

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

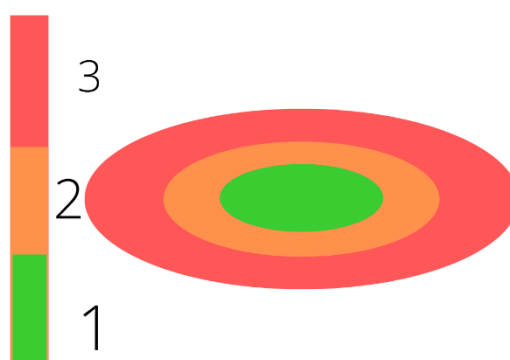
NO	Penulis	Judul	Keterangan
1.	Rizky Aditya Saputra	Visualisasi Banjir Secara Spasial Berdasarkan Data Sensor Ketinggian Berbasis Mikrokontroller Arduino(<i>Prototype</i>)	<ul style="list-style-type: none"> • Membangun sebuah prototipe sistem pemantauan ketinggian air menggunakan mikrokontroler Arduino 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).



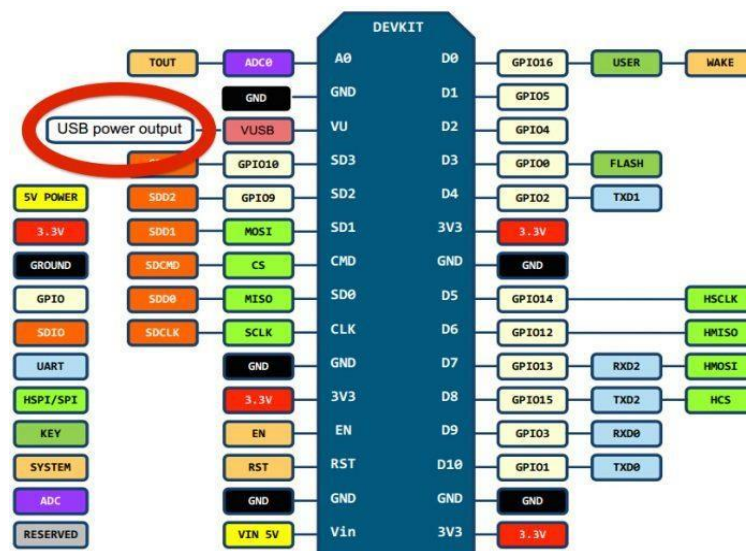
Gambar 2. 2 ESP 8266 NODEMCU V322

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.

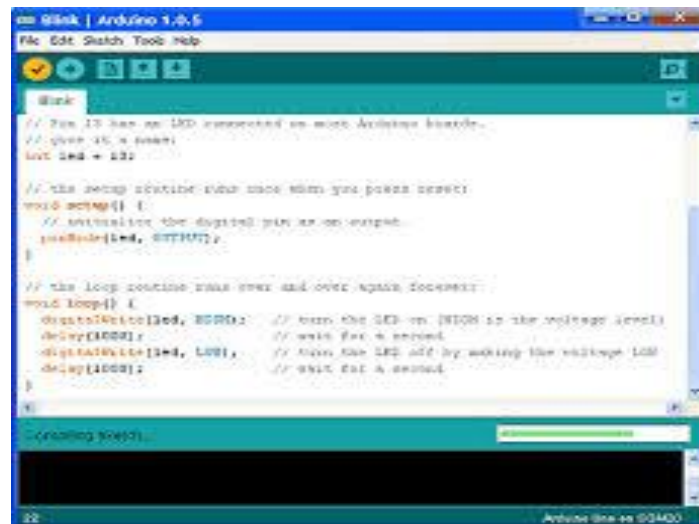
Tampilan awal perangkat lunak pemrograman Arduino dapat sebagaimana dapat dilihat pada Gambar 2.2 berikut.



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.



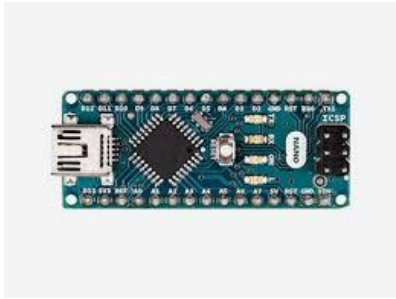
Gambar 2. 5 Tampilan Utama Sketch Arduino

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

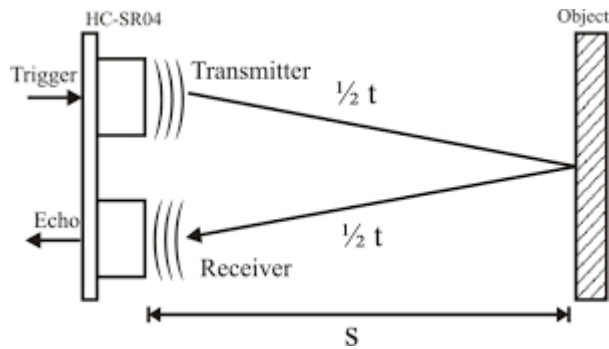
HC-SR04 merupakan sensor ultrasonik yang dapat digunakan untuk mengukur jarak antara penghalang dan sensor. Konfigurasi pin dan tampilan sensor HC-SR04 diperlihatkan pada Gambar 2.5.



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)

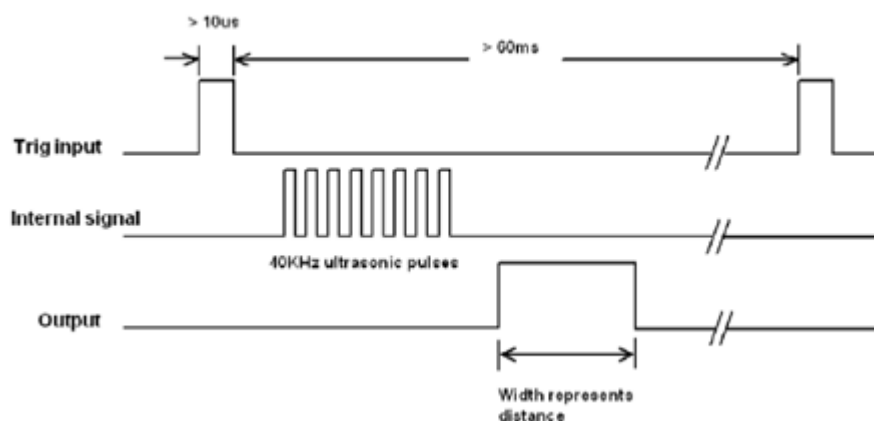
Dimana :

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 }

```

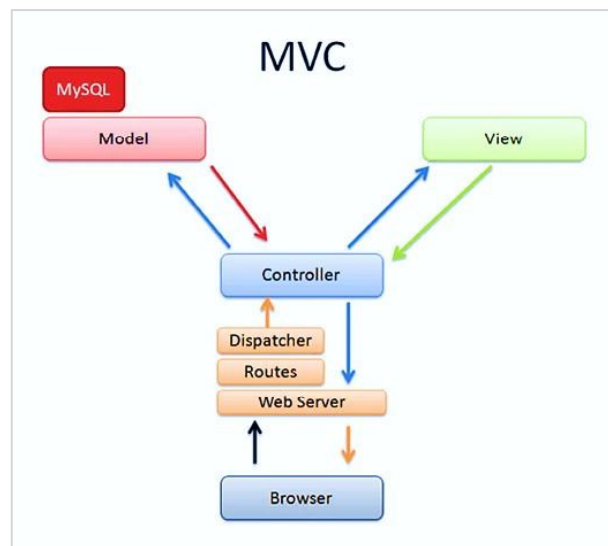
Gambar 2. 11 Tag CSS.

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 getAll()
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, $Pengajuan_seminar, $Pelaksanaan_seminar, $Revisi, $Tanggal)
15    {
16        $data = array (
17            'Pelaksanaan'      => $Pelaksanaan,
18            'Tanggal_mulai'    => $Tanggal_mulai,
19            'Tanggal_selesai'  => $Tanggal_selesai,
20            'Pengajuan_seminar' => $Pengajuan_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:





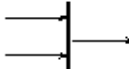

a. *Use Case Diagram*

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 perekrutan 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).