019 200312 1 002

Heri Priyanto, S.T., M.T
NIP. 19750412 200312 1001

Sekretaris,

Penguji Pendamping,

Hafiz Muhardi, S.T., M.Kom.
NIDK. 8885370018

M.Azhar Irwansyah, S.T., M.Eng
NIP. 19850606 200812 1002.

Pontianak, 22 Juni 2022
Dekan,

Dr. rer. nat. Ir. R. M. Rustamaji, M.T.
NIP. 19680116 199403 1 003

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah Subhanahu Wa Ta'ala yang senantiasa memberikan limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penelitian ini yang “Berjudul Visualisasi Banjir Secara Spasial Berdasarkan Data Sensor Ketinggian Berbasis Mikrokontroler Arduino(*Prototype*)”.

Dalam penyelesaian penelitian ini, penulis secara langsung maupun tidak langsung telah mendapatkan bimbingan, pengarahan dan bantuan dari beberapa pihak. Untuk itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Dr.rer.nat.Ir. R.M. Rustamaji, M.T, IPU. selaku Dekan Fakultas Teknik Universitas Tanjungpura.
2. Bapak Dr. Yus Sholva, S.T., M.T. selaku ketua Jurusan Informatika Fakultas Teknik Universitas Tanjungpura.
3. Bapak Muhammad Azhar Irwansyah, S.T., M.Eng. selaku ketua Program Studi Informatika Universitas Tanjungpura.
4. Dr. Yus Sholva, S.T., M.T selaku dosen pembimbing utama.
5. Hafiz Muhardi, S.T., M.Kom selaku dosen pembimbing kedua.
6. Heri Priyanto, S.T., M.T selaku dosen penguji utama.
7. M.Azhar Irwansyah, S.T., M.Eng selaku dosen penguji kedua.
8. Dan pihak-pihak lain yang secara langsung atau tidak membantu penulis.

Penulis menyadari dalam penyusunan skripsi ini masih belum sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun demi sempurnanya skripsi ini sehingga dapat bermanfaat bagi masyarakat serta untuk pengembangan wawasan dan peningkatan ilmu pengetahuan bagi kita semua.

Pontianak, 11 Maret 2022
Penulis,

Rizky Aditiya Saputra

ABSTRAK

Banjir adalah peristiwa bencana alam yang terjadi ketika aliran air yang berlebihan merendam daratan. Banjir dapat terjadi karena hujan besar yang tidak berhenti pada waktu yang cukup lama sehingga air hujan akan memenuhi - yang ada. Tujuan dari penelitian ini adalah untuk menghasilkan sebuah Web dan prototipe sistem yang terdiri dari prototipe pengukur ketinggian air pada berbasis teknologi tertanam dan aplikasi berbasis web yang keduanya saling berkomunikasi melalui protocol internet untuk memberikan informasi kepada pengguna dan masyarakat. Sesuai dengan tujuan penelitian, penelitian dilakukan dengan desain sistem meliputi perancangan UML (Unified Modelling Language) untuk menggambarkan proses kerja dari sisi perangkat lunak. Sistem informasi Manajemen ini dibuat menggunakan bahasa pemrograman PHP (Framework CodeIgniter) menggunakan database MySQL. Pengujian menggunakan pengujian blackbox yang menggunakan metode Automatic Testing (Katalon) dan UAT (User Acceptance Testing) yang menggunakan metode LSR (Likert Summated Rating). Hasil akhir dari tugas akhir ini adalah Visualisasi banjir secara spasial berdasarkan data Sensor Ketinggian berbasis Mikrokontroler Arduino(Prototype).Hasil ide dari penulis untuk membangun sebuah sistem yang bisa membantu pihak terkait dalam memantau ketinggian air di secara real time melalui jaringan internet.

Kata kunci: UML (*Unified Modelling Language*),PHP (Framework CodeIgniter), *blackbox*, UAT (*User Acceptance Testing*).

ABSTRACT

Floods are natural disaster events that occur when excessive water flows submerge land. Floods can occur due to heavy rains that do not stop for a long time so that rainwater will fill the existing rivers. The purpose of this research is to produce a Web and a prototype system consisting of a prototype for measuring water levels in rivers based on embedded technology and web-based applications, both of which communicate with each other through internet protocols to provide information to users and the public. In accordance with the research objectives, the research was carried out with system design including the design of UML (Unified Modeling Language) to describe the work process from the Software side. This Management information system was created using the PHP programming language (Framework CodeIgniter) using a MySQL database. Tests using blackbox testing using the Automatic Testing (Katalon) and UAT (User Acceptance Testing) methods using the LSR (Likert Summated Rating) method. The final result of this final project is a spatial flood visualization based on Microcontroller-based Altitude Sensor data.

Keywords: UML (Unified Modeling Language), PHP (Framework Codeigniter), Blackbox, UAT (User Acceptance Testing).

DAFTAR ISI

Halaman Pernyataan	ii
Halaman Pengesahan.....	iii
Kata Pengantar	iv
Abstrak.....	v
Abstract.....	vi
Daftar Isi	vi
Daftar Tabel.....	ix
Daftar Gambar	x
Daftar Kode Program	xii
Bab I Pendahuluan.....	13
1.1 Latar Belakang.....	13
1.2 Perumusan Masalah.....	14
1.3 Tujuan Penelitian.....	15
1.4 Pembatasan Masalah	15
1.5 Sistematika Penulisan.....	15
Bab II Tinjauan Pustaka.....	17
2.1 Kajian Terkait.....	17
2.2 NODE MCU V3	21
2.3 Mikrokontroler	25
2.3.1 Jenis-jenis Mikrokontroler	26
2.4 Modul Arduino	26
2.4.1. Perangkat Lunak Pemrograman Arduino	27
2.4.2. Papan (Board) Arduino Nano.....	29
2.5 Modul Sensor HC-SR04.....	31
2.6 Website	33
2.7 HTML.....	34
2.8 <i>Framework</i> dalam pengembangan <i>Website</i>	34
2.9 Sistem Informasi Manajemen.....	Error! Bookmark not defined.
2.10 Web Service.....	38
2.11 Javascript Object Notation	39
2.12 <i>Unified Modelling Language</i>	Error! Bookmark not defined.
2.13 Basis Data.....	45
2.14 BlackBox Testing	46
Bab III Metodologi Penelitian	49
3.1 Metodologi Penelitian	49
3.1.1. Bahan Penelitian.....	49

3.1.2.	Perangkat Lunak.....	49
3.1.3.	Perangkat Keras.....	50
3.1.4.	Metode Penelitian.....	51
3.2	Perancangan Sistem.....	54
3.2.1.	Analisis Kebutuhan Sistem	54
3.2.2.	Antarmuka Sistem	74
3.2.3.	Perancangan <i>Database</i>	80
3.3	Perancangan Pengujian Sistem.....	82
3.3.1	Pengujian <i>Hardware</i>	82
3.4	Pembuatan Sistem	86
3.5	Pengujian Sistem	86
Bab IV	Hasil dan Analisis.....	88
4.1	Hasil Implementasi	88
4.1.1.	Rangkaian Mikrokontroler Sistem	88
4.1.2.	Proses Koneksi dan Komunikasi Data <i>Hardware</i> Sistem	89
4.1.3.	Prototipe Alat Monitoring Ketinggian Air	Error!
	Bookmark not defined.	
4.1.4.	Hasil Implementasi <i>Software</i> Sistem.....	93
4.2	Pengujian	101
4.2.1.	Pengujian <i>Software</i>	101
4.2.2.	Pengujian <i>Hardware</i>	104
4.3	Analisis Fungsi <i>Monitoring</i> Sistem.....	109
4.4	Analisis Hasil Pengujian Secara Keseluruhan.....	111
Bab V	Hasil dan Analisis.....	112
5.1	Kesimpulan.....	112
5.2	Saran	1
Daftar pustaka		2

DAFTAR TABEL

Tabel 2. 1 Tabel Perbandingan Penelitian.....	17
Tabel 2. 2 Tabel Penelitian Dilakukan.....	21
Tabel 2. 4 Spesifikasi NODEMCU V3.....	23
Tabel 2. 5 Spesifikasi Arduino Nano.....	30
Tabel 2. 6 Relasi Pada Kelas <i>Diagram</i>	44
Tabel 2. 7 Multiplicity.....	44
Tabel 3. 1 Definisi Aktor.....	59
Tabel 3. 2 Definisi Use Case.....	60
Tabel 3. 3 Skenario <i>Use Case Diagram</i> : Melihat Peta Monitoring.....	61
Tabel 3. 4 Skenario <i>Use Case Diagram</i> : Tentang Aplikasi.....	62
Tabel 3. 5 Skenario <i>Use Case Diagram</i> : Login.....	62
Tabel 3. 6 Skenario <i>Use Case Diagram</i> : Logout.....	63
Tabel 3. 7 Skenario <i>Use Case Diagram</i> : Cek Status Login.....	63
Tabel 3. 8 Skenario <i>Use Case Diagram</i> : Melihat Grafik.....	65
Tabel 3. 9 Skenario <i>Use Case Diagram</i> : Tambah Sensor.....	65
Tabel 3. 10 Skenario <i>Use Case Diagram</i> : Ubah Sensor.....	66
Tabel 3. 11 Skenario <i>Use Case Diagram</i> : Hapus Sensor.....	68
Tabel 3. 12 Skenario <i>Use Case Diagram</i> : Lihat Sensor.....	70
Tabel 3. 13 Spesifikasi Tabel <i>Admins</i>	80
Tabel 3. 14 Spesifikasi Tabel Sensor.....	80
Tabel 3. 16 Spesifikasi Tabel Data.....	81
Tabel 3. 18 Rancangan Tabel Pengujian Rangkaian Mikrokontroler Secara Umum.....	82
Tabel 3. 14 Rancangan Tabel Hasil Pengukuran Sensor pada Permukaan Air Datar.....	84
Tabel 3. 15 Rancangan Tabel Hasil Pengukuran Sensor pada Permukaan Air	84
Tabel 3. 16 Skenario Pengujian Aplikasi Web.....	85
Tabel 3. 17 Rancangan Tabel Pengujian Black Box.....	85
Tabel 4. 1 Komponen Pembangun Rangkaian Mikrokontroler.....	88
Tabel 4. 2 Komponen-komponen Pembangun Prototipe <i>Hardware</i> Sistem.....	95
Tabel 4. 3 Skenario Pengujian Black Box Testing.....	101
Tabel 4. 4 Pengujian Halaman Pembuka dan Peta Monitoring.....	102
Tabel 4. 5 Pengujian Halaman Login.....	102
Tabel 4. 6 Pengujian Halaman Admin.....	103
Tabel 4. 7 Pengujian Form Tambah Sensor.....	103
Tabel 4. 8 Pengujian Form Ubah Sensor.....	103

DAFTAR GAMBAR

Gambar 2. 1 ESP 8266 NODEMCU V322.....	23
Gambar 2. 2. Skematik posisi Pin NodeMcu Dev Kit v3	24
Gambar 2. 3 Tampilan awal <i>Software</i> arduino de.....	28
Gambar 2. 4 Tampilan Utama Sketch Arduino.....	29
Gambar 2. 5 Board System Minimum Arduino Nano	30
Gambar 2. 6 Konfigurasi Pin dan Tampilan Sensor HC-SR04.....	31
Gambar 2. 7 Prinsip kerja HC-SR04.....	32
Gambar 2. 8 Timing <i>Diagram</i> Pengoperasian Sensor HC-SR04.....	33
Gambar 2. 9 Tag HTML.	34
Gambar 2. 10 Tag CSS.	35
Gambar 2. 11 Pola MVC Codeigniter.....	36
Gambar 2. 12 Tag Model.	37
Gambar 2. 13 Tag Controller.	38
Gambar 2. 14 Tag JSON.	40
Gambar 2. 15 Keterangan activity <i>Diagram</i>	43
Gambar 2. 16 Aplikasi Katalon Studio	Error! Bookmark not defined.
Gambar 3. 1 <i>Diagram</i> alir penelitian.	51
Gambar 3. 2 Arsitektur Sistem Monitoring Ketinggian Air	54
Gambar 3. 3 Desain Arsitektur Aplikasi Web	56
Gambar 3. 4 Desain Prototipe Wadah Air	56
Gambar 3. 5 Perancangan Perhitungan Ketinggian Air pada Wadah.....	58
Gambar 3. 6 Struktur Antarmuka <i>Website</i> Sistem	75
Gambar 3. 7 Desain Halaman Login Admin Aplikasi	77
Gambar 3. 8 Desain Halaman Tentang Aplikasi.....	77
Gambar 3. 9 Desain Halaman Peta Monitoring	78
Gambar 3. 10 Desain Halaman Peta Monitoring	79
Gambar 3. 11 Desain Halaman Master Data Grafik	79
Gambar 4. 1 Rangkaian Mikrokontroler Sistem	88
Gambar 4. 2 Hasil Baca Sensor pada Serial Monitor Arduino IDE.....	90
Gambar 4. 3 Hasil Implementasi <i>Hardware</i> Sistem Keseluruhan	94
Gambar 4. 4 Hasil Implementasi <i>Hardware</i> Sistem Secara Keseluruhan	95

Gambar 4. 5 Tampilan Halaman Utama Website	96
Gambar 4. 6 Tampilan Halaman Tentang Aplikasi	96
Gambar 4. 7 Tampilan Halaman Peta <i>Monitoring</i>	97
Gambar 4. 8 Halaman Login Website Sistem.....	97
Gambar 4. 9 Halaman Dashboard Admin	Error! Bookmark not defined.
Gambar 4. 10 <i>Form</i> Tambah Sensor	98
Gambar 4. 11 Grafik Hasil Baca Sensor	99
Gambar 4. 12 Form Ubah Data	99
Gambar 4. 13 Form Ubah Data Sensor	100
Gambar 4. 14 Form Hapus Sensor	100

DAFTAR KODE PROGRAM

Kode Program 4. 1 Skrip mengambil data ketinggian air	89
Kode Program 4. 2 Skrip Koneksi ke Jaringan Internet pada	90
Kode Program 4. 3 Skrip Komunikasi Serial pada	91
Kode Program 4. 4 Skrip Sebagai Penerima Data	92
Kode Program 4. 5 Skrip Setting IP address atau Web Server Tujuan.....	92
Kode Program 4. 6 Skrip Upload Variabel “Data” ke Server.....	92

BAB I PENDAHULUAN

1.1 Latar Belakang

Banjir adalah peristiwa bencana alam yang terjadi ketika aliran air yang berlebihan merendam daratan. Dalam arti "air mengalir", kata ini juga dapat berarti masuknya pasang laut yang mengakibatkan volume air di suatu badan air seperti atau danau meluap atau melimpah ke daratan. Banjir dapat terjadi karena hujan besar yang tidak berhenti pada waktu yang cukup lama sehingga air hujan akan memenuhi - yang ada. Banjir juga dapat terjadi karena kurangnya daerah resapan air ketika menghadapi curah hujan yang tinggi, kondisi geografis suatu daerah yang rendah sehingga mendapat kiriman air dari daerah yang lebih tinggi, dan lain sebagainya.

Bencana merupakan suatu kejadian atau peristiwa yang memberikan kerugian yang besar pada masyarakat, yang bersifat merusak, merugikan dan mengambil waktu yang panjang untuk pemulihannya (Purnomo et al., 2018). Pengertian ini lebih diperjelas dalam Undang-undang Nomor 24 Tahun 2007 tentang Penanggulangan Bencana, bencana merupakan rangkaian peristiwa yang memberikan dampak langsung berupa ancaman terhadap kehidupan masyarakat yang disebabkan, baik oleh faktor alam atau faktor non alam sehingga dampak langsung yang ditimbulkan adalah kerusakan lingkungan, kerugian harta benda, dampak psikologis, serta timbulnya korban jiwa. Banjir telah menjadi persoalan tiada akhir bagi manusia di seluruh dunia dari dulu, sekarang dan yang akan datang. Banjir merupakan bencana yang terjadi akibat dari aktivitas geologi (alami) atau aktivitas manusia dan bahkan bisa secara bersamaan diakibatkan oleh alam dan manusia.

Kota Pontianak terbagi oleh keberadaan Kapuas yang mengalir dari arah Tenggara dan bermuara di Laut Natuna. Topografi Kota Pontianak relatif datar berkisar antara 0,1 s/d 1,5 meter diatas permukaan laut. Letaknya yang dekat dengan laut dan sehingga menyebabkan wilayah Kota Pontianak secara umum dipengaruhi oleh pasang surut air laut sehingga mudah tergenang (Kurnia, M. I., Gusti Z. M., 2018). Kota Pontianak berada di lokasi yang diidentifikasi berpotensi

mengalami banjir/genangan yang cukup dalam dengan waktu yang cukup lama ketika terjadinya hujan. Drainase di Kota Pontianak sudah cukup baik, meski masih banyak ruas jalan utama yang terendam banjir akibat hujan yang terjadi hanya beberapa jam saja yang disebabkan curah hujan yang sangat tinggi dan air Kapuas sedang pasang, sehingga antara air Kapuas dan air daratan menjadi sejajar sehingga air tidak bisa turun ke daerah aliran sungai.

Sebagai bagian dari masyarakat yang diharapkan peran sertanya untuk mendukung pemerintahan dalam menangani masalah banjir, maka muncul ide dari penulis untuk membuat suatu sistem dalam skala kecil atau *prototype* untuk membantu memantau ketinggian air secara akurat dengan suatu perangkat sensor yang dapat dipantau secara *real-time* melalui koneksi internet. Kebutuhan dalam membangun sistem yang dapat memvisualisasikan ketinggian air tentunya membutuhkan metode dan alat yang sesuai. Sehingga Tugas Akhir ini dibuat dengan Judul “Visualisasi Banjir Secara Spasial Berdasarkan Data Sensor Ketinggian Berbasis Mikrokontroler Arduino(*Prototype*)”, dengan tujuan untuk memvisualisasikan ketinggian permukaan air sehingga dapat mengambil tindakan untuk melakukan penanganan.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka penulis merancang Visualisasi Banjir Secara Spasial Berdasarkan Data Sensor Ketinggian Berbasis Mikrokontroler, sebuah sistem monitoring yang bisa menyajikan informasi ketinggian air pada . Sistem yang akan dibangun berupa prototipe yang terdiri dari bagian *Hardware* dan *Software*. Yang dimaksud dengan bagian *Hardware* adalah sebuah alat ukur berbasis mikrokontroler yang memiliki kemampuan secara otomatis dapat mengukur jarak antara objek air melalui perangkat yang terhubung dengan internet. Nilai jarak yang dimaksud adalah data ketinggian air seperti yang dimaksud dalam penelitian ini. Setelah itu data akan dikirim ke database server untuk diolah oleh bagian *Software*. Bagian *Software* yang dimaksud adalah sebuah aplikasi berbasis website yang dibangun untuk menyajikan informasi ketinggian air dalam bentuk visual yang menarik sehingga mudah untuk dipahami dan dianalisis oleh pengguna.

Dari uraian di atas, maka permasalahan yang dapat dirumuskan adalah bagaimana merancang sistem yang bisa melakukan tugas monitoring terhadap ketinggian air di kemudia nmenyajikan data hasil monitoring ke dalam sebuah aplikasi berbasis website.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan dari penelitian ini adalah untuk menghasilkan sebuah Web dan prototipe sistem yang terdiri dari prototipe pengukur ketinggian air pada berbasis teknologi tertanam dan aplikasi berbasis web yang keduanya saling berkomunikasi melalui protocol internet untuk memberikan informasi kepada pengguna dan masyarakat.

1.4 Pembatasan Masalah

Adapun batasan masalah pada penelitian ini di antaranya adalah :

- Sistem yang dibangun hanya berupa sebuah prototipe.
- Sistem hanya memvisualisasikan suatu wilayah dengan 1 sensor.
- Sistem tidak mengukur volume air, akan tetapi hanya untuk mengukur ketinggian air di menggunakan sensor jarak HC-SR04 yang berjumlah 1 buah.
- Untuk pengujian berupa sebuah prototipe mini.
- Sistem menggunakan Mikrokontroler ESP 2866 sebagai pengendali sensor.
- jarak dan ESP 8266 sebagai modul untuk mengirim hasil baca sensor ke database melalui internet.

1.5 Sistematika Penulisan

Adapun sistematika dalam penulisan tugas akhir ini tersusun atas lima bab, yang terdiri dari BAB I Pendahuluan, BAB II Tinjauan Pustaka, BAB III Metodologi Penelitian dan Perancangan Sistem, BAB IV Hasil Perancangan dan Analisis Sistem, serta Bab V Kesimpulan dan Saran.

BAB I Pendahuluan merupakan bab pertama dalam penelitian, pada bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, pembatasan masalah dan

sistematika penulisan skripsi.

BAB II Tinjauan Pustaka, merupakan bab yang berisi landasan teori terkait dengan penelitian yang akan di lakukan, yaitu uraian tentang poin-poin penting dari hasil penelitian yang dilakukan oleh peneliti-peneliti sebelumnya.

BAB III Metodologi Penelitian adalah bab yang berisi tentang Alat dan Data Penelitian, Langkah Penelitian seperti Studi Literatur Analisis kebutuhan, pengumpulan data, perancangan sistem, pembuatan perangkat lunak, pengujian program, penarikan kesimpulan

BAB IV Hasil dan Analisis merupakan bab yang berisi Hasil perancangan dan Analisis aplikasi. Bab ini berisi penjelasan aplikasi yang dibangun sesuai dengan isi dari Bab III. Setiap hasil yang disajikan akan dilakukan analisis untuk mengarah kepada suatu kesimpulan.

BAB V Kesimpulan dan Saran adalah bab yang berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran atau rekomendasi untuk perbaikan, pengembangan atau kesempurnaan atau kelengkapan penelitian yang dilakukan.

BAB II TINJAUAN PUSTAKA

2.1 Kajian Terkait

Terdapat beberapa pustaka yang digunakan sebagai referensi dan acuan dalam melakukan penelitian ini. Diantaranya adalah:

Tabel 2. 1 Tabel Perbandingan Penelitian

NO	Penulis	Judul	Keterangan
1.	Eggi Chandra (2017)	Perancangan Sistem Pemantau Ketinggian Sampah Menggunakan Mikrokontroler Arduino dan Aplikasi Berbasis Web	<ul style="list-style-type: none"> ● Membangun sebuah prototipe sistem pemantauan ketinggian sampah menggunakan mikrokontroler Arduino dan aplikasi berbasis web. Mikrokontroler Arduino berfungsi untuk mengambil data ketinggian sampah ● Target penggunaan aplikasi hanya digunakan oleh admin ● Aplikasi berbasis Web
2.	KHAIRUL (2018)	Implementasi Pengendali Sistem Buka Tutup Pintu Air Otomatis Berbasis Arduino Uno R3 Dan Website	<ul style="list-style-type: none"> ● Membangun alat pengendali sistem buka tutup pintu air otomatis berbasis Arduino Uno. ● Target penggunaan aplikasi hanya admin ● Aplikasi berbasis Web

3.	Hengki(2017)	Prototipe Sistem Telemetri Tinggi Muka Air dan Kontrol Pintu Air Otomatis Berbasis Mikrokontroler ATmega328P dan ESP8266	<ul style="list-style-type: none"> ● Telah dibuat sebuah prototipe instrumen untuk mengukur tinggi permukaan air dan kontrol pintu air berdasarkan prinsip kerja dari gelombang ultrasonik. Sistem terdiri dari mikrokontroler ATmega328P, sensor Ultrasonik HC-SR04, modul ESP8266, dan Motor Servo. ●
4.	Rausan Fikri(2015)	Rancang Bangun Sistem Monitoring Ketinggian Permukaan Air Menggunakan Mikrokontroler ATMEGA328P Berbasis Web Service	<ul style="list-style-type: none"> ● Telah dilakukan pembuatan alat sistem monitoring ketinggian permukaan air dengan menggunakan mikrokontroler ATmega328P berbasis web service. ● terdiri atas mikrokontroler ATmega328P, sensor ultrasonik HC-SR04, sensor suhu dan kelembaban DHT-11, modul bluetooth HC-05 dan modul ethernet ENC28J60. ● Sistem juga dilengkapi dengan panel surya dan baterai berkapasitas 35Ah sebagai sumber energi listrik.

5.	Adhitya Permana(2015)	Rancang bangun sistem monitoring Volume dan pengisian air menggunakan sensor ultrasonik Berbasis mikrokontroler avr atmega8	<ul style="list-style-type: none"> ● monitoring persediaan air pada bak penampungan menjadi hal yang penting mengingat ketersediaan air di kota Pontianak terbatas. ● Membuat suatu alat yang dapat melakukan monitoring air secara otomatis. ● Sistem monitoring ketinggian air dengan sensor ultrasonik berbasis mikrokontroler AVR ATmega8 adalah sistem yang dapat mengetahui ketinggian air secara otomatis. ● Interface sistem ini menggunakan handphone yang dihubungkan dengan bluetooth.
6.	SUMARDI SADI & ILHAM SYAH PUTRA(2018)	RANCANG BANGUN MONITORING KETINGGIAN AIR DAN SISTEM KONTROL PADA PINTU AIR BERBASIS ARDUINO DAN SMS GATEWAY	<ul style="list-style-type: none"> ● Penelitian ini bertujuan untuk mengembangkan prototype system peringatan dini banjir ● dengan menggunakan sensor ultrasonic yang diintegrasikan dengan arduino uno untuk mengukur ketinggian air. ● Alat dan bahan yang digunakan yaitu: Arduino Uno, sensor ultrasonic, modul

			<p>GSM shield sebagai media pengirim dan penerima sms</p> <ul style="list-style-type: none"> • <i>Hardware</i>, dan Pembuatan program arduino dengan menggabungkan modul GSM • Shield dan sensor ultrasonic.
7.	Achmad Muzakky(2018)	PERANCANGAN SISTEM DETEKSI BANJIR BERBASIS IoT	<ul style="list-style-type: none"> • Penelitian ini bertujuan untuk monitoring level air secara online sebagai informasi dini terhadap terjadinya banjir. • Monitoring ini menggunakan pendekatan teknologi Internet of things (IoT) agar informasi level dapat diketahui secara real time. Water level sensor digunakan sebagai pembaca data dan Node MCU ESP8266 sebagai pemroses dan mengirimkan data secara nirkabel ke smartphone android lewat aplikasi BLYNK, hasil penelitian ini adalah suatu sistem deteksi level air yang dapat menginformasikan level aman, • siaga, awas serta dapat memberikan notifikasi. Dengan demikian sistem deteksi ini

			<ul style="list-style-type: none"> • akan dapat dimanfaatkan untuk informasi awal terjadinya banjir.
--	--	--	---

Tabel 2. 2 Tabel Penelitian Dilakukan

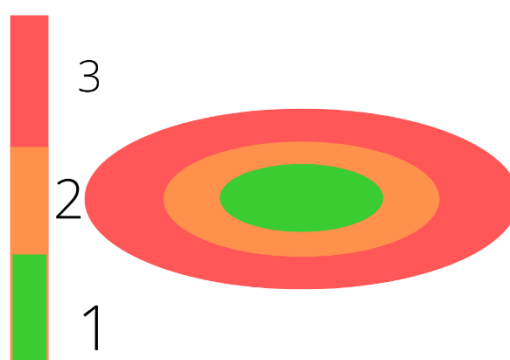
NO	Penulis	Judul	Keterangan
1.	Rizky Aditya Saputra	Visualisasi Banjir Secara Spasial Berdasarkan Data Sensor Ketinggian Berbasis Mikrokontroller Arduino(<i>Prototype</i>)	<ul style="list-style-type: none"> • Membangun sebuah prototipe sistem pemantauan ketinggian air menggunakan mikrokontroler Arduino dan aplikasi berbasis web. Mikrokontroler Arduino berfungsi untuk mengambil data ketinggian air • Target penggunaan aplikasi hanya digunakan oleh admin • Aplikasi berbasis Web • Aplikasi ini menampilkan daerah rawan banjir di Pontianak

2.2 Visualisasi

Secara umum visualisasi merupakan rekayasa gambar, *Diagram*, animasi dalam menampilkan suatu informasi. Menurut (Kard, dkk 1998) visualisasi secara merupakan penggunaan media komputer sebagai alat pendukung, dalam melakukan penggambaran visual interaktif agar dapat memperkuat pengamatan informasi yang

dilakukan. Sedangkan menurut Mc Cormick visualisasi merupakan suatu metode komputasi. Metode tersebut dapat mengubah simbolis menjadi geometris. Berdasarkan defenisi diatas bahwa visualisasi merupakan metode yang memanfaatkan teknologi komputer untuk mengungkapkan suatu gagasan suatu informasi berupa gambar, tulisan, peta, grafik, dan lainnya yang interaktif yang dapat mempermudah dalam mengembangkan pemahaman yang lebih dalam. Berikut ini merupakan gambar dari visualisasi

Visualisasi



Gambar 2. 1 Visualisasi

2.3 Spasial

Spasial dapat diartikan sebagai sesuatu yang berkaitan dengan ruang atau tempat. Dalam geografi ada istilah data spasial yang dikenal juga sebagai data geospasial atau informasi letak geografis, itu adalah data atau informasi yang mengidentifikasi lokasi geografis kenampakan dan batas di Bumi, seperti kenampakan alami atau buatan, perairan atau laut, dan banyak lagi.

Data spasial biasanya disimpan sebagai koordinat dan topologi, dan merupakan data yang dapat dipetakan. Data spasial sering diakses oleh banyak orang, dimanipulasi atau dianalisis melalui komponen Sistem Informasi Geografis (SIG).

2.4 NODE MCU V3

NodeMCU pada dasarnya adalah pengembangan dari ESP 8266 dengan firmware berbasis e-Lua. Pada NodeMcu dilengkapi dengan micro usb port yang berfungsi untuk pemrograman maupun power supply. Selain itu juga pada NodeMCU di lengkapi dengan tombol push button yaitu tombol reset dan flash. NodeMCU menggunakan bahasa pemrograman Lua yang merupakan package dari esp8266. Bahasa Lua memiliki logika dan susunan pemrograman yang sama dengan c hanya berbeda syntax. Jika menggunakan bahasa Lua maka dapat menggunakan tool Lua loader maupun Lua uploder.

Selain dengan bahasa Lua NodeMCU juga support dengan software Arduino IDE dengan melakukan sedikit perubahan board manager pada Arduino IDE. Sebelum digunakan Board ini harus di Flash terlebih dahulu agar support terhadap tool yang akan digunakan. Jika menggunakan Arduino IDE menggunakan firmware yang cocok yaitu firmware keluaran dari Ai- Thinker yang support AT Command. Untuk penggunaan tool loader Firmware yang di gunakan adalah firmware NodeMCU dari sumber (Kurnia Dewi, Suti. 2018).



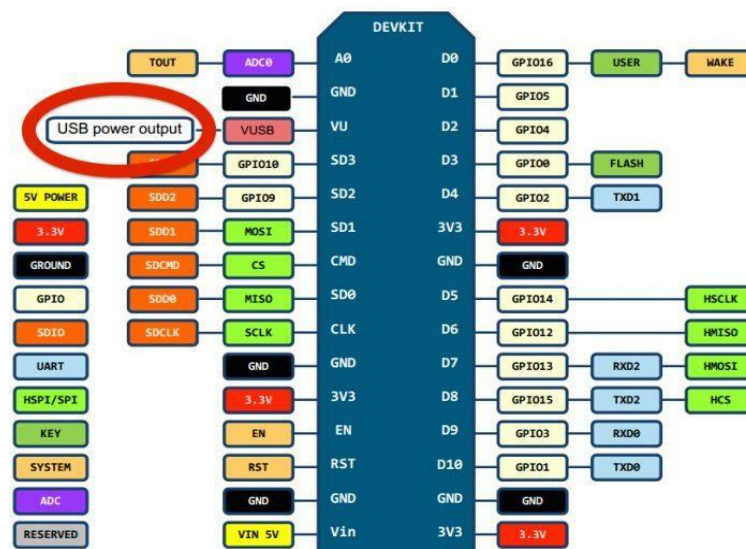
Gambar 2. 2 ESP 8266 NODEMCU V322

Adapun untuk spesifikasi dari NodeMCU V3 dapat dilihat pada table 5.1.

Tabel 2. 3 Spesifikasi NODEMCU V3

SPESIFIKASI	NODEMCU V3
-------------	------------

Mikrokontroler	ESP8266
Ukuran Board	57 mmx 30 mm
Tegangan Input	3.3 ~ 5V
GPIO	13 PIN
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
Flash Memory	4 MB
Clock Speed	40/26/24 MHz
WiFi	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz – 22.5 Ghz
USB Port	Micro USB
Card Reader	Tidak Ada
USB to Serial Converter	CH340G



Gambar 2. 3. Skematik posisi Pin NodeMcu Dev Kit v3

2.5 Mikrokontroler

Mikrokontroler adalah sebuah chip yang di dalamnya terdapat mikroprosesor yang telah di kombinasikan I/O dan memori RAM/ROM (Prayudha et al., 2014). Menurut Iswanto dalam Kurnia Dewi (2018:9) Mikrokontroler adalah suatu rangkaian terintegrasi (IC) yang berkerja untuk aplikasi pengendalian. Meskipun mempunyai bentuk lebih kecil dari computer pribadi dan mainframe, mikrokontroler dibangun dengan elemen-elemen yang sama. Mikrokontroler adalah alat yang mengerjakan instruksi-instruksi yang diberikan, artinya bagian utama dari sistem otomatis /terkomputerisasi adalah program yang di dalamnya dibuat oleh programmer. Program menginstruksikan mikrokontroler untuk melakukan jalinan yang panjang dari aksi-aksi sederhana untuk melakukan tugas yang lebih kompleks sesuai keinginan programmer.

Dari beberapa definisi yang disebutkan, dapat disimpulkan bahwa mikrokontroler adalah sebuah sistem fungsional dalam sebuah chip yang mempunyai prosesor, memori dan perlengkapan input dan output yang menjadi kendali dari sebuah program yang ditulis. Umumnya mikrokontroler terdiri dari:

1. RAM (Random Access Memory)

RAM adalah memori yang dapat dibaca atau ditulis. Data pada RAM bersifat volatile, artinya data akan terhapus jika catu daya dimatikan. RAM digunakan untuk menyimpan data sementara, yaitu data yang tidak begitu vital bila hilang akibat aliran daya terputus,

2. ROM (Read Only Memory)

ROM merupakan memori yang hanya dapat dibaca. Data dalam ROM tidak akan terhapus meskipun catu daya diputuskan (bersifat nonvolatile) karena sifatnya yang demikian, ROM digunakan untuk menyimpan program.

3. Register

Register adalah tempat penyimpanan nilai-nilai yang akan digunakan dalam proses, telah disediakan oleh mikrokontroler.

4. SFR (Special Function Register)

SFR adalah register khusus yang berfungsi mengatur jalannya

mikrokontroler dan terletak pada RAM.

5. Input dan Output Pin

Pin input berfungsi sebagai penerima sinyal dari luar, pin ini dapat dihubungkan ke media inputan seperti keypad, sensor dan lain sebagainya. Pin output adalah bagian yang berfungsi untuk mengeluarkan sinyal dari hasil proses algoritma mikrokontroler.

6. Interrupt

Interrupt bagian dari mikrokontroler yang berfungsi sebagai bagian yang dapat melakukan interupsi, sehingga ketika program utama sedang berjalan, program tersebut dapat diinterupsi (melompat ke program interrupt service routine)

2.5.1. Jenis-jenis Mikrokontroler

Menurut Anna (2010), “Secara teknis, hanya ada 2 macam mikrokontroler. Pembagian ini didasarkan pada kompleksitas instruksi-instruksi yang dapat diterapkan pada mikrokontroler tersebut. Pembagian itu yaitu RISC dan CISC sebagaimana dijelaskan di bawah ini,

1. RISC merupakan kependekan dari Reduced Instruction Set Computer. Instruksi yang dimiliki terbatas, tetapi memiliki fasilitas yang lebih banyak. Contoh RISC yaitu Mikrokontroler AVR, PIC (Peripheral Interface Controller, Mikrokontroler ARM.
2. CISC kependekan dari Complex Instruction Set Computer. Instruksi bisa dikatakan lebih lengkap tapi dengan fasilitas secukupnya. Contoh CISC yaitu Mikrokontroler MCS-51.”

2.6 Modul Arduino

Menurut Yenni, dkk (2016), “Arduino adalah platform dari physical computing yang bersifat open source. Pengertian physical computing adalah suatu sistem atau perangkat fisik yang dibangun dengan menggunakan perangkat keras dan perangkat lunak yang bersifat interaktif, yaitu dapat menerima rangsangan dari lingkungan dan memberikan respon balik.” Menurut Silvia, dkk (2014), “Arduino adalah sebuah rangkaian yang dikembangkan dari mikrokontroler

berbasis Atmega328. Arduino memiliki 14 kaki digital input/output, di mana 6 kaki digital di antaranya dapat digunakan sebagai sinyal PWM (Pulse Width Modulation) sinyal PWM berfungsi untuk mengatur kecepatan perputaran motor."

Berbagai jenis kartu Arduino tersedia, antara lain Arduino Uno, Arduino Diecimila, Arduino Duemilanove, Arduino Leonardo, Arduino Mega, dan Arduino Nano. Walaupun ada berbagai jenis kartu Arduino, secara prinsip pemrograman yang diperlukan menyerupai. Hal yang membedakan adalah kelengkapan fasilitas dan pin-pin yang perlu digunakan. Pada penelitian ini modul Arduino yang digunakan adalah Arduino Nano.

2.6.1. Perangkat Lunak Pemrograman Arduino

Perangkat lunak Arduino yang akan digunakan adalah driver dan IDE. IDE Arduino adalah perangkat yang sangat canggih ditulis dengan menggunakan Java.

IDE Arduino terdiri dari :

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa Processing.
2. Compiler, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa Processing. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.
3. Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam memory di dalam papan Arduino. Sebuah kode program dalam Arduino umumnya disebut dengan istilah sketch.

Tampilan awal perangkat lunak pemrograman Arduino dapat sebagaimana dapat dilihat pada Gambar 2.2 berikut.



Gambar 2. 4 Tampilan awal Software arduino de

Gambar 2.2 merupakan tampilan utama sketch arduino untuk memulai pemrograman pada arduino board dengan penjelasan sebagai berikut:

1. *Verify* berfungsi untuk mengecek program yang telah dibuat atau lebih dikenal dengan istilah compile.
 2. *Upload* berfungsi untuk memasukan atau menulis program yang telah ditulis ke dalam mikrokontroller.
 3. *New* berfungsi untuk membuat file yang telah disimpan sebelumnya.
 4. *Open* berfungsi untuk membuka file yang telah disimpan sebelumnya.
 5. *Save* berfungsi untuk menyimpan sketch yang telah dibuat.
 6. Bagian ini digunakan untuk inisialisasi program seperti menentukan port *input* maupun *output*.
 7. Bagian ini merupakan tempat program utama dibuat dan program akan
 8. dieksekusi secara terus-menerus sampai didapat output yang diinginkan.
- Pada bagian ini menunjukkan mikrokontroler yang digunakan dan port serial yang digunakan.



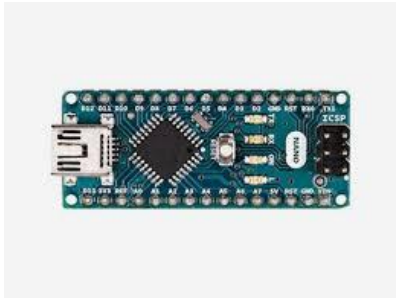
Gambar 2. 5 Tampilan Utama Sketch Arduino

Sumber: (Arduino. 2018. *Arduino Nano*. <https://store.arduino.cc/usa/arduino-nano>.)

2.6.2. Papan (Board) Arduino Nano

ESP 2866 merupakan suatu sistem minimum lengkap dengan ukuran yang relatif kecil berbasis mikrokontroler ATmega328 untuk ESP 2866.3x dan ATmega168 untuk ESP 28662.x. ESP 2866 memiliki fungsi yang sama dengan arduino duemilanove, tetapi dalam bentuk yang berbeda. ESP 2866 didesain dan diproduksi oleh Gravitech. Input dan Output, ESP 2866 memiliki jumlah pin sebanyak 14 pin yang dapat digunakan untuk input dan output dan memiliki beberapa fungsi pinMode(), digitalWrite(), digitalRead(). Pin ESP 2866 bekerja pada tegangan 5V dengan Arus 20mA serta memiliki tahanan pull-up 20 - 50k ohm. ESP 2866 memiliki beberapa fasilitas untuk komunikasi dengan komputer, Arduino lainnya maupun microcontroller lainnya. Atmega 328 menyediakan serial komunikasi UART TTL (5V), yang tersedia pada pin digital 0 (RX) dan 1 (TX). Board system minimum ESP 2866 dapat dilihat pada Gambar 2.5.

Gambar 2.1 Board System Minimum Arduino Nano.



(a). Tampak Depan



(b). Tampak Samping.

Gambar 2. 6 Board System Minimum Arduino Nano

Sumber: (<https://store.arduino.cc>)

ESP 2866 memiliki spesifikasi sebagai berikut:

Tabel 2. 4 Spesifikasi Arduino Nano

No	Spesifikasi	Keterangan
1.	Mikrokontroler	ATmega328
2.	Arsitektur	AVR
3.	Operasi Tangan	5 Volt
4.	SRAM	2 KB
5.	Kecepatan clock	16 MHz
6.	Memory Flash	32 KB dan 2 KB digunakan oleh <i>bootloader</i>
7.	Pin Analog	8
8.	EEPROM	1 KB
9.	Arus DC tiap pin I/O	40 mA (pinI/O)
10.	Input Tegangan	7-12 Volt
11.	Pin I/O	22 (6 bisa untuk PWM)
12.	Pin Output PWM	6
13.	Size PCB	18 x 45 mm
14.	Berat	7 g

Sumber (<https://store.arduino.cc>)

ESP 2866 dapat diberi tenaga dengan power yang diperoleh dari koneksi kabel Mini-B USB atau via Supply Power External yang dapat langsung dihubungkan ke board ESP 2866 melalui pin 30 atau Vin (unregulated 6V - 20V) dan bisa juga menggunakan pin 27 dengan tegangan 5V. Memori ESP 2866 pada Chip ATmega 328 memiliki memori 32 KB dengan 2 KB telah digunakan untuk bootloader. ATmega 328P memiliki jumlah SRAM 2 KB dan EEPROM 1 KB (yang dapat dibaca dan ditulis menggunakan EEPROM library). Setiap port digital dan analog dalam Atmega328 dapat digunakan sebagai input dan output dengan menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. ESP 2866 memiliki 6 pin analog dengan resolusi 10 bit dan 14 pin digital

2.7 Modul Sensor HC-SR04

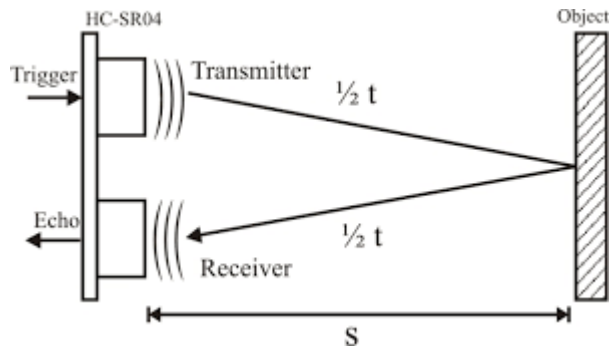
HC-SR04 merupakan sensor ultrasonik yang dapat digunakan untuk mengukur jarak antara penghalang dan sensor. Konfigurasi pin dan tampilan sensor HC-SR04 diperlihatkan pada Gambar 2.5.



Gambar 2. 7 Konfigurasi Pin dan Tampilan Sensor HC-SR04

Sumber (<http://kelas-fisika.com>)

HC-SR04 memiliki 2 komponen utama sebagai penyusunnya yaitu ultrasonic transmitter dan ultrasonic receiver. Fungsi dari ultrasonic transmitter adalah memancarkan gelombang ultrasonik dengan frekuensi 40 KHz kemudian ultrasonic receiver menangkap hasil pantulan gelombang ultrasonik yang mengenai suatu objek. Waktu tempuh gelombang ultrasonik dari pemancar hingga sampai ke penerima sebanding dengan 2 kali jarak antara sensor dan bidang pantul seperti yang diperlihatkan pada Gambar 2.6.



Gambar 2. 8 Prinsip kerja HC-SR04

Sumber (<https://widuri.raharja.info>)

Prinsip pengukuran jarak menggunakan sensor ultrasonik HC-SR04 adalah ketika pulsa trigger diberikan pada sensor, transmitter akan mulai memancarkan gelombang ultrasonik, pada saat yang sama sensor akan menghasilkan output TTL transisi naik menandakan sensor mulai menghitung waktu pengukuran, setelah receiver menerima pantulan yang dihasilkan oleh suatu objek maka pengukuran waktu akan dihentikan dengan menghasilkan output TTL transisi turun. Jika waktu pengukuran adalah t dan kecepatan suara adalah 340 m/s , maka jarak antar sensor dengan objek dapat dihitung dengan menggunakan Persamaan 2.1.

$$S = t \times \frac{340 \text{ M/s}}{2}$$

(2.1)

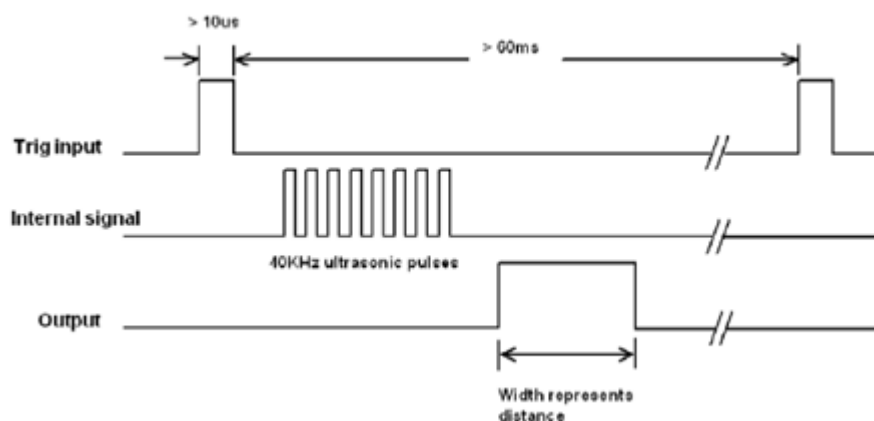
Dimana :

s = Jarak antara sensor dengan objek (m)

t = Waktu tempuh gelombang ultrasonik dari transmitter ke receiver (s)

Pemilihan HC-SR04 sebagai sensor jarak yang akan digunakan pada penelitian ini karena memiliki fitur sebagai berikut kinerja yang stabil, pengukuran jarak yang akurat dengan ketelitian $0,3 \text{ cm}$. Pengukuran maksimum dapat mencapai 4 meter dengan jarak minimum 2 cm , ukuran yang ringkas dan dapat beroperasi pada level tegangan TTL. Prinsip pengoperasian sensor ultrasonik HC-SR04 adalah diawali dengan memberikan pulsa Low (0) ketika modul mulai dioperasikan, kemudian diberikan pulsa High (1) pada trigger selama 10 us sehingga modul mulai memancarkan 8 gelombang kotak dengan frekuensi 40 KHz , tunggu hingga transisi naik terjadi pada output dan mulai perhitungan waktu hingga transisi turun terjadi, setelah itu gunakan Persamaan 5.1 untuk mengukur jarak antara sensor dengan

objek. *Timing Diagram* pengoperasian sensor ultrasonik HC-SR04 diperlihatkan pada Gambar 2.7.



Gambar 2. 9 *Timing Diagram* Pengoperasian Sensor HC-SR04

Sumber (<http://kelas-fisika.com>)

2.8 Website

Website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada website disebut dengan *web page* dan link dalam website memungkinkan pengguna bisa berpindah dari satu page ke page lain (*hyper text*), baik di antara page yang disimpan dalam server yang sama maupun server diseluruh dunia. Pages diakses dan dibaca melalui browser seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* dan aplikasi browser lainnya (Hakim 2014).

Website (situs web) merupakan alamat (URL) yang berfungsi sebagai tempat penyimpanan data dan informasi dengan berdasarkan topik tertentu. URL adalah suatu sarana yang digunakan untuk menentukan lokasi informasi pada suatu Web. Situs atau Web dapat dikategorikan menjadi 2 (dua) yaitu:

1. Web Statis, yaitu web yang berisi atau menampilkan informasi-informasi yang sifatnya statis (tetap).
2. Web Dinamis, yaitu web yang menampilkan informasi serta dapat berinteraksi dengan user yang sifatnya dinamis.

2.9 HTML

HTML (*Hypertext Markup Language*) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah *web Internet* dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Sebuah bahasa markah untuk membuat halaman web dan bahasa yang digunakannya masih sangat standar seperti salah satu fungsinya untuk membuat tabel, menambahkan objek suara, video dan animasi adalah pengertian dari HTML (Hidayatullah, Priyanto and Kawistara 2017). Pengertian di atas dapat disimpulkan bahwa HTML adalah sebuah dokumen yang berisikan *tag*, beberapa elemen dan atribut untuk menampilkan halaman pada *web browser*.



```
index.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Page Title</title>
5  </head>
6  <body>
7
8  <h1>This is a Heading</h1>
9  <p>This is a paragraph.</p>
10
11 </body>
12 </html>
```

Gambar 2. 10 Tag HTML.

2.10 Framework dalam pengembangan Website

Framework yang digunakan dalam pengembangan sistem informasi manajemen kerja praktek Jurusan Informatika adalah *framework CSS Bootstrap* untuk *interface* dan *framework PHP Codeigniter* untuk manajemen data. Kedua *framework* tersebut dipilih karena kelebihan yang dimilikinya. Berikut ini penjelasan dari *framework* yang digunakan:

1. *Framework* CSS Bootstrap

Bootstrap menyediakan kumpulan komponen-komponen *interface* dasar pada *website* yang bersifat *open-source* yang dibuat untuk mempermudah pengembang aplikasi *website* dalam membangun suatu antarmuka *website*. Tujuan pembuatan bootstrap adalah untuk menyediakan komponen/ *library* yang dapat diolah sesuai kebutuhan, terdokumentasi dengan baik, dan fleksibel atau mudah digunakan bersama dengan bahasa pemrograman HTML, CSS, dan Javascript.

Kelebihan dari *framework* CSS bootstrap yaitu *Preprocessors*, *One Framework every device* dan *comprehensive docs* (Official Website Bootstrap, 2020). *Framework* CSS Bootstrap mendukung dua *preprocessors* CSS yang paling populer yaitu *Less* dan *Sass*. *Preprocessors* CSS mempermudah pengembang aplikasi dalam mengembangkan aplikasi dan efisien serta optimal dalam menuliskan kode yaitu hanya dengan menulis beberapa baris kode saja. *One Framework every device*, artinya bootstrap dapat digunakan untuk berbagai macam perangkat karena bootstrap bersifat *responsive*. *Comprehensive docs* artinya bootstrap memiliki dokumentasi dan tutorial yang banyak baik dari segi kode maupun template-template *website*.



```
index.css
1  body {
2    background-color: lightblue;
3  }
4
5  h1 {
6    color: white;
7    text-align: center;
8  }
9
10 p {
11   font-family: verdana;
12   font-size: 20px;
13 }
```

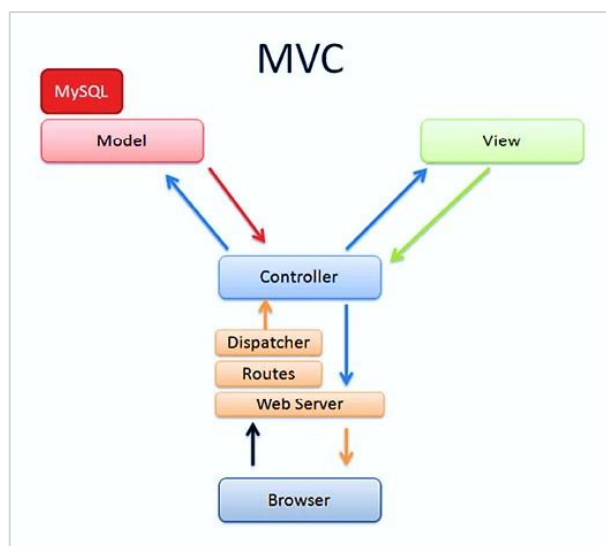
Gambar 2. 11 Tag CSS.

2. Framework Codeigniter

Framework CodeIgniter merupakan salah satu framework PHP yang digunakan untuk mempermudah pembuatan sebuah website yang bersifat dinamis dengan

menggunakan konsep MVC (*Model, View, Controller*). Framework CodeIgniter bersifat *open source* (Ellislab. 2020). Framework CodeIgniter telah menyediakan beberapa macam library sehingga dapat mempermudah pembuatan website. Framework CodeIgniter memiliki beberapa kelebihan, seperti: mudah dipelajari, mudah dimodifikasi, mempunyai dokumentasi yang lengkap, dan menggunakan konsep MVC sehingga struktur kode menjadi lebih terstruktur.

Codeigniter Bekerja berdasarkan konsep dasar MVC. MVC (*Model View Controller*) merupakan sebuah *pattern/* teknik pemrograman yang memisahkan *bisnis logic* (alur pikir), *data logic* (penyimpanan data), dan *presentation logic* (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data, dan proses. Gambar dibawah ini menunjukkan pola MVC terdiri dari 3 bagian, yaitu:



Gambar 2. 12 Pola MVC Codeigniter.

- 1) *Model* yang bertugas dalam pengolahan data. *Model* ini berhubungan dengan data dan interaksi ke database atau *webservice*, mempresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain.



```

1 <?php
2
3 class Model_jadwal extends CI_Model
4 {
5     public function getTAll()
6     {
7         $this->db->from('tbl_pelaksanaan');
8         $this->db->limit(1);
9         $this->db->order_by('Id_pelaksanaan', 'DESC');
10        $query = $this->db->get();
11        return $query->result();
12    }
13
14    public function insertJadwal($Pelaksanaan, $Tanggal_mulai, $Tanggal_selesai, $Pengkajuan_seminar, $Pelaksanaan_seminar, $Revisi, $Tanggal)
15    {
16        $data = array (
17            'Pelaksanaan'      => $Pelaksanaan,
18            'Tanggal_mulai'    => $Tanggal_mulai,
19            'Tanggal_selesai'  => $Tanggal_selesai,
20            'Pengkajuan_seminar' => $Pengkajuan_seminar,
21            'Pelaksanaan_seminar' => $Pelaksanaan_seminar,
22            'Revisi'           => $Revisi,
23            'Tanggal'         => $Tanggal
24        );
25
26        $this->db->insert('tbl_pelaksanaan', $data);
27    }
28
29 }

```

Gambar 2. 13 Tag Model.

- 2) *View* bertugas dalam menampilkan data ke *end-user*. *View* bisa berupa halaman web, html, css, javascript dan lain-lain. Di dalam *view* hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. *View* dapat dikatakan sebagai halaman *website* yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript.
- 3) *Controller* bertindak sebagai penghubung antara data/ model dan view. Tugas controller adalah menyediakan berbagai variabel yang akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap *input*.



```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Welcome extends CI_Controller {
5
6     /**
7      * Index Page for this controller.
8      *
9      * Maps to the following URL
10     * http://example.com/index.php/welcome
11     * - or -
12     * http://example.com/index.php/welcome/index
13     * - or -
14     * Since this controller is set as the default controller in
15     * config/routes.php, it's displayed at http://example.com/
16     *
17     * So any other public methods not prefixed with an underscore will
18     * map to /index.php/welcome/<method_name>
19     * @see https://codeigniter.com/user_guide/general/urls.html
20     */
21     public function index()
22     {
23         $this->load->view('welcome_message');
24     }
25 }
26

```

Gambar 2. 14 Tag Controller.

Codeigniter dibuat pertama kali oleh Rick Ellis yang merupakan CEO dari Ellislab ini memiliki beberapa kelebihan yaitu: memiliki dokumentasi yang lengkap dan bagus, kompatibel dengan semua jenis hosting, memiliki performa yang bagus, memiliki fleksibilitas dalam aturan penulisan kode, mudah untuk dipelajari dan digunakan (Ellislab. 2020). *Framework Codeigniter* juga memiliki hasil *benchmark* yang tinggi dan cepat dibandingkan *framework php* yang lain (zend, laravel, kohana, fuel, yii, symphony). Berdasarkan kelebihan yang dimiliki pada *framework codeigniter*, oleh karena itu penelitian ini menggunakan *framework codeigniter* untuk implementasi pengembangan sistem informasi manajemen kerja praktek.

2.11 Web Service

Web service dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah *database* ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan di *platform* apa sebuah data itu dikonsumsi. *Web service* mampu menunjang interoperabilitas. Sehingga *web service* mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada.

Menurut W3C *Web services Architecture Working Group* pengertian *Web service* adalah sebuah sistem *Software* yang di desain untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan. *Interface web service* dideskripsikan dengan menggunakan format yang mampu diproses oleh mesin (khususnya WSDL). Sistem lain yang akan berinteraksi dengan *web service* hanya memerlukan SOAP, yang biasanya disampaikan dengan HTTP dan XML sehingga mempunyai korelasi dengan standar Web (*Web Services Architecture Working Group*, 2021).

Web pada umumnya digunakan untuk melakukan respon dan request yang dilakukan antara client dan server. Sebagai contoh, seorang pengguna layanan web tertentu mengetikkan alamat URL web untuk membentuk sebuah request. Request akan sampai pada server, diolah dan kemudian disajikan dalam bentuk sebuah respon. Dengan singkat kata terjadilah hubungan *client-server* secara sederhana.

Sedangkan pada *web service* hubungan antara client dan server tidak terjadi secara langsung. Hubungan antarclient dan server dijembatani oleh file *web service* dalam format tertentu. Sehingga akses terhadap database akan ditangani tidak secara langsung oleh server, melainkan melalui perantara yang disebut sebagai *web service*. Peran dari *web service* ini akan mempermudah distribusi sekaligus integrasi database yang tersebar di beberapa server sekaligus.

2.12 Javascript Object Notation

Menurut Deitel (2012, p1303) JSON (*JavaScript Object Notation*) adalah suatu format pertukaran data komputer. Format dari JSON adalah berbasis teks, dapat terbaca oleh manusia, digunakan untuk mempresentasikan struktur data sederhana, dan tidak bergantung dengan bahasa apapun. Biasanya digunakan pada aplikasi *Ajax*. Format JSON sering digunakan untuk mentransmisikan data terstruktur melalui koneksi jaringan. Secara umum, JSON digunakan untuk mentransmisikan data antara server dan aplikasi web.

Jenis media internet yang resmi untuk JSON adalah aplikasi/json. Format JSON sering digunakan untuk serialisasi dan mengirimkan data terstruktur melalui koneksi jaringan, terutama untuk pengiriman data antara server dan aplikasi web melayani sebagai alternatif ke XML.

A screenshot of a code editor window titled 'mahasiswa.json'. The editor shows a JSON object with the following structure:

```
1 {  
2   "mahasiswa":  
3     {  
4       "name": "MULidan",  
5       "NIM": "D1041151007",  
6       "Jurusan": "Informatika",  
7       "Fakultas": "Teknik"  
8     }  
9 }
```

The code is color-coded: curly braces are blue, strings are orange, and the key 'mahasiswa' is in red. Line numbers 1 through 9 are visible on the left side of the editor.

Gambar 2. 15 Tag JSON.

Berikut ini adalah

a. Leaflet

Leaflet merupakan library javascript open source yang berguna untuk membangun aplikasi peta interaktif berbasis web. Leaflet support dengan platform mobile dan platform desktop, HTML5 dan CSS3 serta OpenLayer dan Google Maps API yang merupakan library javascript untuk membangun aplikasi peta yang sangat populer saat ini.

Dengan memanfaatkan leaflet, developer yang tidak memiliki latar belakang GIS pun dapat dengan mudah menampilkan peta interaktif berbasis web pada server. Leaflet mampu menampilkan layer dari file geojson, memberi style dan membuat layer yang interaktif seperti menampilkan marker yang menampilkan popup informasi ketika di klik.

b. Open Street Maps(OSM)

OpenStreetMap (OSM) adalah sebuah proyek berbasis web untuk membuat peta seluruh dunia yang gratis dan terbuka, dibangun sepenuhnya oleh sukarelawan dengan melakukan survey menggunakan GPS, mendigitasi citra satelit, dan mengumpulkan serta membebaskan data geografis yang tersedia di publik.

Melalui Open Data Commons Open Database License 1.0, kontributor OSM dapat memiliki, memodifikasi, dan membagikan data peta secara luas. Terdapat beragam jenis peta digital yang tersedia di internet, namun sebagian besar memiliki keterbatasan secara legal maupun teknis. Hal ini membuat masyarakat, pemerintah, peneliti dan akademisi, inovator, dan banyak pihak lainnya tidak dapat menggunakan data yang tersedia di dalam peta tersebut secara bebas. Di sisi lain, baik peta dasar OSM maupun data yang tersedia di dalamnya dapat diunduh secara gratis dan terbuka, untuk kemudian digunakan dan didistribusikan kembali.

Di banyak tempat di dunia ini, terutama di daerah terpencil dan terbelakang secara ekonomi, tidak terdapat insentif komersil sama sekali bagi perusahaan pemetaan untuk mengembangkan data di tempat ini. OSM dapat menjadi jawaban di banyak tempat seperti ini, baik itu untuk pengembangan ekonomi, tata kota, kontinjensi bencana, maupun untuk berbagai tujuan lainnya.

2.13 *Unified Modelling Language*

Unified Modelling Language (UML) merupakan salah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *Software* yang terkait dengan objek. UML sendiri berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi *Diagram*. UML mempunyai banyak *Diagram* yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. Menurut Nugroho Adi (2010), dalam merancang UML terdapat beberapa *Diagram* yang dibuat sebagai berikut:



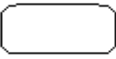

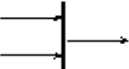

a. *Use Case Diagram*

Use Case Diagram merupakan pemodelan yang menggambarkan kelakuan sistem secara keseluruhan. *Diagram Use Case Diagram* mendeskripsikan interaksi antar satu lebih aktor dengan sistem yang akan dibuat. Secara garis besar, *Diagram Use Case Diagram* digunakan untuk mengetahui fungsi apa saja yang ada dan siapa saja yang berhak menjalankan fungsi tersebut pada sistem.

Dalam konteks UML, tahap konseptualisasi dilakukan dengan pembuatan *Use Case Diagram* yang sesungguhnya merupakan deskripsi peringkat tertinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunanya. Selanjutnya, *Use Case Diagram* tidak hanya sangat penting pada tahap analisis, tetapi juga sangat penting untuk perancangan, untuk menemukan kelas-kelas yang terlibat dalam aplikasi, dan untuk melakukan pengujian.

b. *Activity Diagram*

Activity Diagram (Diagram Aktivitas) merupakan *Diagram* alir kerja atau aktivitas dari sebuah sistem atau proses bisnis. *Diagram* aktivitas merupakan penjabaran dari *Use Case Diagram* *Diagram* yang menggambarkan aktivitas sistem. *Diagram* aktivitas memiliki struktur *Diagram* yang mirip *flowchart* atau data *flow Diagram* pada perancangan terstruktur. Memiliki pula manfaat yaitu apabila kita membuat *Diagram* ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan.

Simbol	Keterangan
 	Start Point End Point
	Activities
	Fork (Percabangan)
	Join (Penggabungan)
 Swimlane	Decision Sebuah cara untuk mengelompokkan activity berdasarkan Actor (mengelompokkan activity dalam sebuah urutan yang sama)

Gambar 2. 16 Keterangan activity Diagram.





c. *Class Diagram*

Class Diagram adalah *Diagram* yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang kita kembangkan dimana *Diagram* ini memberi kita gambaran (*Diagram* statis) tentang sistem/perangkat lunak dan relasi-relasi yang ada di dalamnya.

- **Atribut:** merupakan properti dari sebuah kelas yang melambangkan nilai-nilai yang mungkin ada pada kelas tersebut.
- **Operation atau Method :** merupakan *behavior* (tingkah laku) atau fungsi yang dapat dilakukan oleh kelas tersebut.

Dalam *Class Diagram* dikenal beberapa istilah salah satu diantaranya yakni Panah *Navigability* (pengatur alur arah). Panah *navigability* dalam suatu proses *association* menggambarkan arah mana *association* dapat ditransfer atau disusun.

Tabel 2. 5 Relasi Pada Kelas Diagram.

asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas.
agregasi 	Relasi antar kelas dengan makna semua bagian.

Selain panah *navigability* dikenal pula istilah *Multiplicity* yakni merupakan angka kemungkinan bagian dari hubungan kelas dengan single instance (bagian) pada titik yang lain. *Multiplicity* berupa *single number* (angka tunggal) atau *range number* (angka batasan).

Tabel 2. 6 Multiplicity

<i>Multiplicities</i>	Keterangan
0..1	Nol atau satu bagian. Notasi $n \dots m$ menerangkan n sampai m bagian
0..* or *	Tak hingga pada jangkauan bagian (termasuk kosong)
1	Tepat Satu Bagian
1..*	Sedikitnya hanya satu bagian

2.14 Basis Data

Secara teori basis data adalah suatu sistem yang memproses *input* berupa data menjadi *output* yaitu informasi yang diinginkan. Sedangkan menurut fungsinya adalah himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah. Untuk memperoleh basis data yang handal perlu diperhatikan hal-hal seperti keamanan data, dan kemungkinan untuk pengembangan basis data tersebut.

Basis Data atau *Database* sering dianggap sebagai salah satu sistem komponen yang harus ditunjang oleh perangkat lunak dan perangkat keras. Perangkat lunak bisa berupa aturan-aturan, tata cara, program atau aplikasi (*Software* khusus), ataupun pengolahan data. Sedangkan perangkat keras dapat berupa kertas, buku, seperangkat komputer, atau lemari arsip. Salah satu kegunaan *database* adalah membuat model konseptual. Pembuatan model konseptual adalah pembuatan model berdasarkan sistem informasi dari user, tanpa tergantung pada *database management system* (DBMS) yang akan dipakai.

Secara lebih lengkap, pemanfaatan *database* dilakukan untuk memenuhi sejumlah tujuan (objektif) seperti berikut:

a. Kecepatan dan Kemudahan (*Speed*)

Pemanfaatan *database* memungkinkan untuk dapat menyimpan atau melakukan manipulasi terhadap data atau menampilkan data tersebut dengan cepat dan mudah.

b. Efisiensi Ruang Penyimpanan (*Space*)

Karena adanya keterkaitan antar data dalam sebuah *database*, maka redundansi data pasti akan terjadi. Banyaknya redundansi yang ada tentu saja akan memakan banyak tempat (memori). Dengan *database* efisiensi penggunaan ruang memori dapat dilakukan dengan cara membuat pengkodean atau membuat relasi-relasi antar kelompok data yang saling berhubungan.

c. Keakurasian Data (*Accuracy*)

Pemanfaatan pengkodean data atau pembentukan relasi antar data dengan penerapan aturan atau batasan tipe data, domain data, keunikan data, dan sebagainya yang secara ketat dapat diterapkan dalam sebuah *database*, akan sangat

berguna untuk menekan ketidakakuratan pemasukan atau penyimpanan data.

d. Ketersediaan Data (*Availability*)

Pertumbuhan data sejalan dengan berjalannya waktu akan semakin membutuhkan ruang penyimpanan yang semakin besar. Untuk itu, data-data yang tidak dibutuhkan lagi dapat dihapus atau dipindahkan ketempat lain, selain itu data juga dapat kita sebarakan keberbagai tempat yang terhubung dengan sistem.

e. Kelengkapan Data (*Completeness*)

Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita dapat menambah *record* data, selain itu kita juga dapat melakukan perubahan struktur dalam *database*.

f. Keamanan Data (*Security*)

Kita dapat menentukan user mana saja yang boleh mengakses database beserta objek-objek didalamnya, dan menentukan jenis operasi apa saja yang bisa diakses.

g. Kebersamaan Pemakai (*Sharability*)

Karena data yang ada dalam suatu *database* bisa disebar keberbagai tempat yang terhubung dengan sistem, maka setiap user bisa mengakses data secara bersamaan dari tempat yang terpisah.

2.15 Prototype

Metode *prototype* merupakan teknik Pengembangan sistem dengan penggambaran *prototype* sehingga pemilik sistem memiliki gambaran jelas tentang sistem yang akan dibangun oleh para tim pengembang atau developer. Metode pengembangan perangkat lunak jenis *prototype* biasanya digunakan jika para pengembang mengalami kesulitan memahami apa yang dibutuhkan oleh para klien, dalam hal ini klien juga biasanya kurang bisa memahami dan menerjemahkan sistem yang mereka inginkan.

2.16 BlackBox Testing

Pengujian *black box* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan cek fungsional perangkat lunak tanpa memperhatikan struktur logika. Tujuan pengujian *black box* adalah untuk mencari

kesalahan/kegagalan dalam operasi tingkat tinggi, yang mencakup kemampuan dari perangkat lunak, operasional/tata laksana, skenario pemakai. Fungsi dari pengujian ini berdasarkan kepada apa yang dapat dilakukan oleh sistem.

Pengujian *black box* menurut Roger S. Pressman. (2002:551), memaparkan tentang pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perekrut perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian dengan metode *Black-Box* berusaha menemukan kesalahan dalam beberapa kategori, yaitu :

- a. Pengujian penginstalan aplikasi pada perangkat.
- b. Pengujian fungsi-fungsi yang hilang atau tidak benar.
- c. Kesalahan aplikasi dalam menyampaikan informasi.

Proses pengujian dapat dilakukan dengan cara manual ataupun otomatis. Namun pengujian secara manual memiliki banyak kelemahan diantaranya cakupan pengujian yang dilakukan tidak cukup luas, karena penguji cenderung melakukan hal yang sama berulang-ulang, sehingga antarmuka yang lain tidak diuji. Kelemahan lain yaitu, sulit mereproduksi kegagalan karena urutan perintah pengujian tidak dicatat, dan pengujian manual tidak melakukan perekam *response time* secara otomatis (A.S 2014)

Pengujian secara otomatis (automatic testing) dapat mengeksekusi setiap proses pengujian oleh perangkat lunak (*tools*). Terdapat beberapa keuntungan automatic testing pada GUI, yaitu memungkinkan pengulangan urutan perintah pengujian, sehingga kegagalan yang terjadi dapat direproduksi, pengujian dapat lebih cepat, serta pengujian otomatis memungkinkan pencatatan *response time* secara otomatis. (A.S 2014)

Untuk kebutuhan proses pengujian dilakukan dengan menggunakan *tools* yaitu Katalon Studio. *Tools* tersebut merupakan aplikasi *open source* untuk pengujian secara otomatis yang dikembangkan oleh Katalon LLC. Katalon merupakan antarmuka IDE khusus untuk pengujian API, web, dan seluler. Katalon memiliki tiga fitur utama untuk menunjang beberapa pengujian di beberapa platform yaitu API testing, Web testing dan Mobile testing. Selain itu, Katalon telah terintegrasi

dengan beberapa teknologi dari luar seperti github (LLC 2020).

BAB III METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

3.1.1. Bahan Penelitian

- a. Modul ESP 8266 sebagai mikrokontroler.
- b. Modul RC-SR04 sebagai sensor jarak.
- c. Wadah transparan yang didesain sebagai prototipe Ketinggian air .

3.1.2. Perangkat Lunak

Dalam penelitian ini perangkat lunak yang digunakan adalah:

- a. Arduino IDE (Integrated Development Environment) digunakan untuk melakukan pemrograman pada mikrokontroler yaitu ESP 8266 dan HC SR 04
- b. PHP 7.0.2 digunakan untuk akses API dari aplikasi ke database.
- c. Star UML digunakan untuk merancang UML.
- d. Apache digunakan sebagai web server yang digunakan pada server.
- e. Framework CodeIgniter 3.1 untuk membangun website sistem.
- f. Sistem informasi Windows 10 64 bit.
- g. Sublime text adalah text editor yang digunakan programmer dalam penulisan bahasa pemrograman dan bahasa markup.
- h. Xampp adalah salah satu *web server* yang berdiri sendiri (localhost).
- i. Geojson.io adalah tool online yang memudahkan membuat file Geojson berupa tipe geometri seperti point, linestring, polygon dan geometry lainnya. Lalu geojson tergenerate secara otomatis menjadi peta.
- j. Open Street Maps(OSM) adalah sebuah proyek berbasis web untuk membuat peta seluruh dunia yang gratis dan terbuka, dibangun sepenuhnya oleh sukarelawan dengan melakukan survey menggunakan GPS, mendigitasi citra satelit, dan mengumpulkan serta membebaskan data geografis yang tersedia di publik.

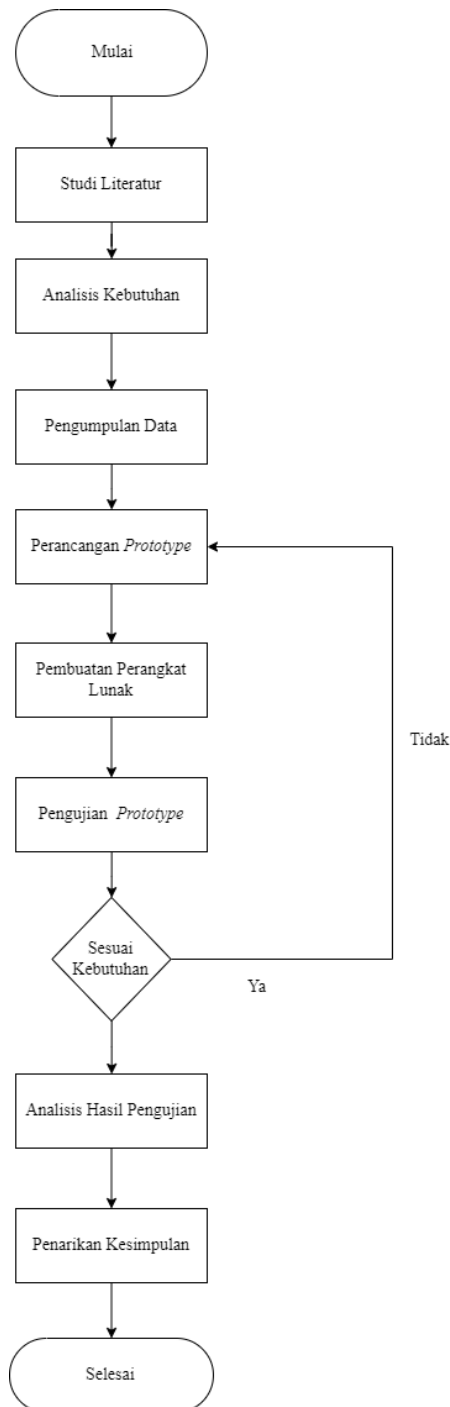
3.1.3. Perangkat Keras

Pembuatan sistem akan dimulai dari perancangan perangkat keras yang akan digunakan untuk memonitoring ketinggian air di atau parit. Adapun alat penunjang yang digunakan dalam penelitian ini adalah sebagai berikut.

- a. Satu unit laptop ASUS A456U dengan spesifikasi Intel Core i5, Nvidia Geforce 930, Ram 4GB, Harddisk 1 TB.
- b. Satu unit smartphone VIVO Z1 Pro dengan sistem operasi Android 10, Ram 6 GB, Memori Internal 128 GB dan layar 6' Inch

3.1.4. Metode Penelitian

Langkah-langkah penelitian yang dilakukan dalam pembuatan Visualisasi Banjir Secara Spasial Berdasarkan Data Sensor Ketinggian Berbasis Mikrokontroler Arduino(*Prototype*) dapat dilihat pada Gambar 3.1 berikut:



Gambar 3. 1 Diagram alir penelitian.

Berdasarkan gambar 3.1 langkah-langkah penelitian yang akan dilakukan adalah:

1. Studi literatur

Studi literatur merupakan tahap pencarian dan pemahaman teori dari referensi ilmiah. Teori yang didapat akan menjadi landasan dalam melakukan perancangan sistem. Pada penelitian ini dibutuhkan literatur mengenai dampak meluapnya air dan parit, cara kerja teknologi sistem tertanam yang meliputi cara kerja mikrokontroler dengan ESP 8266, cara membangun sebuah *webservice*, serta teori-teori dalam penelitian sebelumnya yang berkaitan dengan topik yang dibahas.

2. Analisis kebutuhan

Seperti yang telah dipaparkan pada awal bab, bahwa pada tahap ini akan dikumpulkan informasi terkait kebutuhan sistem untuk dapat memonitoring ketinggian air pada baik dari sisi *Hardware*, *Software* dan sumber daya manusia. Semua informasi akan coba diuraikan agar dapat sesuai dengan kebutuhan sistem yang akan dibangun.

3. Pengumpulan Data

Mengumpulkan berbagai data yang terkait pembangunan sistem informasi manajemen kerja praktek, meliputi data kemahasiswaan, dosen, dan berkas pelaksanaan kerja praktek.

4. Perancangan *Prototype*

Penulis mendeskripsikan apa yang harus dilakukan sistem agar sesuai dengan kebutuhan pengguna. Untuk memenuhi kebutuhan sistem ini, maka dirancang *prototype* dari sisi *Hardware* dan *Software*. Kemudian dibangun juga sebuah *prototype* tempat air yang menjadi media untuk pengujian dengan beberapa objek air yang berbeda. Secara keseluruhan sistem dirancang untuk memenuhi fungsionalitas sistem yaitu *monitoring* ketinggian air yang mana pengguna dapat melakukan pemantauan ketinggian air kapan pun dan di mana pun melalui aplikasi *web*. Dalam

melakukan perancangan akan dibantu dengan *Unified Modelling Language (UML) Diagram*.

5. Pembuatan Perangkat Lunak

Pembuatan perangkat lunak merupakan proses menerjemahkan desain sistem ke dalam produk nyata. Pada tahap ini sistem mulai dikembangkan berdasarkan desain yang telah dibuat. Dalam proses implementasi, mulai dilakukan penerjemahan desain menggunakan kode bahasa program dan konfigurasi sistem agar program dapat berjalan dengan baik. Sistem yang dibuat merupakan sistem berbasis website, sehingga sistem ini diimplementasikan menggunakan bahasa pemrograman PHP dengan *framework codeigniter* untuk manajemen data dan *framework CSS bootstrap* untuk antarmuka (interface).

6. Pengujian Prototype

Pengujian dan evaluasi program yang dilakukan dengan mengetahui apakah aplikasi telah memenuhi kebutuhan pengguna, *user friendly* dan efisien. Program yang telah dibuat diuji cobakan untuk mengetahui apakah ada kesalahan. Bila terdapat kesalahan dalam pemrograman maka akan dilakukan analisa ulang untuk melihat proses perancangan sistem dan bila terjadi ketidak sesuaian data yang digunakan maka akan kembali melihat proses pengumpulan data, apakah terdapat data yang tidak sesuai atau kekurangan data. Pengujian dilakukan menggunakan metode blackbox, UAT dan pengujian katalon.

7. Analisis Hasil Pengujian

Analisis yang dilakukan dalam tahap pengujian dan validasi untuk mengetahui karakteristik system yang dibangun dan mengidentifikasi konsisten sistem. Hasil analisis akan digunakan sebagai dasar perbaikan.

8. Penarikan Kesimpulan

Berisi kesimpulan-kesimpulan dari proses pengembangan perangkat lunak, baik pada tahap analisis kebutuhan perangkat lunak, perancangan, pembuatan, dan terutama pada analisis kinerja perangkat lunak. Bagian ini juga berisi saran-saran yang perlu diperhatikan berdasar keterbatasan-

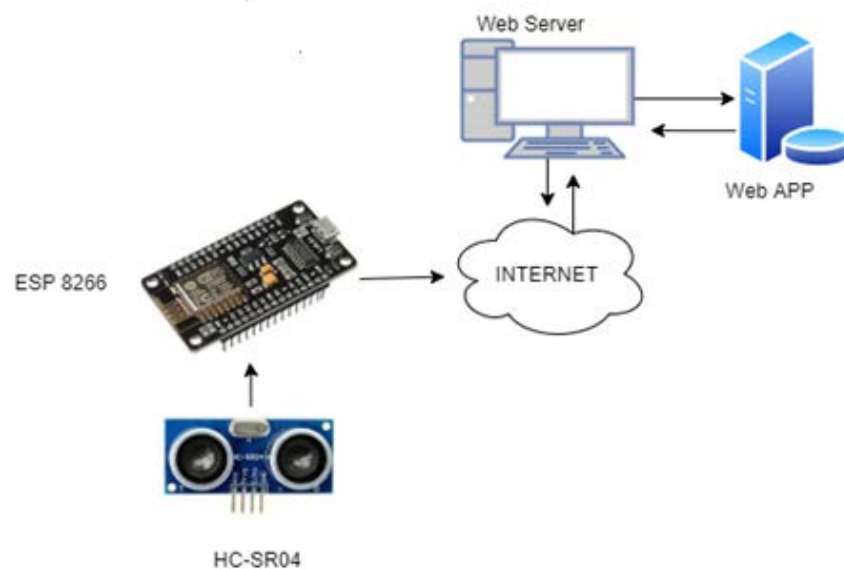
keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama melakukan kerja praktek.

3.2 Perancangan Sistem

3.2.1. Analisis Kebutuhan Sistem

3.2.1.1. Analisis Kebutuhan Sistem

Desain arsitektur sistem monitoring ketinggian air yang dirancang disajikan dalam Gambar 3.4 berikut.



Gambar 3. 2 Arsitektur Sistem Monitoring Ketinggian Air

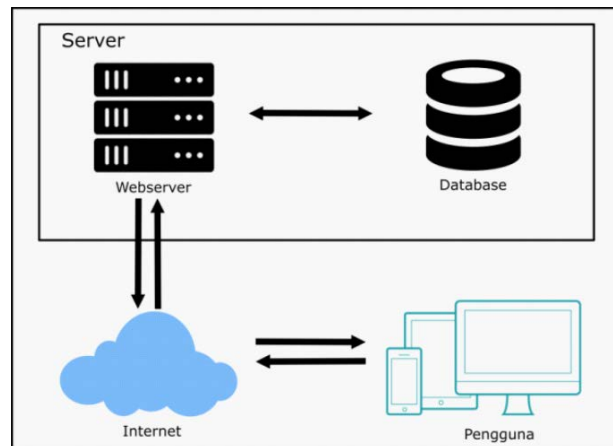
Dari gambar 3.2 di atas dapat dijabarkan rincian proses kerja sistem sebagai berikut.

- a. ESP8266 sebagai kendali dari sistem ketika mendapat *supply* tegangan 5 Volt, maka ESP8266 akan menginisialisasi semua perangkat yang telah terhubung, seperti Sensor HC-SR04. Seiring dengan itu ESP8266 akan mencoba terhubung ke internet melalui jaringan Wifi yang telah ditentukan.
- b. Kemudian,ESP8266 me-*request* data jarak dari sensor HC-SR04.

- c. Sensor HC-SR04 mengukur jarak antara permukaan tumpukan air dengan sensor, lalu mengirimkannya ke *Web server* menggunakan Wifi di perangkat ESP8266, hasil pengukuran jarak ini nantinya akan menjadi data ketinggian air yang bisa dipantau secara *real time* dan selanjutnya bisa digunakan sebagai referensi untuk melakukan kontrol pada tempat air tersebut.
- d. ESP8266 yang telah berhasil terhubung ke internet akan mengirimkan data hasil pengukuran dari HC-SR04 ke *database* server berupa data jarak dengan satuan *centimeter*.
- e. Data tingkat ketinggian air di dalam tempat air ditampilkan melalui *website* sistem.
- f. Pengguna secara umum aplikasi *website* bisa memantau kondisi air melalui peta persebaran dan pengguna yang berstatus sebagai admin dapat melakukan manajemen, sensor, melihat detail hasil baca sensor, melihat grafik hasil baca sensor.
- g. ESP8266 akan memberikan notifikasi berupa tampilan yang menyalakan di PETA dengan 3 (tiga) parameter kepenuhan yaitu, kondisi Normal dengan tampilan zona di peta berwarna hijau, kondisi Peringatan dengan tanda warna Zona kuning dan kondisi Berbahaya dengan zona merah yang menyala sesuai dengan hasil baca sensor.

3.2.1.2. Desain Arsitektur Aplikasi Website

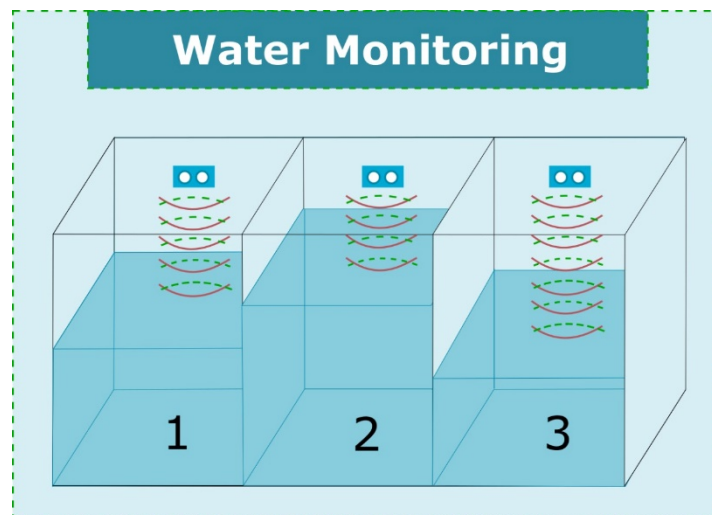
Pada arsitektur ini antara pengguna dan *server* akan dihubungkan melalui internet. Pengguna akan melakukan *request* ke *server* untuk mendapatkan informasi mengenai ketinggian air pada wadah. Rancangan arsitektur aplikasi bisa dilihat pada Gambar 3.3.



Gambar 3. 3 Desain Arsitektur Aplikasi Web

3.2.1.3. Desain Prototipe dan Perhitungan Ketinggian Air

Pada tahap ini menentukan prototipe yang akan digunakan. Prototipe ini akan berfungsi sebagai media uji coba perangkat pengukur ketinggian air yang telah dibuat. Adapun desain prototipe tersebut bisa dilihat pada Gambar 3.4.



Gambar 3. 4 Desain Prototipe Wadah Air

Pada gambar 3.4 kita dapat melihat perangkat pengukur ketinggian air akan diletakkan di bagian atas wadah. Wadah itu sendiri dibagi menjadi tiga blok sesuai dengan sensor yang digunakan di sisinya.

Pada tahap selanjutnya adalah perancangan perhitungan ketinggian air pada wadah air . Prinsip kerja sensor ultrasonik HC-SR04 pada umumnya digunakan sebagai pengukur jarak. Pada sistem pemantau ketinggian air yang dibangun ini, sensor HC-SR04 digunakan sebagai pengukur jarak permukaan air . Sensor HC-SR04 yang digunakan berjumlah 1 buah dan di tempatkan di atas permukaan air menghadap ke bawah seperti yang ditunjukkan pada Gambar 3.5. Untuk menentukan nilai ketinggian air, maka penulis menentukan tinggi awal sebelum diisi air dikurangi dengan nilai tinggi air di wadah air tersebut. Nilai tersebut menentukan wadah dikatakan sudah berbahaya, hampir penuh dan belum penuh. Berikut persamaan yang digunakan.

$$\Delta T_s = J_0 - \Delta J$$

(3.1)

Dimana :

T_s : Jarak ketinggian Permukaan air air .

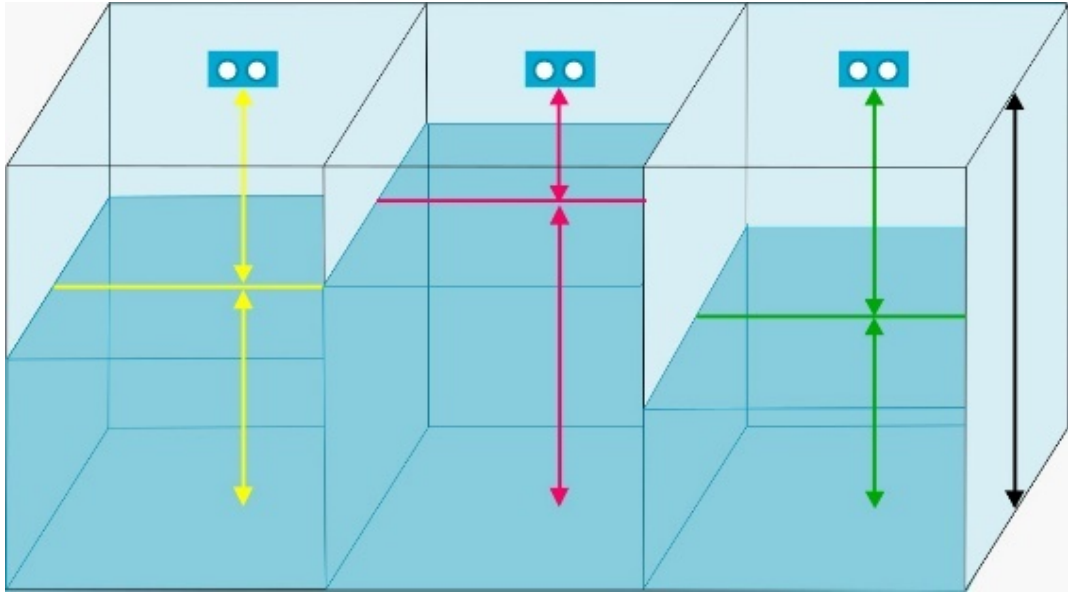
J_0 : Jarak sensor dengan permukaan wadah yang belum berisi air .

J_1 : Jarak sensor 1 pada blok wadah air .

J_2 : Jarak sensor 2 pada blok wadah air .

J_3 : Jarak sensor 3 pada blok wadah air .

T_1 : Jarak sensor dengan permukaan air .



Gambar 3. 5 Perancangan Perhitungan Ketinggian Air pada Wadah

3.2.1.4. Analisis Kebutuhan *Hardware* dan *Software*

Dalam analisis kebutuhan *Hardware* dan *Software* ini, seluruh tools yang digunakan untuk membuat sistem informasi manajemen kerja praktek berbasis website didefinisikan. Dari hasil analisis, untuk dapat membuat dan menjalankan sistem informasi praktik kerja industri berbasis website, syarat yang harus dipenuhi yaitu:

- a. Untuk membuat sistem informasi manajemen berbasis website perangkat yang digunakan adalah PC/ Laptop, *Framework Codeigniter 3.1*, *Framework Bootstrap 4.0*, *Web Server Apache*, *Database Server MySQL*, *Draw io*, *Sublime text* dan *Web Browser*.
- b. Untuk dapat menjalankan atau mengakses sistem informasi manajemen ini perangkat yang digunakan adalah perangkat yang memiliki sambungan internet seperti smartphone , Komputer desktop atau laptop dan *web browser*. *Web browser* yang dapat digunakan antara lain *Mozilla Firefox*, *Google Chrome*.

3.2.1.5. Analisis Unified Modelling Language (UML)

Perancangan untuk Prototipe Sistem *Monitoring* Ketinggian Air digambarkan oleh penulis menggunakan UML (*Unified Modelling Language*).

Model UML yang digunakan adalah *Use Case Diagram*, *Sequence Diagram* dan *Activity Diagram*.

Adapun fitur dari Prototipe Sistem *Monitoring* Ketinggian Air ini secara umum adalah menyediakan informasi ketinggian air dalam bentuk visual pada aplikasi *web*.

3.2.1.5.1. Use Case Diagram

Use Case Diagram adalah sebuah *Diagram* yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Bagian yang ditekankan dari *Use Case Diagram* adalah “apa” bukan “bagaimana”. Sistem yang dibangun terdiri dari 1 (satu) aktor yaitu admin.

3.2.1.5.2. Definisi Aktor

Definisi aktor aplikasi sistem pemantauan ketinggian air menggunakan mikrokontroler arduino dan aplikasi berbasis *web* dapat dilihat pada Tabel 3.1 berikut

Tabel 3. 1 Definisi Aktor

No.	Aktor	Deskripsi
1	Masyarakat	Mayarakat hanya dapat mengakses halaman peta monitoring ketinggian air tanpa harus <i>login</i> .
2	Admin	Admin di sini berperan sebagai admin sistem yang melakukan <i>login</i> untuk mengakses dashboard admin di mana dapat melakukan manajemen terhadap, sensor dan admin. Kemudian bisa melihat grafik ketinggian air .

No.	Aktor	Deskripsi
		Lalu melakukan <i>logout</i> .

3.2.1.5.3. Definisi Use Case

Definisi *Use Case Diagram* sistem pemantauan ketinggian air menggunakan mikrokontroler arduino dan aplikasi berbasis *web* dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Definisi Use Case

No.	Use Case	Deskripsi
1	Validasi	Merupakan proses pengecekan hak akses siapa yang berhak mengakses informasi ketinggian air , yang dalam kasus ini ada admin. Kemudian hak akses proses pengelolaan data ketinggian air dan pengelolaan data pengguna sistem yang pada kasus ini merupakan hak khusus bagi admin sistem. Login wajib untuk fungsi-fungsi yang berkaitan dengan akses perubahan ke basis data, oleh karena itu fungsi-fungsi yang melakukan perubahan basis data harus mengecek validasi pengguna yang mengakses fungsi-fungsi tersebut. Validasi merupakan generalisasi dari proses <i>login</i> , <i>logout</i> dan cek status login.
2	<i>Login</i>	Merupakan proses untuk <i>login</i> pengguna ke <i>website</i> sistem.
3	<i>Logout</i>	Merupakan proses untuk <i>logout</i> pengguna dari <i>website</i> sistem.
4	Cek Status <i>Login</i>	Merupakan proses untuk memeriksa apakah admin sistem sudah melakukan <i>login</i> atau belum.
5	Melihat Peta	Merupakan proses melihat status ketinggian air terakhir

No.	Use Case	Deskripsi
	<i>Monitoring</i>	dari tabel <i>sensor result</i> yang disajikan dalam bentuk peta persebaran air .
6	Tentang Aplikasi	Merupakan proses untuk menampilkan pengembang aplikasi dan penjelasan singkat mengenai cara kerja sistem.
7	Melihat Status Sensor	Proses <i>monitoring</i> terhadap ketinggian terakhir air dalam <i>database</i> yang ditangkap oleh sensor pada alat ketinggian air kemudian disajikan oleh aplikasi dalam bentuk nilai persentase ketinggian air disertai gambar status ketinggian air apakah normal, waspada, peringatan.
8	Melihat Grafik	Proses <i>monitoring</i> terhadap ketinggian air dalam bentuk grafik ketinggian air per waktu yang ditentukan oleh sistem, sehingga informasi tersebut berguna untuk dianalisa yang berujung pada tindakan lebih lanjut oleh pihak terkait.
9	Mengelola Sensor	Merupakan proses generalisasi yang meliputi empat buah proses pengelolaan data sensor sistem yaitu tambah sensor, ubah sensor, hapus sensor dan melihat sensor.

3.2.1.5.4. Skenario Use Case

Skenario jalannya masing-masing *Use Case Diagram* yang telah didefinisikan sebelumnya dapat dilihat pada Tabel 3.4 (a-s) berikut.

Tabel 3. 3 Skenario Use Case Diagram: Melihat Peta Monitoring

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu lihat peta	

	2. Mengambil data pada database sistem untuk ditampilkan dalam bentuk peta. Kemudian mengambil data terakhir <i>sensor results</i> pada database untuk ditampilkan status ketinggian air pada wadah tersebut.
--	---

Tabel 3. 4 Skenario Use Case Diagram: Tentang Aplikasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu tentang aplikasi	
	2. Menampilkan penjelasan singkat mengenai aplikasi sistem dan data pengembang aplikasi.

Tabel 3. 5 Skenario Use Case Diagram: Login

Aksi Aktor	Reaksi Sistem
Skenario Normal	
4. Memasukan <i>username</i> dan <i>password</i>	
	5. Memeriksa valid tidaknya data masukan dengan memeriksa tabel <i>users</i>
	6. Masuk ke aplikasi <i>monitoring</i> ketinggian air
Skenario Alternatif	
1. Memasukan <i>username</i> dan <i>password</i>	
	2. Memeriksa valid tidaknya data yang dimasukan

	3. Menampilkan pesan <i>login</i> tidak valid
4. Memasukan <i>username</i> dan <i>password</i> yang valid	
	5. Memeriksa valid tidaknya data masukan
	6. Masuk ke aplikasi <i>monitoring</i> ketinggian air

Tabel 3. 6 Skenario Use Case Diagram: Logout

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu logout	
	2. Melakukan logout

Tabel 3. 7 Skenario Use Case Diagram: Cek Status Login

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Memeriksa ke variabel <i>session</i> sebagai penanda <i>login</i> apakah pengguna sudah <i>login</i>
	2. Mengembalikan status <i>login</i> , sudah <i>login</i> atau belum

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Memeriksa status login
2. Memasukan data sensor sesuai kolom yang ada	
	3. Memeriksa valid tidaknya data masukan
	4. Menyimpan data sensor ke basis data
	5. Menampilkan pesan sukses disimpan
Skenario Alternatif	
	1. Memeriksa status login
2. Memasukan data sensor sesuai kolom yang ada	
	3. Memeriksa valid tidaknya data masukan
	4. Mengeluarkan pesan bahwa data masukan tidak valid
5. Memperbaiki data masukan yang tidak valid	
	6. Memeriksa valid tidaknya data masukan
	7. Menyimpan data sensor ke basis data

Aksi Aktor	Reaksi Sistem
	8. Menampilkan pesan sukses disimpan

Tabel 3. 8 Skenario Use Case Diagram: Melihat Grafik

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Memeriksa status login
	2. Mengambil data hasil baca tiap sensor pada database sistem untuk diolah pada aplikasi sistem.
	3. Menampilkan data hasil olahan aplikasi sistem berupa rata-rata ketinggian air dalam bentuk grafik per waktu yang telah ditentukan oleh sistem

Tabel 3. 9 Skenario Use Case Diagram: Tambah Sensor

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	6. Memeriksa status login
7. Memasukan data sensor sesuai kolom yang ada	
	8. Memeriksa valid tidaknya data masukan
	9. Menyimpan data sensor ke basis data
	10. Menampilkan pesan sukses disimpan
Skenario Alternatif	

	9. Memeriksa status login
10. Memasukan data sensor sesuai kolom yang ada	
	11. Memeriksa valid tidaknya data masukan
	12. Mengeluarkan pesan bahwa data masukan tidak valid
13. Memperbaiki data masukan yang tidak valid	
	14. Memeriksa valid tidaknya data masukan
	15. Menyimpan data sensor ke basis data
	16. Menampilkan pesan sukses disimpan

Tabel 3. 10 Skenario Use Case Diagram: Ubah Sensor

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Memeriksa status login
2. Memasukan kata kunci dan kategori pencarian	
	3. Mencari data sensor yang akan diubah
	4. Menampilkan data sensor yang dicari (belum semua kolom data sensor ditampilkan dan bisa banyak data sensor yang memenuhi data pencarian)

5. Memilih data sensor yang akan diubah	
	6. Menampilkan semua kolom data sensor yang akan diubah
7. Mengubah data sensor	
	8. Memeriksa valid tidaknya data masukan
	9. Menyimpan data yang telah diubah ke basis data
	10. Menampilkan pesan bahwa data sukses disimpan
Skenario Alternatif	
	1. Memeriksa status login
2. Memasukan kata kunci dan kategori pencarian	
	3. Mencari data sensor yang akan diubah
	4. Menampilkan data sensor yang dicari (belum semua kolom data sensor ditampilkan dan bisa banyak data sensor yang memenuhi data pencarian)
5. Memilih data sensor yang akan diubah	
	6. Menampilkan semua kolom data mesin yang akan diubah

Aksi Aktor	Reaksi Sistem
7. Mengubah data sensor	
	8. Memeriksa valid tidaknya data masukan
	9. Menampilkan pesan bahwa data masukan tidak valid
10. Memperbaiki data masukan yang tidak valid	
	11. Memeriksa valid tidaknya data masukan
	12. Menyimpan data yang telah diubah ke basis data
	13. Menampilkan pesan bahwa data sukses disimpan

Tabel 3. 11 Skenario Use Case Diagram: Hapus Sensor

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Memeriksa status login
2. Memasukan kata kunci dan kategori pencarian	
	3. Mencari data sensor yang akan diubah
	4. Menampilkan data sensor yang dicari (belum semua kolom datasensor ditampilkan dan bisa banyak data sensor yang memenuhi data pencarian)

Aksi Aktor	Reaksi Sistem
5. Memilih data sensor yang akan dihapus	
	6. Menampilkan pesan konfirmasi apakah data akan benar-benar dihapus
7. Mengklik pilihan setuju data dihapus	
	8. Menghapus data sensor dari basis data
	9. Menampilkan pesan bahwa data sukses dihapus
Skenario Alternatif	
	1. Memeriksa status login
2. Memasukan kata kunci dan kategori pencarian	
	3. Mencari data sensor yang akan dihapus
	4. Menampilkan data sensor yang dicari (belum semua kolom data sensor ditampilkan dan bisa banyak data sensor yang memenuhi data pencarian)
5. Memilih data sensor yang akan dihapus	
	6. Menampilkan pesan konfirmasi apakah data akan benar-benar dihapus

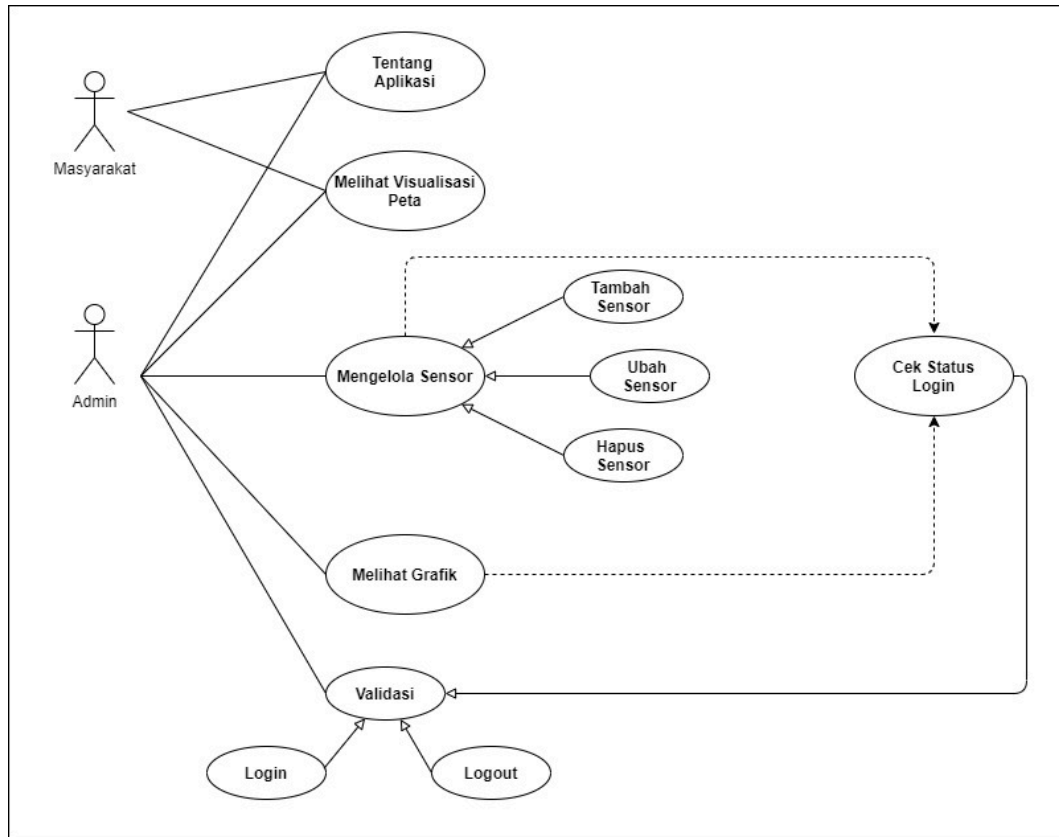
Aksi Aktor	Reaksi Sistem
7. Mengklik pilihan tidak setuju data dihapus	
	8. Kembali ke form pencarian sensor

Tabel 3. 12 Skenario Use Case Diagram: Lihat Sensor

Aksi Aktor	Reaksi Sistem
Skenario Normal	
	1. Memeriksa status login
	2. Menampilkan data sensor yang dicari (belum semua kolom data sensor ditampilkan dan bisa banyak data sensor yang memenuhi data pencarian)
3. Memilih sensor yang akan dicari	
	4. Menampilkan data sensor (semua kolom) dari sensor yang dipilih

3.2.1.5.5. Use Case Diagram Sistem

Berdasarkan analisis kebutuhan dibuatlah *Use Case Diagram* untuk membantu memberikan gambaran mengenai interaksi apa saja yang dapat dilakukan oleh *users* saat menggunakan sistem. *Diagram Use Case Diagram Sistem monitoring* dapat dilihat pada Gambar 3.8 berikut.

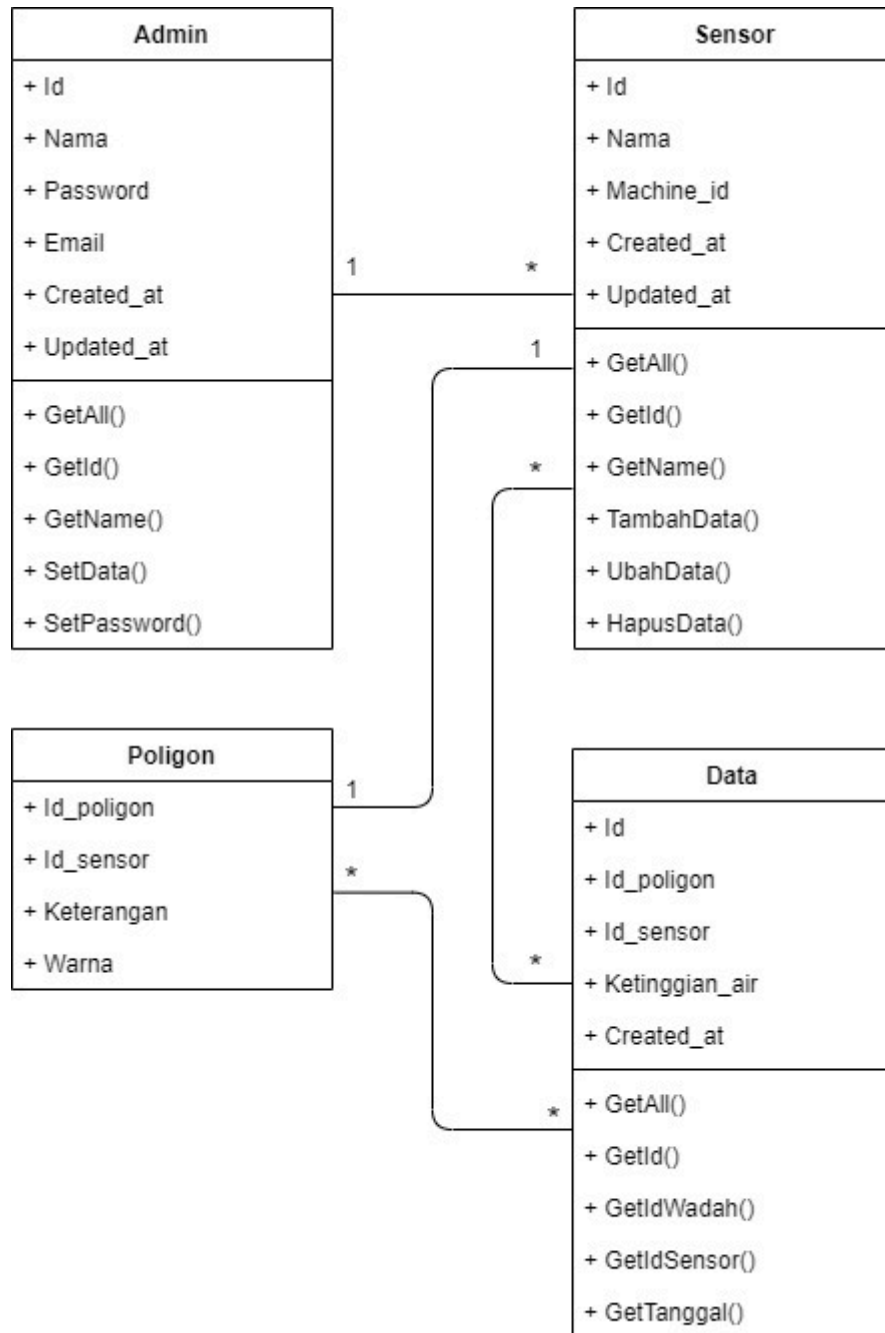


Gambar 3.8 *Diagram Use Case Diagram Sistem*

Gambar 3.8 menggambarkan fungsi dari aplikasi yang akan dibuat. Pada gambar diberi 3 aktor, yaitu petugas dan masyarakat dan pengguna. Semua *user* dapat mengakses halaman peta *monitoring* ketinggian air. Khusus *user* yang berstatus admin bisa melakukan beberapa aktivitas seperti melihat status wadah, melihat grafik, manajemen admin dan manajemen Sensor.

3.2.1.6. Class Diagram

Class Diagram digunakan untuk melakukan perancangan struktur kelas-kelas yang terdapat pada aplikasi yang akan dibuat. Adapun *class Diagram* aplikasi yang akan dibuat dapat dilihat pada Gambar 3.9 berikut.



Gambar 3.9 *Class Diagram* Sistem

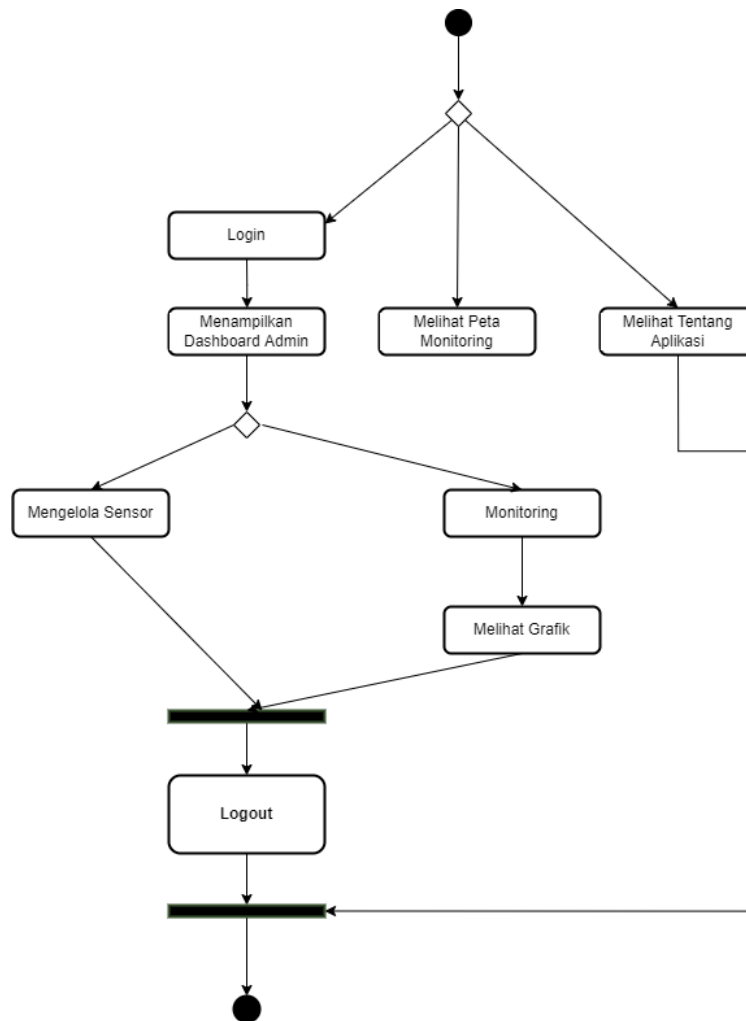
Gambar 3.9 memberikan gambaran struktur dan deskripsi *class*, serta hubungan antara *class* yang akan dibangun dalam sistem. Kelas-kelas yang ada pada struktur sistem di atas memiliki tugas masing-masing sesuai dengan fungsi-fungsi yang ditentukan oleh sistem.

3.2.1.7. Sequence Diagram

Sequence Diagram menjelaskan permodelan sistem yang menunjukkan interaksi antara objek yang disusun berdasarkan urutan waktu. *Diagram* ini merupakan proses yang ada pada *use case*. *Sequence Diagram* dari sistem pemantau ketinggian air dapat dilihat pada lampiran A.

3.2.1.8. Activity Diagram

Activity Diagram akan membahas alur aktifitas dalam sistem yang akan dibuat, mulai dari bagaimana alur berawal, *decision* yang mungkin terjadi dan bagaimana akan berakhir. Berikut *Diagram* aktifitas dapat dilihat pada Gambar 3.10.

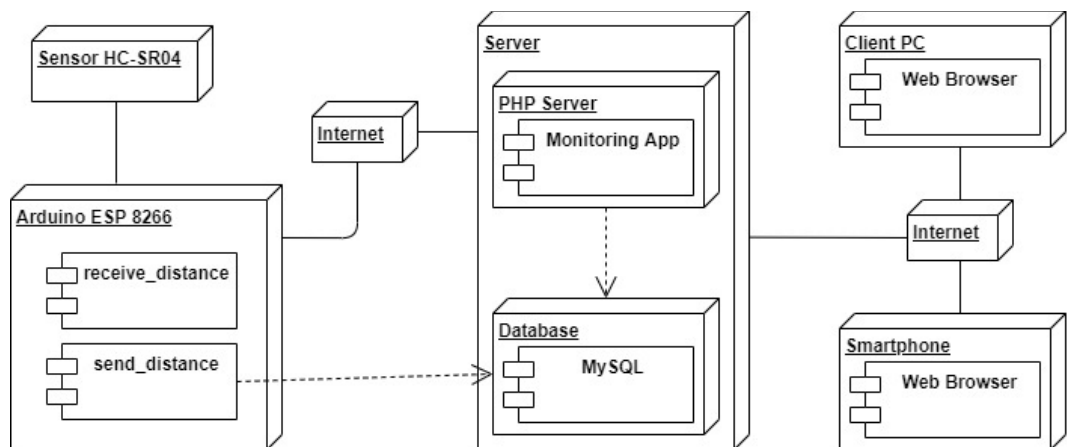


Gambar 3.10 Activity Diagram

Gambar 3.10 menggambarkan alur user dalam menggunakan aplikasi baik sebagai admin maupun petugas. Aktifitas dimulai dengan mengakses URL *website*, menampilkan halaman menu dimana *users* harus *login*, baik sebagai admin maupun petugas. Keduanya bisa sama-sama mengakses halaman *monitoring*, perbedaannya adalah jika login sebagai admin maka sistem memberikan hak kelola sistem seperti mengelola *users*, mengelola *machines* dan mengelola *sensors*. Kemudian melakukan *logout*.

3.2.1.9. Deployment Diagram

Deployment Diagram akan menjelaskan bagaimana sistem saling berhubungan, mulai dari perangkat untuk mengambil data, yaitu sensor HC-SR04 yang tersambung pada ESP 8266, hingga perangkat untuk melihat data yang telah diambil, yaitu client PC dan smartphone android. Berikut *deployment Diagram* dapat dilihat pada Gambar 3.11.



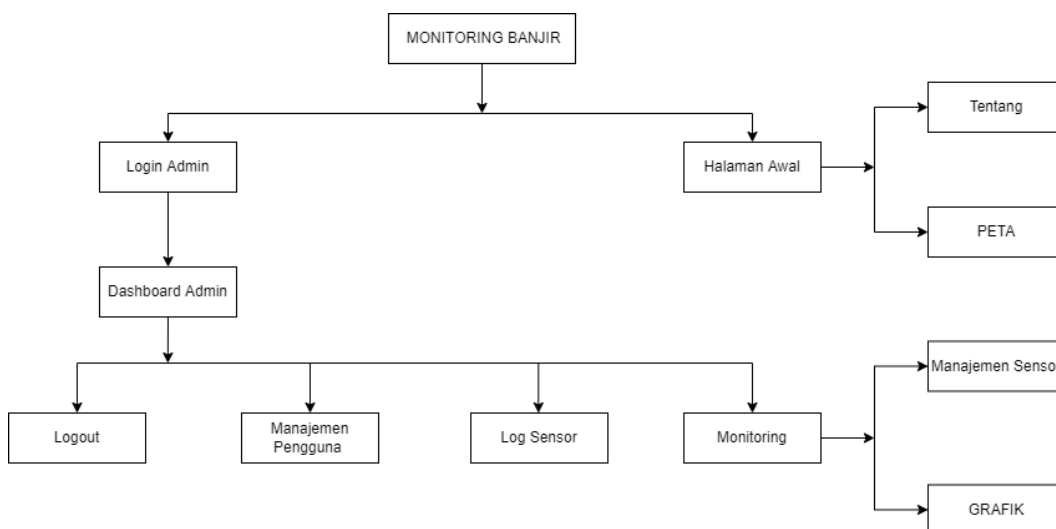
Gambar 3.11 *Deployment Diagram* Sistem

3.2.2. Antarmuka Sistem

Pada bagian ini akan dijelaskan mengenai perancangan antarmuka sistem pada aplikasi *web*

3.2.2.1. Rancangan Struktur Antarmuka Aplikasi Web

Struktur antarmuka pada *website* sistem monitoring ketinggian air dapat dilihat pada Gambar 3.6 berikut.



Gambar 3. 6 Struktur Antarmuka *Website* Sistem

Penjelasan dari Gambar 3.6 di atas adalah sebagai berikut.

a. MONITORING BANJIR

MONITORING BANJIR merupakan halaman utama ketika pengguna mengakses URL (*Uniform Resource Locator*) *website*.

b. Halaman Awal

Halaman ini berisi informasi umum terkait *monitoring* ketinggian air yang bisa diakses oleh pihak manapun selain admin, diantaranya adalah :

a) Tentang

Halaman ini berisi tentang aplikasi sistem dan paparan ringkas mengenai fungsionalitas sistem.

b) PETA

Halaman ini merupakan maps persebaran di kota Pontianak disertai dengan informasi visual tentang ketinggian air pada setiap pintu air.

Login Admin

Halaman ini berisi form username dan password yang digunakan untuk masuk ke dashboard admin.

c) Dashboard Admin

Halaman ini merupakan halaman yang hanya bisa diakses oleh pengguna dengan status admin. Beberapa menu dalam halaman ini adalah :

d) Monitoring

Menu ini merupakan fitur inti dari sistem, di mana admin dapat melakukan beberapa hal sebagai berikut.

1. Manajemen Sensor

Melakukan fungsi tambah, ubah dan hapus sensor.

2. GRAFIK

Melakukan pengamatan terhadap perkembangan atau perubahan ketinggian air dalam waktu satu hari.

e) Log Sensor

Merupakan menu untuk melihat history dari hasil baca sensor yang masuk ke *database server*.

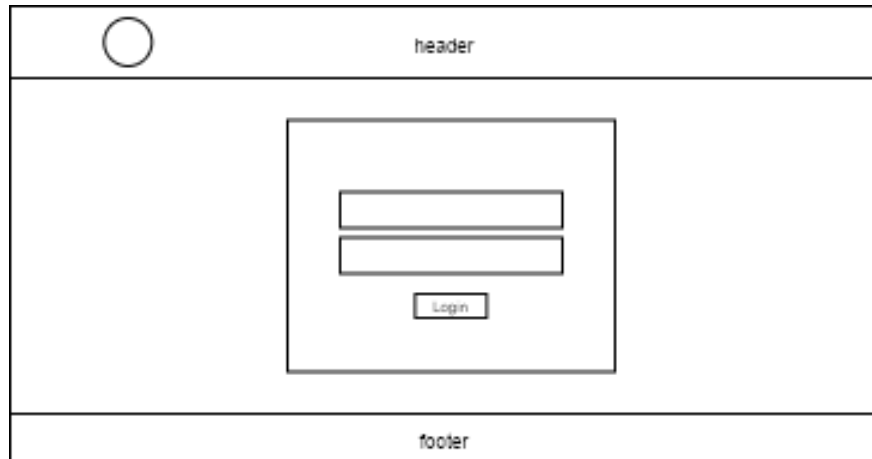
f) Logout

Merupakan menu untuk keluar dari dashboard admin dan kembali ke halaman awal sistem.

3.2.2.2. Layout Aplikasi

3.2.2.2.1. Desain Halaman Login Admin

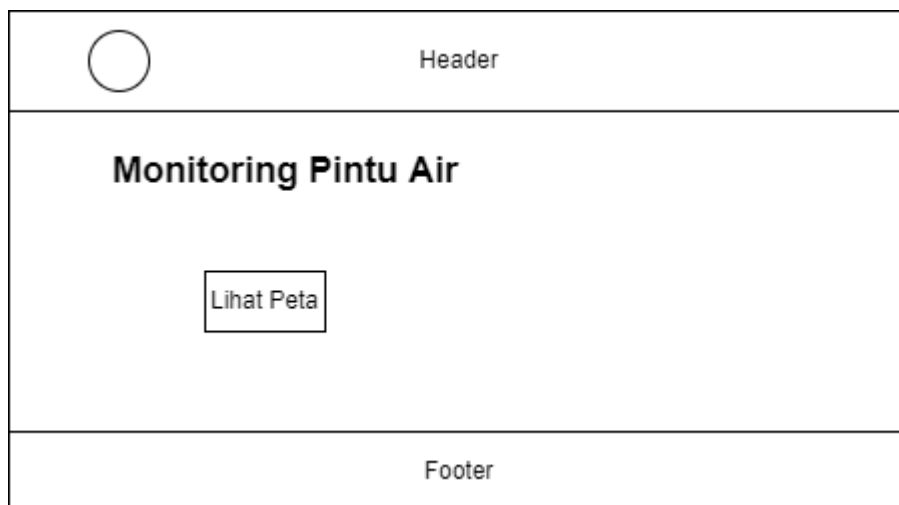
Pada halaman ini, pengguna disajikan *form* yang harus diisi dengan *username* dan *password* yang telah terdaftar agar pengguna dapat masuk ke halaman admin *website*. Gambar 3.13 merupakan gambar rancangan halaman *login* admin aplikasi.



Gambar 3. 7 Desain Halaman Login Admin Aplikasi

3.2.2.2.2. Desain Halaman Awal Monitoring

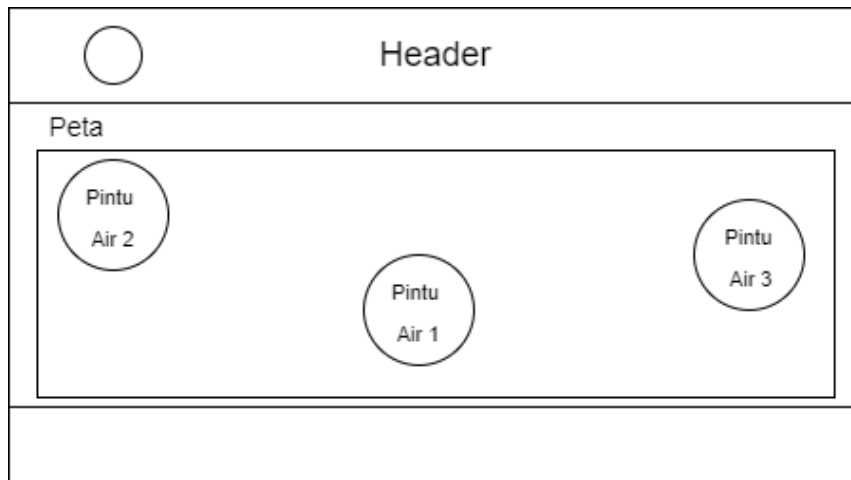
Pada halaman monitoring terbagi menjadi dua, yaitu halaman perkenalan aplikasi dan halaman peta *monitoring*. Dapat dilihat pada Gambar 3.8 dan 3.9 berikut.



Gambar 3. 8 Desain Halaman Tentang Aplikasi

3.2.2.2.3. Desain Halaman Peta

Perancangan halaman *dashboard* admin dapat dilihat pada Gambar 3.16 berikut.

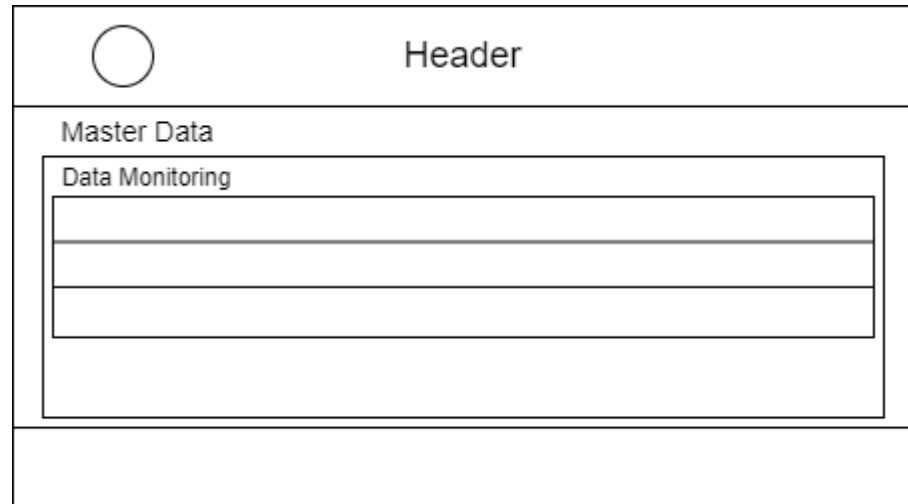


Gambar 3. 9 Desain Halaman Peta Monitoring

Pada Gambar 3.9 akan menampilkan deskripsi singkat dari Peta aplikasi web yang dibangun. Kemudian pada Gambar 3.15 akan ditampilkan hasil *monitoring* dalam bentuk peta persebaran.

3.2.2.2.4. Desain Halaman Dashboard Master Data Monitoring Admin

Perancangan halaman *Dashboard* Master Data Monitoring Admin dapat dilihat pada Gambar 3.10 berikut.

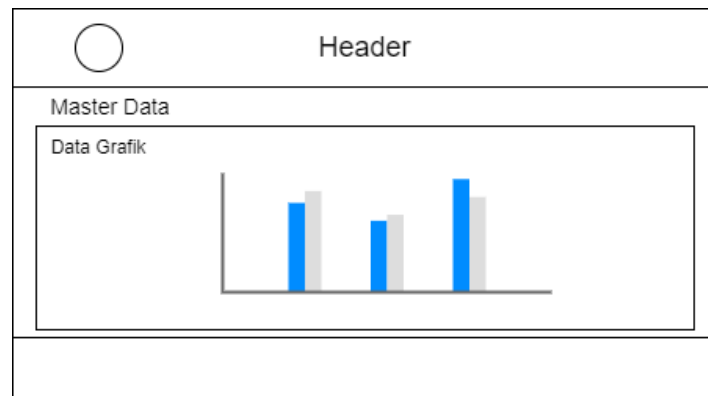


Gambar 3. 10 Desain Halaman Peta Monitoring

Pada Gambar 3.10 akan menampilkan deskripsi singkat dari kegunaan aplikasi web yang dibangun. Kemudian pada Gambar 3.15 akan ditampilkan hasil *monitoring* dalam bentuk peta persebaran.

3.2.2.2.5. Desain Halaman Dashboard Master Data Grafik Admin

Perancangan halaman *dashboard* admin dapat dilihat pada Gambar 3.16 berikut.



Gambar 3. 11 Desain Halaman Master Data Grafik

Pada Gambar 3.11 akan menampilkan deskripsi singkat dari kegunaan aplikasi web yang dibangun. Kemudian pada Gambar 3.12 akan ditampilkan hasil *monitoring* dalam bentuk Grafik persebaran.

3.2.3. Perancangan *Database*

3.2.3.1. Spesifikasi *Database*

Tabel yang digunakan dalam penelitian ini dibuat menggunakan *database* MySQL. Berikut adalah tabel-tabel yang dibuat dalam sistem yang dibangun:

3.2.3.1.1. Spesifikasi Tabel Akun

nama *database* : Banjir

nama tabel : Admin

keterangan : tabel berisi data *admin* yang terdaftar pada sistem

Tabel 3. 13 Spesifikasi Tabel *Admins*

Nama <i>Field</i>	Tipe	Boleh Kosong	<i>Key</i>	<i>Extra</i>
<i>Id</i>	int(20)	Tidak	<i>Primary key</i>	AUTO_INCREMENT
<i>Name</i>	varchar(255)	Tidak		
<i>Password</i>	varchar(255)	Tidak		

3.2.3.1.2. Spesifikasi Tabel Sensor

nama *database* : Banjir

nama tabel : Sensor

keterangan : Tabel berisi data Sensor yang terpasang dalam sistem

Tabel 3. 14 Spesifikasi Tabel Sensor

Nama <i>Field</i>	Tipe	Boleh Kosong	<i>Key</i>	<i>Extra</i>
<i>Id</i>	int (20)	Tidak	<i>Primary key</i>	AUTO_INCREMENT
<i>Name</i>	varchar (255)	Tidak		
<i>Lokasi</i>	varchar (255)	Tidak		
<i>Lat</i>	double (11, 8)	Tidak		
<i>Long</i>	double (11, 8)	Tidak		

3.2.3.1.3. Spesifikasi Tabel Data

nama *database* : Monitoring Banjir
 nama tabel : Data
 keterangan : tabel berisi data dan *sensor*

Tabel 3. 15 Spesifikasi Tabel Data

Nama Field	Tipe	Boleh Kosong	Key	Extra
<i>Id_data</i>	int (20)	Tidak	<i>Primary key</i>	AUTO_INCREMENT
<i>Id_sensor</i>	int (20)	Tidak		
<i>Id_poligon</i>	Int (20)	Tidak		
<i>Ketinggian</i>	varchar (255)	Tidak		
<i>Tanggal</i>	Datetime	Tidak		

3.2.3.1.4. Spesifikasi Tabel Poligon

Nama *database* : Monitoring Banjir
 Nama tabel : Poligon
 keterangan : tabel berisi data poligon

Tabel 3. 16 Spesifikasi Data Poligon

Nama Field	Tipe	Boleh Kosong	Key	Extra
<i>Id_poligon</i>	int (20)	Tidak	<i>Primary key</i>	AUTO_INCREMENT
<i>Id_sensor</i>	int (20)	Tidak		
<i>Keterangan</i>	varchar (255)	Tidak		
<i>Warna</i>	varchar (255)	Tidak		

3.3 Perancangan Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan dan kekurangan pada sistem yang dibangun. Pengujian akan dilakukan pada sisi perangkat keras (*Hardware*) yang meliputi komponen-komponen pembangun rangkaian mikrokontroler sistem dan perangkat lunak (*Software*) yang meliputi aplikasi Arduino IDE dan aplikasi *web monitoring* yang dibangun.

3.3.1 Pengujian *Hardware*

Pada tahap ini akan dilakukan pengujian terhadap komponen-komponen yang membangun rangkaian mikrokontroler untuk mengetahui apakah sistem yang berfungsi sebagai seperangkat alat pengukur ketinggian air sudah berjalan sebagaimana mestinya atau tidak. Adapun tahapan pengujian *Hardware* adalah sebagai berikut.

1. Pengujian *Hardware* rangkaian mikrokontroler secara umum.
2. Pengujian sensor HC-SR04 pengukur ketinggian air .
3. Pengujian ESP8266 dalam mengirim data ke *database*.

Untuk memudahkan dalam analisa hasil pengujian, maka setiap hasil dari langkah pengujian akan disampaikan dalam bentuk tabel. Berikut adalah tabel yang dibuat untuk pengujian rangkaian mikrokontroler secara umum.

Tabel 3. 17 Rancangan Tabel Pengujian Rangkaian Mikrokontroler Secara Umum.

No	Pengujian	Test Case	Hasil yang diharapkan	Hasil pengujian

Tabel 3.18 memiliki 3 kolom dengan penjelasan sebagai berikut.

- a. Pengujian berisi komponen-komponen yang akan diuji dari rangkaian mikrokontroler.
- b. Test Case berisi kasus atau fungsi yang akan diuji dari setiap komponen tersebut
- c. Hasil yang diharapkan berisi parameter keberhasilan dari setiap komponen yang telah ditentukan sesuai dengan kebutuhan sistem.
- d. Hasil pengujian berisi hasil yang didapat saat pengujian dijalankan dari setiap komponen untuk dilakukan analisis apakah hasilnya sesuai dengan yang diharapkan atau tidak.

Setelah dilakukan pengujian pada rangkaian mikrokontroler secara umum, maka selanjutnya dilakukan pengujian terhadap sensor ultrasonik HC-SR04. Pengujian dilakukan dengan melakukan pengukuran terhadap keadaan permukaan status ketinggian air.

Hasil pengukuran permukaan air ini diharapkan menjadi tolak ukur akurasi dari nilai ketinggian air, sehingga bisa dijadikan data yang berguna bagi pihak terkait. Berikut rancangan tabel yang dibuat dalam pengujian akurasi sensor HC-SR04 dalam mengukur jarak pada tiga permukaan air.

Tabel 3. 18 Rancangan Tabel Hasil Pengukuran Sensor pada Permukaan Air Datar.

Sensor 1			Sensor 2			Sensor 3		
Jarak yang diukur manual (cm)	Jarak yang terukur sensor (cm)	% Akurasi (100% - selisih *100%))	Jarak yang diukur manual (cm)	Jarak yang terukur sensor (cm)	% Akurasi (100% - selisih *100%))	Jarak yang diukur manual (cm)	Jarak yang terukur sensor (cm)	% Akurasi (100% - selisih *100%))
Rata-rata akurasi			Rata-rata akurasi			Rata-rata akurasi		

Tabel 3. 19 Rancangan Tabel Hasil Pengukuran Sensor pada Permukaan Air

Wadah	Ukuran Manual Bidang (cm)			Jarak yang Diukur Sensor (cm)		
	Lever 1	Level 2	Level 3	Sensor 1	Sensor 2	Sensor 3
I						
II						

3.4.2. Pengujian Software

Pengujian ini fokus pada aplikasi *website* yang dibuat. Tujuan pengujian untuk mengetahui aplikasi yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan aplikasi tersebut. Pengujian perangkat lunak ini menggunakan pengujian *black box*. *Black box Testing* atau pengujian *black box* merupakan tahap pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu

sesuai dengan proses bisnis yang diinginkan. *Black box testing* berusaha menemukan kesalahan dalam beberapa kategori yaitu :

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan interface.
3. Kesalahan dalam struktur data/akses data.

Untuk memudahkan dalam proses pengujian, maka dibuatlah sebuah skenario pengujian *Software* sebagai berikut.

Tabel 3. 20 Skenario Pengujian Aplikasi Web

No	Kelas Uji	Butir Uji	Jenis Pengujian
1	Halaman Login	Verifikasi User	<i>Black Box</i>
		Verifikasi Password	
2	Halaman Utama/ Monitoring	Menampilkan menu utama	<i>Black Box</i>
3	Halaman Admin	Menampilkan dashboard admin	<i>Black Box</i>

Hasil pengujian akan direkap ke dalam tabel yang memiliki format seperti pada tabel 3.17 sebagai berikut.

Tabel 3. 21 Rancangan Tabel Pengujian Black Box

Kasus / Diuji	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian

Tabel 3.17 terdiri dari 4 kolom dengan penjelasan sebagai berikut.

- a. Kasus / diuji berisi bagian dari aplikasi yang akan diuji.
- b. Jenis pengujian berisi aktivitas-aktivitas aplikasi berupa tindakan dari pengguna ke aplikasi.
- c. Hasil yang diharapkan berisi keterangan output yang seharusnya ditunjukkan aplikasi berdasarkan input yang diberikan oleh pengguna.
- d. Hasil pengujian berisi keterangan “berhasil” apabila output yang diberikan aplikasi sesuai dengan output yang seharusnya, sebaliknya akan berisi keterangan “tidak berhasil” apabila aplikasi memberikan output yang berbeda dengan output yang seharusnya.

3.4 Pembuatan Sistem

Pembuatan perangkat lunak sistem merupakan tahap menerapkan perancangan yang telah dilakukan terhadap sistem sehingga siap untuk dioperasikan. Implementasi sistem meliputi implementasi lingkungan, implementasi data dan implementasi antar muka.

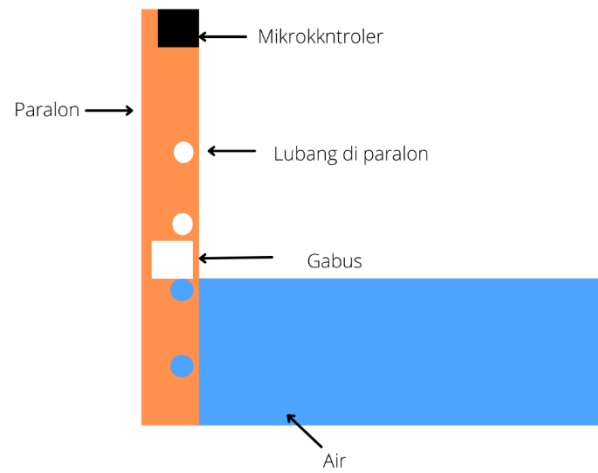
3.5 Pengujian Sistem

Pada tahap ini akan dilakukan pengujian terhadap komponen-komponen yang membangun rangkaian mikrokontroler untuk mengetahui apakah sistem yang berfungsi sebagai seperangkat alat pengukur ketinggian air sudah berjalan sebagaimana mestinya atau tidak. Adapun tahapan pengujian *Hardware* adalah sebagai berikut.

4. Pengujian *Hardware* rangkaian mikrokontroler secara umum.
5. Pengujian sensor HC-SR04 pengukur ketinggian air .
6. Pengujian ESP8266 dalam mengirim data ke *database*.

Untuk memudahkan dalam analisa hasil pengujian, maka setiap hasil dari langkah pengujian akan disampaikan dalam bentuk tabel. Berikut adalah tabel yang dibuat untuk pengujian rangkaian mikrokontroler secara umum.

Desain bentuk mikrokontroler



Gambar 3. 12 Desain bentuk mikrokontroler

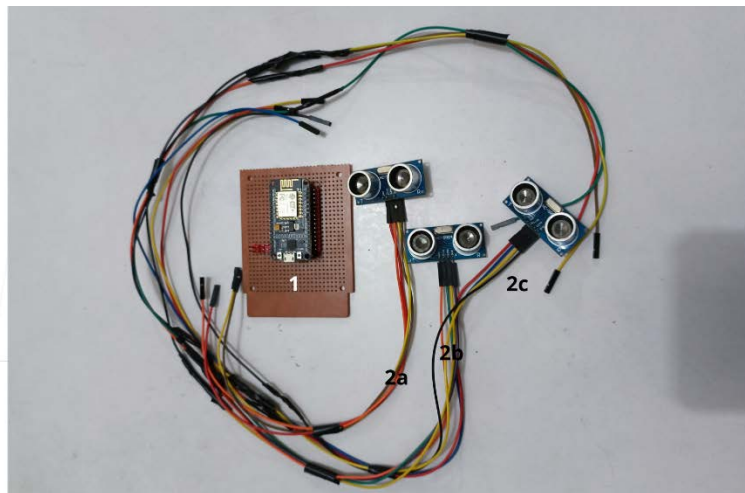
BAB IV HASIL DAN ANALISIS

4.1 Hasil Implementasi

Hasil implementasi dari perancangan yang telah dibuat sebelumnya, diantaranya adalah hasil rangkaian mikrokontroler sistem, hasil implementasi bagian *Hardware* sistem dan hasil implementasi bagian *Software* sistem.

4.1.1. Rangkaian Mikrokontroler Sistem

Hasil rangkaian mikrokontroler sistem yang terdiri dari ESP 8266, Sensor dan RC-SR04 dan warna dapat dilihat pada Gambar 4.1. Rangkaian mikrokontroler ini berfungsi sebagai alat pengukur ketinggian air pada sistem pemantau ketinggian air yang dibangun.



Gambar 4. 1 Rangkaian Mikrokontroler Sistem

Penjelasan Gambar 4.1 dapat dilihat pada Tabel 4.1 berikut.

Tabel 4. 1 Komponen Pembangun Rangkaian Mikrokontroler

Nomor	Nama Komponen	Fungsional
1	Modul ESP 8266	Sistem Kendali Utama
2a	Sensor HC-SR04 -1	Sensor Jarak
2b	Sensor HC-SR04 -2	Sensor Jarak

2c	Sensor HC-SR04 -3	Sensor Jarak
----	-------------------	--------------

4.1.2. Proses Koneksi dan Komunikasi Data *Hardware* Sistem

Proses komunikasi data yang terjadi pada prototipe alat sistem monitoring ketinggian air dan aplikasi *website* sistem adalah sebagai berikut :

1. ESP 8266 terhubung dengan sensor HC-SR04 melalui pin digital I/O pada papan Arduino. Sensor HC-SR04 melalui pin *triger* dan *echo* mendapatkan nilai ketinggian air yang seterusnya bisa diakses oleh ESP 8266. Adapun kode program Bahasa C yang dibuat dengan Arduino IDE untuk mengambil data dari sensor dapat dilihat pada Kode Program 4.1.

Kode Program 4.1 Skrip mengambil data ketinggian air

```

void baca_jarak1() {
  digitalWrite(TRIGGER_PIN1, LOW); // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(TRIGGER_PIN1, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(TRIGGER_PIN1, LOW);
  duration1 = pulseIn(ECHO_PIN1, HIGH);
  distance1 = (duration1 / 2) / 29.1;
  Serial.print(distance1);
  Serial.println(" cm");
}

void baca_jarak2() {
  digitalWrite(TRIGGER_PIN2, LOW); // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(TRIGGER_PIN2, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(TRIGGER_PIN2, LOW);
  duration2 = pulseIn(ECHO_PIN2, HIGH);
  distance2 = (duration2 / 2) / 29.1;
  Serial.print(distance2);
  Serial.println(" cm");
}

```

2. Berdasarkan Kode Program 4.1, hasil pembacaan jarak atau yang dikalikan sebagai ketinggian permukaan air dari sensor HC-SR04 dimuat dalam bentuk Integer, kemudian data tersebut digabungkan ke dalam variabel “text” yang bertipe String. Hasil baca sensor bisa dilihat secara *real time* pada aplikasi Arduino IDE dengan fitur *serial monitor*, Gambar 4.2 adalah hasil baca sensor yang ditampilkan dalam *serial monitor*.

Gambar 4.2 Hasil Baca Sensor pada Serial Monitor Arduino IDE

```

<hr>
<address>Apache/2.4.46 (Win64) OpenSSL/1.1.1lg PHP/7.4.10 Server at 192.168.43.2 Port 80</address>
</body></html>

closing connection

baca jarak Jarak 1 : 14 cm  Jarak 2 : 15 cm  Jarak 3 : 15 cm
connecting to 192.168.43.2
Requesting URL: /belajar/index.php/simpan/sensor?data=14&data2=15&data3=15
HTTP/1.1 404 Not Found
Date: Wed, 01 Jun 2022 09:40:42 GMT
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1lg PHP/7.4.10
Content-Length: 299
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.46 (Win64) OpenSSL/1.1.1lg PHP/7.4.10 Server at 192.168.43.2 Port 80</address>
</body></html>

closing connection

baca jarak Jarak 1 : 14 cm  Jarak 2 : 15 cm  Jarak 3 : 15 cm
connecting to 192.168.43.2
Requesting URL: /belajar/index.php/simpan/sensor?data=14&data2=15&data3=15
HTTP/1.1 404 Not Found
Date: Wed, 01 Jun 2022 09:40:53 GMT
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1lg PHP/7.4.10
Content-Length: 299
Connection: close
Content-Type: text/html; charset=iso-8859-1

 Autoscrol   Show timestamp

```

Gambar 4. 2 Hasil Baca Sensor pada Serial Monitor Arduino IDE

Pada saat yang bersamaan, yang telah terhubung dalam rangkaian mikrokontroler sistem akan melakukan koneksi pada jaringan internet yang telah ditentukan, sebagaimana dapat dilihat pada potongan Kode Program 4.2 yang ditanam dalam Modul tersebut.

Kode Program 4. 2 Skrip Koneksi ke Jaringan Internet pada .

```

#include <ESP8266WiFi.h>

#define TRIGGER_PIN1 16 //D1
#define ECHO_PIN1 5 //D2

#define TRIGGER_PIN2 4 //D1
#define ECHO_PIN2 14 //D2

#define TRIGGER_PIN3 12 //D1
#define ECHO_PIN3 13 //D2

const char* ssid = "note7";
const char* password = "satudiatige";

const char* host = "192.168.43.2";

WiFiClient client;
const int httpPort = 80;
String url;
long duration1, distance1, duration2, distance2, duration3, distance3;
unsigned long timeout;

```

Kode Program 4.2 di atas akan membuat terhubung dengan jaringan internet

yang telah ditentukan seperti pada kode program diatas di atas, kemudian berfungsi dalam proses *upload* data ke *Database Server* sistem.

3. Nilai ketinggian air yang didapatkan dari Sensor RC 04 dikirim ke ESP 8266 yang saling terhubung menggunakan komunikasi serial. Disini fungsi ESP 8266 adalah sebagai pengirim data menggunakan pin Transmitter (TX) dan sebagai penerima data menggunakan pin I/O sebagai mana bisa dilihat pada potongan Kode Program 4.3. dan 4.4.

Kode Program 4. 3 Skrip Komunikasi Serial pada .

```
void baca_jarak1(){
  digitalWrite(TRIGGER_PIN1, LOW); // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(TRIGGER_PIN1, HIGH);
  delayMicroseconds(10); // Added this line
  digitalWrite(TRIGGER_PIN1, LOW);
  duration1 = pulseIn(ECHO_PIN1, HIGH);
  distancel = (duration1/2) / 29.1;
  Serial.print("Jarak 1 : ");
  Serial.print(distancel);
  Serial.print(" cm");
  Serial.print(" ");
}
}
```

4. Berdasarkan Kode Program 4.3 di atas, terhubung dengan ESP 8266 menggunakan pin I/O yaitu pin D3 sebagai receiver (RX).

Kode Program 4. 4 Skrip Sebagai Penerima Data

```
void loop() {
  Serial.print("baca jarak ");
  baca_jarak1();
  baca_jarak2();
  baca_jarak3();
  Serial.println("  ");
  |
  Serial.print("connecting to ");
  Serial.println(host);
}
```

Pada data tersebut dimasukkan dalam Variabel “Data” dengan tipe data *Array* yang masing-masing diinisialkan dengan Variabel “A,B,C” yang akan dikirim ke *server* menggunakan metode POST, sebagaimana prosesnya dapat dilihat pada potongan Kode Program 4.5. dan 4.6.

Kode Program 4. 5 Skrip Setting IP address atau Web Server Tujuan.

```
const char* ssid    = "note7";
const char* password = "satuduatige";

const char* host = "192.168.43.2";
```

Kode Program 4. 6 Skrip Upload Variabel “Data” ke Server.

```
if (!client.connect(host, httpPort)) {
  Serial.println("connection failed");
  //return;
}

url = "/belajar/index.php/simpan/sensor?data="+ String(distance1) +"&data2="+ String(distance2) +"&data3="+String(distance3);

Serial.print("Requesting URL: ");
Serial.println(url);

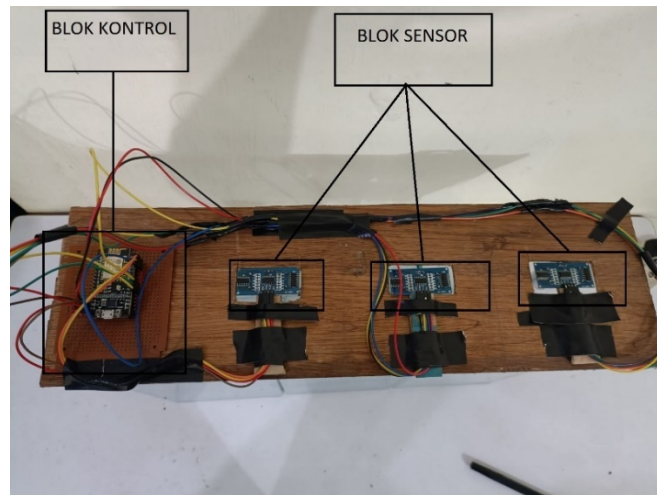
// This will send the request to the server
client.print(String("GET ") + url + " HTTP/1.1\r\n" +
             "Host: " + host + "\r\n" +
             "Connection: close\r\n\r\n");
timeout = millis();
while (client.available() == 0) {
  if (millis() - timeout > 5000) {
    Serial.println(">>> Client Timeout !");
    client.stop();
    return;
  }
}

// Read all the lines of the reply from server and print them to Serial
while(client.available()){
  String line = client.readStringUntil('\r');
  Serial.print(line);
}
```

Kode Program 4.5 merupakan potongan kode program di yang berfungsi untuk membuat koneksi ke Web Server tujuan dan Kode Program 4.6 berfungsi untuk meng-*upload* data ketinggian air ke *server*.

4.1.3. Prototipe Alat Monitoring Ketinggian Air

Desain rangkaian mikrokontroler sistem yang telah diimplementasikan kedalam alat ukur ketinggian air dan prototipe *box* air dapat dilihat pada Gambar 4.3. Prototipe wadah air sebagai media pengujian terbuat dari *box* wadah berbahan plastik berukuran 17 x 12 x 12 cm yang diasumsikan sebagai miniatur dari wadah air pada .

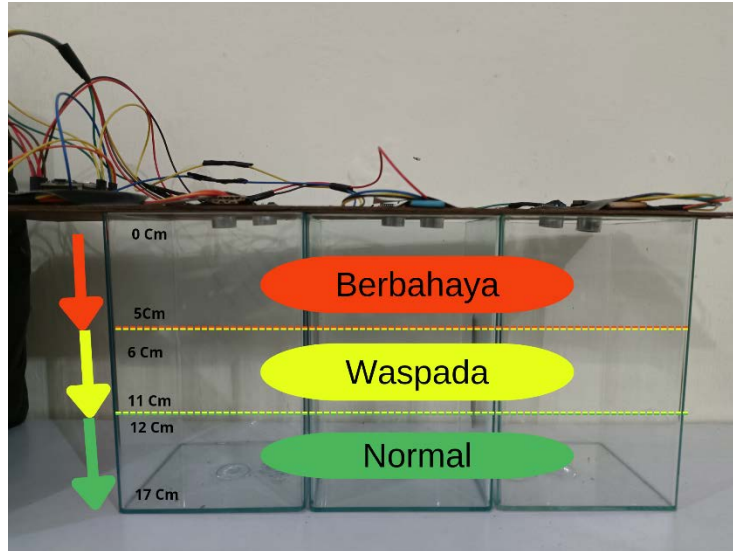


(a)

(b)
(a) Prototipe Alat Pengukur Ketinggian Air dan (b) Prototipe Wadah Air

Gambar 4. 3 Hasil Implementasi Hardware Sistem Keseluruhan

Bagian perangkat keras sistem yang terdiri dari alat pengukur ketinggian air dan *box* wadah air, akan menjadi bagian yang tidak terpisahkan. Desain alat yang dibuat menyesuaikan kondisi *box* wadah air agar dapat bekerja secara maksimal. Ukuran *box* wadah yang dijadikan media untuk menampung air sangat berpengaruh pada tata letak sensor pada alat pengukur ketinggian air tersebut. Untuk melihat hasil implementasi alat dan *box* wadah secara keseluruhan, tampak pada Gambar 4.4 berikut.



Gambar 4. 4 Hasil Implementasi Hardware Sistem Secara Keseluruhan

Gambar 4.4 Hasil Implementasi *Hardware* Sistem Secara Keseluruhan

Berdasarkan Gambar 4.4, menunjukkan bahwa jarak antara sensor dengan dasar wadah adalah 17 cm. Jarak 0-5 cm adalah indikator berbahaya, jarak 6-11 cm adalah indikator waspada dan jarak 12-17 cm adalah indikator normal . Bagian-bagian dari Gambar 4.4 di atas dijelaskan dalam Tabel 4.2 berikut.

Tabel 4. 2 Komponen-komponen Pembangun Prototipe *Hardware* Sistem

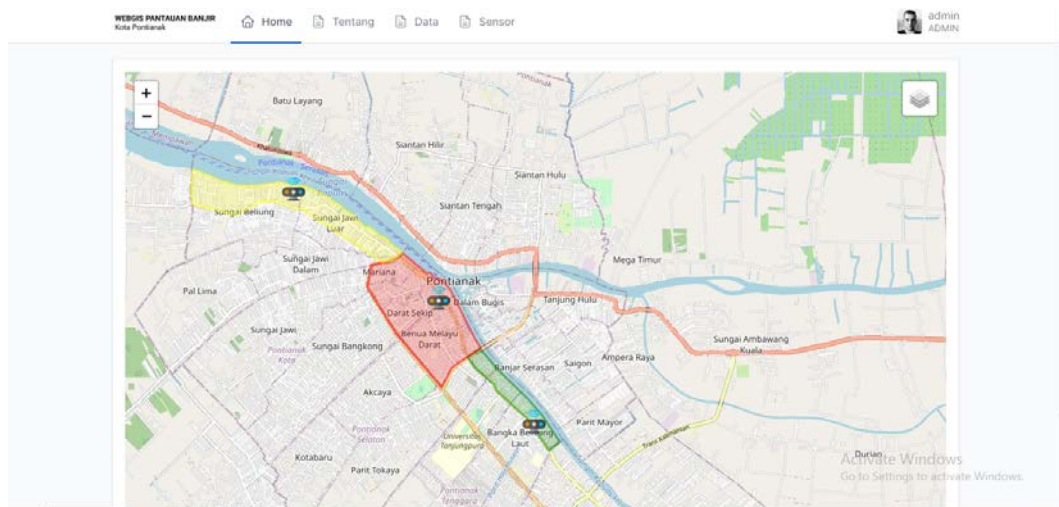
Nomor	Nama Komponen	Fungsional
1	Sensor HC-SR04 -1	Sensor Blok 01
2	Sensor HC-SR04 -2	Sensor Blok 02
3	Sensor HC-SR04 -3	Sensor Blok 03
4	Rangkaian Alat Ukur	Sistem Kendali Utama
5	Akuarium	Wadah Air

4.1.4. Hasil Implementasi *Software* Sistem

4.1.4.1. Halaman Utama *Website* Sistem

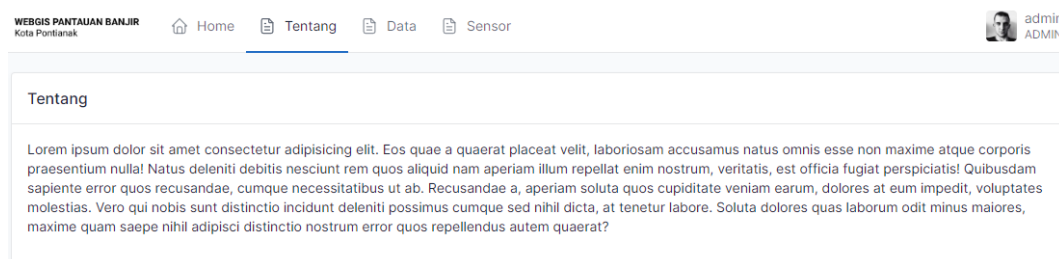
Halaman utama *website* sistem merupakan tampilan yang pertama kali muncul ketika mengakses *webiste* sistem *monitoring* ketinggian air atau yang

dinamai dengan MONITORING BANJIR. Tampilan halaman utama *website* sistem dapat dilihat pada Gambar 4.5.

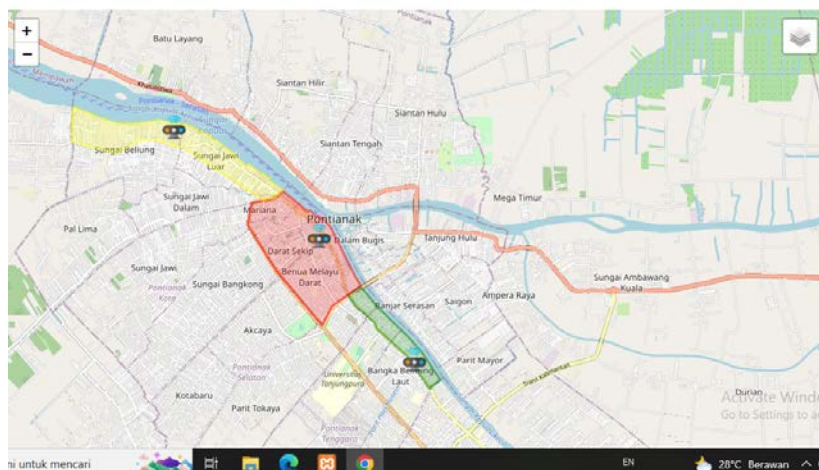


Gambar 4. 5 Tampilan Halaman Utama Website

Berdasarkan Gambar 4.5, pada halaman utama *webiste* terdapat dua tombol, yaitu tombol tentang aplikasi yang menyajikan informasi terkait pengembang aplikasi beserta kegunaan sistem secara umum dan kemudian tombol lihat peta untuk mengakses halaman peta *monitoring* ketinggian air, sebagaimana terlihat pada Gambar 4.6 dan Gambar 4.7.



Gambar 4. 6 Tampilan Halaman Tentang Aplikasi



Gambar 4. 7 Tampilan Halaman Peta *Monitoring*

Gambar 4.7 di atas memberikan informasi ketinggian air pada wadah air , baik pada kondisi penuh, hampir penuh maupun belum penuh. Informasi tersebut disajikan dalam bentuk peta.

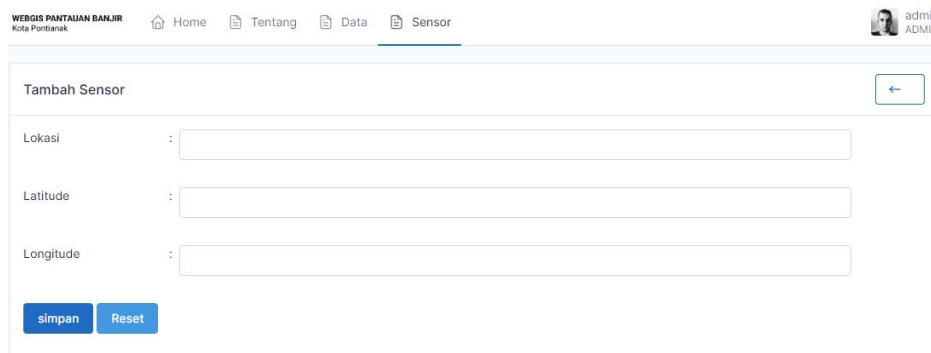
4.1.4.2. Halaman *Login Admin* Sistem

Halaman *login website* memuat *form* yang berisi *username* dan *password* yang harus diisi oleh pengguna yang berstatus *admin* agar dapat masuk ke halaman utama *website* sistem adapun pengguna biasa tidak bisa akses form login. Tampilan halaman *login website* dapat dilihat pada Gambar 4.8.

Gambar 4. 8 Halaman *Login Website* Sistem

4.1.4.3. Halaman Sensor Sistem

Halaman *dashboard admin* merupakan halaman *monitoring* secara detail yang dilakukan oleh *admin*. Ketika mengakses halaman ini yang pertama kali ditampilkan adalah nilai persentase ketinggian air dengan 3 indikator ketinggian yang sudah ditentukan sebelumnya pada setiap sensor dan tombol untuk menambahkan sensor baru pada sistem. Pada setiap menu sensor dilengkapi dengan fitur lihat grafik, manajemen sensor dan hapus sensor. Tampilan halaman

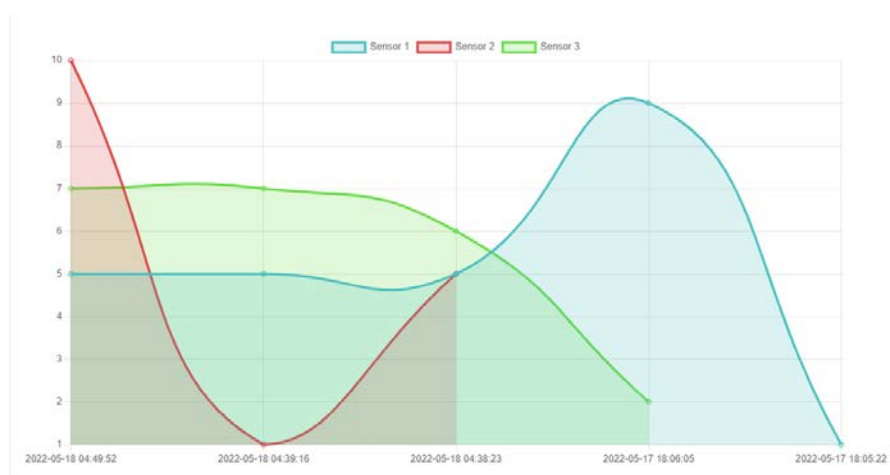


The screenshot shows a web application interface for adding a sensor. At the top left, the logo reads 'WEGIS PANTAUAN BANJIR Kota Pontianak'. The navigation menu includes 'Home', 'Tentang', 'Data', and 'Sensor'. The user profile 'admin ADMIN' is visible in the top right. The main content area is titled 'Tambah Sensor' and contains three input fields: 'Lokasi', 'Latitude', and 'Longitude'. Below these fields are two buttons: 'simpan' and 'Reset'. A back arrow button is located in the top right corner of the form area.

Gambar 4.9 *Form* Tambah Sensor

4.1.4.3.1. Menu Lihat Grafik

Pada menu ini ditampilkan hasil baca dari 3 buah sensor yang terpasang pada wadah. Data disajikan dalam bentuk grafik garis per hari dimulai dari pukul 18.00 - 04.00 WIB. Sumbu x (mendatar) dari grafik merupakan variabel dan sumbu y (tegak) dari grafik merupakan variabel nilai perubahan ketinggian air. Tampilan grafik hasil baca sensor pada setiap wadah dapat dilihat pada Gambar 4.11.



Gambar 4. 10 Grafik Hasil Baca Sensor

4.1.4.3.2. Menu Edit Sensor

Pada menu ini disajikan *form* ubah data wadah yang terdiri dari Nama, Alamat, *Latitude*, *Longitude*. Tampilan *form edit sensor* dapat dilihat pada Gambar 4.12.

WEBSITE PANTAIAN BANJIR
Kota Pontianak

Home Tentang Data Sensor

adminin
ADMIN

Ubah Sensor

Lokasi : Sul Raja Dalam

Latitude : -0.0538576

Longitude : 109.38261239

simpan Reset

Copyright © 2022

Gambar 4. 11 Form Ubah Data

4.1.4.3.3. Menu Manajemen Sensor

Pada menu ini disajikan beberapa sensor yang terpasang pada sebuah wadah. Disertai dengan tombol tambah sensor, tombol edit sensor dan tombol hapus sensor. Tampilan menu manajemen sensor dapat dilihat pada Gambar 4.13.

WEGIS PANTAUAN BANJIR
Kota Pontianak

Home Tentang Data Sensor

admin ADMIN

Ubah Sensor

Lokasi : Sui Raya Dalam

Latitude : -0.0536576

Longitude : 109.36261239

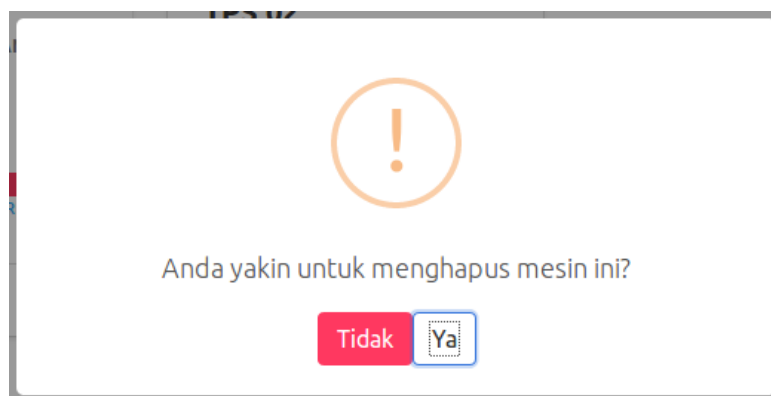
simpan Reset

Copyright © 2022

Gambar 4. 12 Form Ubah Data Sensor

4.1.4.3.4. Menu Hapus Sensor

Semua Sensor yang telah ditambahkan selain dapat diubah, juga bisa dihapus dari data Sensor. Sistem akan memberikan peringatan sebelum pengguna memilih untuk menghapus sebuah Sensor. Terlihat seperti Gambar 4.14 dimana ketika ditekan tombol hapus Sensor maka muncul peringatan sebagai berikut.



Gambar 4. 13 Form Hapus Sensor

4.2 Pengujian

Proses pengujian sistem meliputi pengujian pada sisi alat (*Hardware*), aplikasi (*Software*) dan pengujian hasil *monitoring* sistem. Pengujian pada sisi *Hardware* dilakukan dengan mengoperasikan alat sebagaimana mestinya, hal ini dilakukan untuk mengetahui apakah perangkat alat dapat berfungsi dengan baik atau tidak. Sedangkan pengujian pada sisi *Software* dilakukan dengan *blackbox testing*.

4.2.1. Pengujian *Software*

4.2.1.1. Black Box Testing

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak yang dibangun. Adapun skenario pengujian *black box testing* dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Skenario Pengujian Black Box Testing

No	Kelas Uji	Butir Uji	Jenis Pengujian
1	Halaman Pembuka dan peta <i>monitoring</i>	Menampilkan halaman pembuka dan menu peta <i>monitoring</i>	<i>Black Box</i>
2	Form Login	Verifikasi User	<i>Black Box</i>
		Verifikasi Password	
4	Form Tambah Sensor	Menambahkan data Sensor	<i>Black Box</i>
5	Form Ubah Sensor	Mengubah data Sensor yang telah ada sebelumnya	<i>Black Box</i>
6	Halaman Manajemen Sensor	Menambah, mengubah dan menghapus data sensor	<i>Black Box</i>

Berdasarkan skenario pengujian yang disusun, maka dilakukan pengujian sebagai berikut :

1. Pengujian Halaman Pembuka dan Peta *Monitoring*

Tabel 4. 4 Pengujian Halaman Pembuka dan Peta Monitoring

Kasus / Diuji	Skenari Uji	Hasil Yang Diharapkan	Hasil pengujian
Halaman Pembuka dan Peta Monitoring	Melihat ulasan ringkas mengenai aplikasi dan informasi ketinggian air di	Menampilkan ulasan ringkas aplikasi dan peta <i>monitoring</i>	Berhasil

2. Pengujian Halaman Login

Tabel 4. 5 Pengujian Halaman Login

Kasus / Diuji	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
Halaman Login	Memasukan Username dan Password dengan benar	Masuk ke halaman <i>Utama</i>	Berhasil
	Memasukan Username dan Password dengan salah	Muncul notifikasi kesalahan input data	Berhasil

3. Pengujian Halaman *Admin***Tabel 4. 6** Pengujian Halaman *Admin*

Kasus / Diuji	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
Halaman <i>Admin</i>	Masuk ke halaman <i>admin</i> dengan akun <i>login admin</i>	Menampilkan dashboard <i>admin</i>	Berhasil

4. Pengujian Form Tambah Sensor

Tabel 4. 7 Pengujian Form Tambah Sensor

Kasus / Diuji	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
Form Tambah Sensor	Memasukan data baru Sensor	Menyimpan data ke <i>database</i> server	Berhasil

5. Pengujian Form Ubah Sensor

Tabel 4. 8 Pengujian Form Ubah Sensor

Kasus/ Diuji	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
Form Ubah Sensor	Mengubah data Sensor yang sudah ada	<i>Update</i> data Sensor di <i>database</i> server	Berhasil

6. Pengujian Form Manajemen Sensor

Tabel 4.14 Pengujian Form Manajemen Sensor

Kasus / Diuji	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
Form Manajemen Sensor	Menambah, mengubah dan menghapus data Sensor	Create data, update data dan delete data di database	Berhasil

4.2.2. Pengujian *Hardware***4.2.2.1. Pengujian Rangkaian Mikrokontroler**

Pengujian *Hardware* rangkaian mikrokontroler merupakan pengujian yang dilakukan pada perangkat mikrokontroler dengan berbagai perlakuan yang memungkinkan. Berikut adalah beberapa pengujian yang dilakukan pada sisi rangkaian mikrokontroler. Hasil yang diperoleh dapat dilihat pada Tabel 4.3. Gambar hasil pengujian dapat dilihat pada lampiran A.

Tabel 4.3 Hasil Pengujian Rangkaian Mikrokontroler Secara Umum.

No	Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
1	ESP 8266	<i>Upload sketch</i> program	<i>Upload Done</i>	Berhasil
2	ESP 8266	Komunikasi serial ke	Kirim data ketinggian ke	Berhasil
3	Sensor HC-SR04	Scan jarak	Dapat menghitung jarak	Berhasil
4	ESP 8266	Menghubungkan ke jaringan wifi yang ditentukan	Terhubung ke jaringan wifi	Berhasil

No	Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
5	ESP 8266	Memperoleh data sensor dari Arduino	Menampilkan data sensor sesuai dengan data pada Arduino	Berhasil
6	ESP 8266	<i>Upload</i> data hasil sensor ke server	Data masuk ke <i>database</i> server	Berhasil

Berdasarkan hasil pengujian, dapat disimpulkan beberapa hal diantaranya adalah

1. ESP 8266 dapat terhubung ke dengan komunikasi serial.
2. Sensor HC-SR04 bisa melakukan scan jarak terhadap objek.
3. ESP 8266 Bisa dinyalakan dengan program yang ditanamkan pada ESP 8266.
4. Waktu rata-rata yang diperlukan untuk terhubung ke jaringan wifi adalah 5 detik.

4.2.2.2. Pengujian Sensor HC-SR04 Sebagai Pengukur Ketinggian Air

Pengujian sensor HC-SR04 dilakukan pada sebuah miniatur wadah yang mempunyai ketinggian 17 cm. Sensor HC-SR04 yang sudah terhubung dengan ESP 8266, ditempatkan 1 cm di atas wadah sehingga jarak sensor ke dasar wadah adalah 16 cm. Sedangkan Arduino dihubungkan ke laptop melalui kabel USB untuk melihat hasil pengukuran sensor di serial monitor pada aplikasi Arduino IDE. Kemudian wadah diisi air dengan 3 keadaan, yaitu :

1. Permukaan wadah terisi
2. Permukaan wadah setengah isi
3. Permukaan wadah penuh

Dari ketiga keadaan permukaan air di atas, dapat dijadikan tolak ukur akurasi pengukuran sensor terhadap ketinggian timbulan air. Hasil pengukuran dengan sensor tersebut akan dibandingkan dengan pengukuran secara manual menggunakan penggaris. Dalam pengujian ini, jarak ukur manual antara sensor HC-SR04 dengan dasar wadah adalah 17 cm. Proses pengujian sensor HC-SR04 ini dapat dilihat pada Gambar 4.18.

Gambar 4. 14 Pengujian HC-SR04 pada Tiga Permukaan Air

Pengujian HC-SR04 pada Tiga Permukaan Air

Hasil pengujian sensor HC-SR04 sebagai pengukur ketinggian timbulan air dapat dilihat pada tabel 4.4 sampai dengan tabel 4.6 berikut.

Tabel 4.4 Hasil Pengujian Sensor HC-SR04 pada Permukaan Air Datar.

Hasil Pengujian Sensor HC-SR04								
Sensor 1			Sensor 2			Sensor 3		
Jarak yang diukur manual (cm)	Jarak yang terukur sensor (cm)	% Akurasi (100% - selisih *100%)	Jarak yang diukur manual (cm)	Jarak yang terukur sensor (cm)	% Akurasi (100% - selisih *100%)	Jarak yang diukur manual (cm)	Jarak yang terukur sensor (cm)	% Akurasi (100% - selisih *100%)
17	16	99	17	16	99	17	17	99
11	11	100	11	11	100	11	11	100
10	9	99	10	9	99	10	9	99
Rata-rata akurasi		99,33	Rata-rata akurasi		99,33	Rata-rata akurasi		99,33

Dari Tabel 4.4, nilai pengukuran manual memiliki perbedaan dengan pengukuran

sensor, yaitu memiliki selisih 1 cm, dengan rata-rata akurasi 99,33 %/

4.2.2.3. Pengujian Dalam Mengirim Data ke *Database*

Pada tahap ini dilakukan pengujian pengiriman data dari Arduino ke *database* menggunakan. Pengujian data ke *database* dilakukan untuk mengetahui apakah dapat terhubung dengan jaringan internet. Pengujian dilakukan dengan mengirim data ke *database* dengan metode *Post* data melalui program yang ditanam pada *board*. Pengujian ini dilakukan sebanyak sepuluh kali di mana data yang dikirim ke *database* merupakan nilai hasil baca tiga buah sensor yang ada pada wadah 1, wadah 2 dan wadah 3, seperti yang ditunjukkan pada Gambar 4.19.

+ Opsi		Id_data	Id_sensor	Ketinggian	Tanggal
<input type="checkbox"/>	Ubah Salin Hapus	1	1	1	2022-03-26 21:48:14
<input type="checkbox"/>	Ubah Salin Hapus	2	2	5	2022-03-26 15:48:14
<input type="checkbox"/>	Ubah Salin Hapus	3	3	2	2022-03-26 15:48:14
<input type="checkbox"/>	Ubah Salin Hapus	4	2	1	2022-03-28 21:46:04
<input type="checkbox"/>	Ubah Salin Hapus	5	1	4	2022-03-28 21:46:04
<input type="checkbox"/>	Ubah Salin Hapus	6	3	6	2022-03-28 16:46:03
<input type="checkbox"/>	Ubah Salin Hapus	7	1	1	2022-05-17 18:05:22
<input type="checkbox"/>	Ubah Salin Hapus	8	1	9	2022-05-17 18:06:05
<input type="checkbox"/>	Ubah Salin Hapus	9	1	5	2022-05-18 04:38:23
<input type="checkbox"/>	Ubah Salin Hapus	10	1	5	2022-05-18 04:39:16
<input type="checkbox"/>	Ubah Salin Hapus	11	3	7	2022-05-18 04:47:32
<input type="checkbox"/>	Ubah Salin Hapus	12	1	5	2022-05-18 04:49:52
<input type="checkbox"/>	Ubah Salin Hapus	13	2	10	2022-05-18 04:49:52
<input type="checkbox"/>	Ubah Salin Hapus	14	3	7	2022-05-18 04:49:52

Gambar 4.19 Hasil Baca Sensor di *Database*

Gambar 4.19 merupakan *screenshot* dari tabel hasil baca masing-masing sensor dari ketiga wadah yang diujikan. Data hasil baca sensor yang diterima secara *real time* dapat dilihat di *serial monitor* pada aplikasi Arduino IDE, sebagaimana terlihat pada Gambar 4.20.

```

<hr>
<address>Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.10 Server at 192.168.43.2 Port 80</address>
</body></html>

closing connection

baca jarak Jarak 1 : 14 cm  Jarak 2 : 15 cm  Jarak 3 : 15 cm
connecting to 192.168.43.2
Requesting URL: /belajar/index.php/simpan/sensor?data=14&data2=15&data3=15
HTTP/1.1 404 Not Found
Date: Wed, 01 Jun 2022 09:40:42 GMT
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.10
Content-Length: 299
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.10 Server at 192.168.43.2 Port 80</address>
</body></html>

closing connection

baca jarak Jarak 1 : 14 cm  Jarak 2 : 15 cm  Jarak 3 : 15 cm
connecting to 192.168.43.2
Requesting URL: /belajar/index.php/simpan/sensor?data=14&data2=15&data3=15
HTTP/1.1 404 Not Found
Date: Wed, 01 Jun 2022 09:40:53 GMT
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.10
Content-Length: 299
Connection: close
Content-Type: text/html; charset=iso-8859-1

```

Autoscroll Show timestamp

Gambar 4.20 Hasil Baca Sensor pada Serial Monitor

Dari Gambar 4.20 dapat dilihat bahwa nilai hasil baca ketinggian air dari ketiga sensor beserta waktunya. Nilai tersebut dipisah dengan karakter bintang (*) dimulai dari sebelah kiri adalah sensor 1 dan seterusnya, kemudian diakhiri dengan karakter pagar (#).

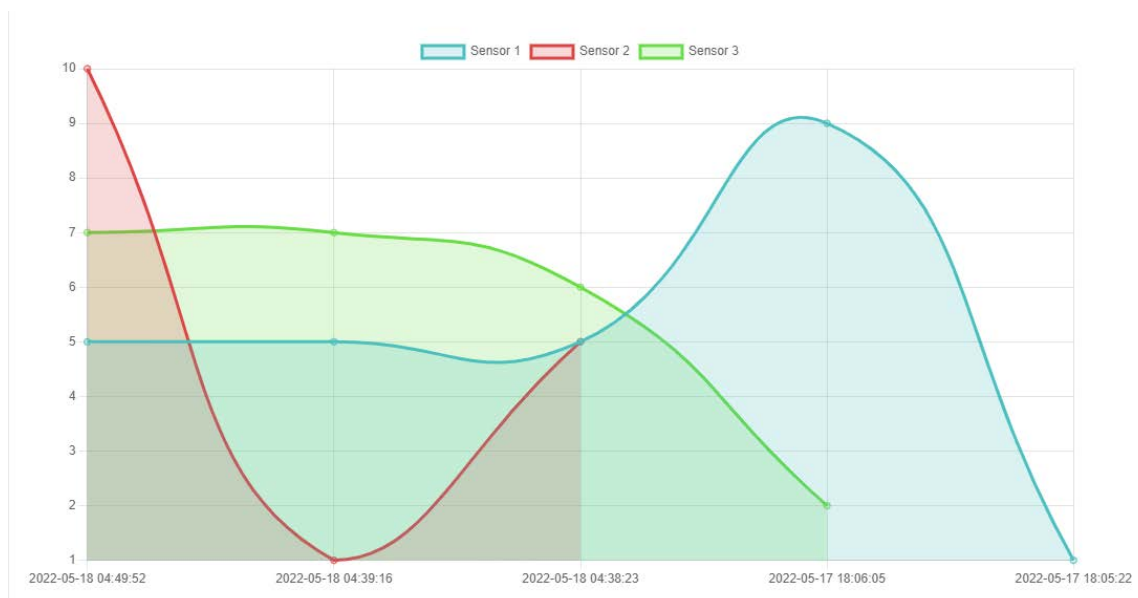
Dalam pengujian mengirim data dari ke *database server*, dilakukan sepuluh kali percobaan pengiriman data, dan semuanya berhasil. Dalam pengujian ini dihitung selisih antara waktu pada *serial monitor* dengan waktu pada *database server*. Selisih nilai tersebut merupakan lama waktu yang dibutuhkan oleh untuk mengirim data ke *database server*. Lama waktu yang dibutuhkan dalam pengiriman data ke *database server*, rata-rata 5 detik. Ini menunjukkan bahwa waktu yang tertera pada *database server* masih bisa dianggap sebagai kondisi *real time* dari ketinggian air di dalam wadah. Data tersebut akan diolah oleh aplikasi *website* untuk kemudian ditampilkan pada antarmuka aplikasi untuk diakses oleh pengguna. Hasil pengujian ini bisa dilihat pada tabel 4.7 berikut.

Tabel 4.7 Hasil Pengujian Kirim Data dari ESP 8266 ke *Database*

Pengiriman ke-	Berhasil (Ya / Tidak)	Waktu pada <i>serial monitor</i>	Waktu pada <i>database server</i>	Waktu yang dibutuhkan (detik)
1	Ya	10:12:44	10:12:47	3
2	Ya	10:12:58	10:12:59	1
3	Ya	10:13:11	10:13:21	10
4	Ya	10:13:23	10:13:24	1
5	Ya	10:13:30	10:13:46	16
6	Ya	10:13:36	10:13:49	12
7	Ya	10:14:01	10:14:11	10
8	Ya	10:14:13	10:14:14	1
9	Ya	10:14:26.	10:14:36	10
10	Ya	10:14:38	10:14:39	1
Rata-rata waktu yang dibutuhkan (detik)				5

4.3 Analisis Fungsi *Monitoring* Sistem

Pada tahap ini akan diuji cara kerja sistem secara keseluruhan dalam melakukan *monitoring* ketinggian air pada Sensor. Hasil *monitoring* sistem yang ditampilkan dalam bentuk grafik dapat dilihat pada Gambar 4.21 berikut.



Gambar 4.21 Grafik Hasil *Monitoring* Sistem

Berdasarkan Gambar 4.21, dapat dilihat bahwa waktu yang ditentukan oleh sistem dalam melakukan *monitoring* adalah dari pukul 00.00 - 23.59 WIB, di pagi hari pada pukul 18.06 WIB pada sensor 1 terlihat ada kenaikan tinggi air diketinggian 9 dan turun kembali pada jam 04.38 WIB di ketinggian 5. terjadi penurunan pada jam 04.39 WIB sampai di titik 1 dan peningkatan grafik pada sensor 2 pada jam 04.49 WIB di ketinggian 10. Pada sensor 3 terjadi peningkatan sampai ketinggian 7 pada jam 04.39 WIB dan stabil sampai jam 04.49 WIB. Ketinggian air dan turun di malam hari pada pukul 18.00 - 04.00 WIB terjadi kenaikan ketinggian air, ini menunjukkan sistem akan *standby* selama 24 jam.. Adapun *monitoring* yang tetap berjalan akan mendata perubahan ketinggian air yang menunjukkan adanya pasang air . Data tersebut menjadi nilai statistik disaat pasang air laut, sekaligus menunjukkan tingkat kepahaman masyarakat sekitar dalam usaha pengelolaan peringatan banjir air yang dilakukan oleh Pemerintah Daerah. Sehingga bisa dianalisis untuk diambil langkah edukasi oleh pihak pengelola kepada masyarakat tersebut.

4.4 Analisis Hasil Pengujian Secara Keseluruhan

Berdasarkan pengujian-pengujian yang telah dilakukan, maka analisis hasil yang dapat diambil diantaranya adalah :

1. Terdapat perbedaan nilai antara pengukuran manual dengan pengukuran menggunakan sensor, yaitu sebesar 1 cm, dengan persentasi rata-rata akurasi sebesar 99,33 %.
2. Waktu yang dibutuhkan oleh rangkaian mikrokontroler dari mulai dihidupkan sampai stabil rata-rata adalah 25 detik.
3. Waktu rata-rata yang dibutuhkan untuk *upload* data ke *server* adalah 5 detik.
4. Semakin sedikit jeda waktu antara pengambilan data dari sensor sampai *upload* data ke *server* maka akan semakin baik karena datanya akan bersifat lebih *real time*.
5. Fungsi *monitoring* yang dilakukan, menunjukkan halaman grafik air sistem monitoring banjir, pada halaman ini masyarakat dapat melihat grafik kenaikan air di , data grafik ini diambil dari alat yang dipasang di bantaran yang akan dikirim ke database dan ditampilkan datanya di website tersebut secara *real time*.
6. Adapun bagi pihak pengelola, informasi ketinggian air yang didapatkan akan menjadi nilai awal dalam menentukan kebijakan untuk melakukan kontrol terkait tinggi air juga memberikan pemberitahuan kepada masyarakat bahwa terjadi peringatan maupun berbahaya di daerah pintu air ,seperti banjir rob,banjir bandang,banjir ceulanang,dll. Selain itu sistem juga bisa memberikan nilai ketinggian air pada waktu pasang air maupun pada waktu yang lain.

BAB V **HASIL DAN ANALISIS**

5.1 Kesimpulan

Berdasarkan hasil analisis dan pengujian terhadap Visualisasi banjir secara spasial berdasarkan Dataj sensor ketinggian Berbasis Mikrokontroller Arduino(*Prototype*), dapat disimpulkan bahwa :

1. Pembangunan aplikasi “Visualisasi Banjir Secara Spasial Berdasarkan Data Sensor Ketinggian Berbasis Mikrokontroler Arduino(*Prototype*)” telah selesai dilakukan, hal tersebut dibuktikan dengan aplikasi sudah mampu memberikan informasi ketinggian permukaan air.
2. Hasil dari penelitian dapat disajikan dalam bentuk poligon, tabel dan grafik. Hal tersebut dapat memudahkan pihak terkait dalam mendapatkan informasi dan mengambil keputusan,
3. Berdasarkan hasil pengujian aplikasi, perbandingan antara *pengukuran* manual dengan pengukuran menggunakan sensor, rata-rata mendapatkan nilai akurasi yang sama, yaitu hasil pengukuran sensor memiliki selisih 1 cm dari nilai hasil pengukuran manual yang menunjukkan tingkat akurasi 99 %.

5.2 Saran

Hal-hal yang menjadi saran dalam pengembangan sistem ini adalah :

1. Tambahkan fitur SMS peringatan jika terjadinya banjir
2. Tambahkan fitur penentuan rute tercepat dan menghindari saat evakuasi ketika terjadinya banjir , sehingga dapat menghindari terjadinya korban jiwa dalam terjadi bencana banjir.
3. Sistem ini dapat dipublikasikan secara *Online* agar dapat diakses oleh masyarakat sebagai sebuah media informasi.
4. Aplikasi dapat dikembangkan untuk aplikasi berbasis *mobile*. Sehingga lebih cepat mendapatkan informasi masalah banjir,juga menarik dari segi visual dan performa yang lebih cepat.

DAFTAR PUSTAKA

- Khairul. 2020 “Implementasi Pengendali Sistem Buka Tutup Pintu Air Otomatis Berbasis Arduino Uno R3 Dan Website”
- Furqon, Achmad, Prasetijo, Agung Budi, Prasetijo, Agung Budi. 2019. “Rancang Bangun Sistem Monitoring dan Kendali Daya Listrik pada Rumah Kos Menggunakan NodeMCU dan Firebase Berbasis Android” dalam *Techné Jurnal Ilmiah Elektroteknika* Vol. 18 No. 2 April 2019 Hal 93 -104
- Chandra, Eggi. Sholva, Yus. Muhardi, Hafiz.2020. ”Perancangan Sistem Pemantau Ketinggian Sampah Menggunakan Mikrokontroler Arduino dan Aplikasi Berbasis Web” dalam *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*Vol. 8 , No. 1 , Januari 2020. Pontianak:
- Agus, S. (2011). *Trik Kolaborasi Codeigniter dan JQuery*. Yogyakarta: Lokomedia.
- Ellislab. (2020, Desember 14). Codeigniter/Ellislab. Retrieved from <http://ellislab.com/codeigniter>
- Hakim, L. (2014). *Proyek Website Super Wow Dengan PHP & jQuery*. Yogyakarta: Lokomedia.
- Ibnu, D. (2011). *Framework Codeigniter Sebuah Panduan dan Best Practice*. Pekanbaru: e-book.
- Anna. 2010. Penggunaan Microcontroller Sebagai Pendeteksi Posisi dengan Menggunakan Sinyal GSM. *Jurnal Informatika*. Vol. 4 (I). Hlm. 431.
- Arduino. 2018. Arduino Nano. <https://store.arduino.cc/usa/arduino-nano>.
- Kurnia Dewi, Suti. 2018.”Perancangan Prototipe Sistem Kontrol Suhu dan Kelembaban pada Gedung Walet dengan Mikrokontroler Berbasis Mobile”. Skripsi. Pontianak: Fakultas Teknik, Teknik Informatika, Universitas Tanjungpura.
- No.3.2014.Shalahuddin dan A.S., Rosa. 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.

<http://family-cybercode.blogspot.co.id/2016/01/mengenal-arduino-nano.html>
(diakses 07 Mei 2021)

<http://tentangarduino.blogspot.co.id/2014/09/macam-macam-arduino-iniakan-saya.htm> 1 (diakses 07 Mei 2021)

<http://www.micropik.com/PDF/HCSR04.pdf> (diakses 14 Maret 2022)

https://wiki.wemos.cc/products:d1:d1_mini (diakses 07 mei 2021)