

BAB II

KAJIAN PUSTAKA

Tinjauan Pustaka diperlukan agar penelitian mempunyai dasar yang kokoh. Penelitian atau riset berarti pencarian teori, pengujian teori atau pemecahan masalah. Teori adalah alur logika atau penalaran yang merupakan seperangkat konsep, definisi, dan proposisi yang disusun secara sistematis, sehingga secara umum teori mempunyai tiga fungsi, yaitu: menjelaskan (*explain*), meramalkan (*predict*), dan mengendalikan (*control*) suatu gejala.

Untuk menjelaskan landasan teori ini, penulis menjelaskan beberapa hal yang berkaitan dengan pintu gerbang otomatis tersebut.

2.1 Teori Dasar

Pemahaman dasar mengenai kelistrikan dan elektronik sangat mendukung dalam penyelesaian skripsi ini disamping perlunya pemahaman mengenai penggunaan *software* yang terdapat pada aplikasi *Arduino Uno* guna mengaktifkan *hardware* tersebut. Listrik jika dipandang dari sumber arusnya dapat dikenal ada 2 (dua) yaitu Arus Bolak Balik (*Alternating Current*) dan Arus Searah (*Direct Current*). *Mikrokontroller Arduino Uno* ini menggunakan kedua sumber tersebut. Penggunaan komponen-komponen elektronik yang mendukung diantaranya : Tahanan (*Resistor*), Lampu LED (*Light Emitting Diode*), *switch*, *buzzer*, *timer*.

2.2 Teori Khusus

Fuzzy Logic (= logika fuzzy / logika kabur) merupakan kecerdasan buatan yang pertama kali dipublikasikan oleh Prof. Dr. Lotfi Zadeh yang berasal dari Pakistan untuk membuat keputusan. *Fuzzy Logic* (Logika Fuzzy) merupakan suatu logika yang memiliki nilai kekaburan atau kesamaran (*fuzzyness*) antara benar dan salah. Logika *Fuzzy* memiliki derajat keanggotaan dalam rentang 0 dan 1. Logika *Fuzzy* adalah suatu cara untuk memetakan suatu ruang input ke output, sehingga dengan menerapkan *Fuzzy Logic* pada suatu sistem kendali (*control*) akan dihasilkan proses yang dapat beroperasi secara otomatis. Logika *fuzzy* digunakan untuk menterjemahkan suatu besaran yang diekspresikan menggunakan bahasa (*linguistic*) misalkan besaran kecepatan motor servo yang diekspresikan dengan maju (terbuka) dan mundur (tertutup), sehingga penulis menganggap logika *fuzzy* merupakan cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang output. Sistem yang menggunakan *Fuzzy Logic* membutuhkan spesifikasi antara masukan dan keluaran, yang secara bentuk umum dari aturan dinyatakan dengan:

$$\text{If } (A_1) \text{ THEN } (B_1)$$

$$\text{If } (A_n) \text{ THEN } (B_n)$$

dimana: A_1, \dots, A_n = *Antecedent* yang merupakan masukan (*input*) yang telah *defuzzifikasi*.

B_1, \dots, B_n = *consequent* yang merupakan keluaran (*output*).

Penelitian yang dilakukan penulis untuk memberikan ide dari pengendalian pintu gerbang menggunakan logika *fuzzy*. Untuk melakukan proses ini secara otomatis, penulis menggunakan sensor jarak *Ultrasonic* dan *Bluetooth* untuk mendeteksi parameternya (misalnya : jarak dan kecepatan).

Beberapa alasan digunakannya logika *fuzzy* pada pintu gerbang otomatis ini (Sri Kusumadewi. *Artificial Intelligence Teknik dan Aplikasinya*. 2003 :154) yaitu:

- a. Konsep logika *fuzzy* mudah dimengerti. Disini konsep matematis yang mendasari penalaran fuzzy sangat sederhana dan mudah dimengerti.
- b. Logika *fuzzy* sangat *fleksibel*
- c. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui pelatihan yang lama.
- d. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.

Dalam sistem Kontrol Logika *Fuzzy* (*Fuzzy Logic Control*) terdapat beberapa tahapan operasional yang meliputi:

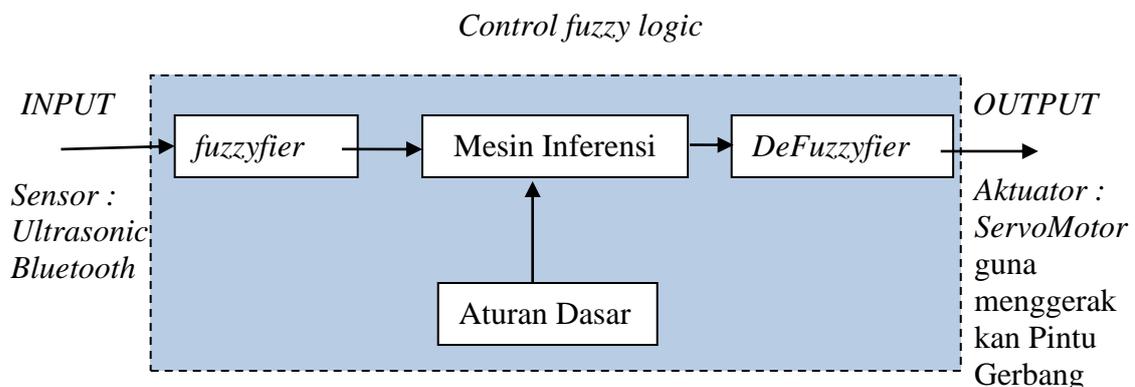
1. *Fuzzyfier*
2. Penalaran (*Inference Machine*)
3. Aturan Dasar (*Rule Based*)
4. *Defuzzyfier*

Fuzzifier : untuk melakukan *fuzzifikasi* yaitu mengubah nilai numerik variabel input (umumnya berupa nilai hasil pembacaan sensor) ke nilai *linguistik* yang berupa derajat keanggotaan pada suatu himpunan *fuzzy variabel input*. 10

Rule Base : berisi aturan IF-THEN yang menggambarkan pemetaan hubungan nilai-nilai *variabel input* dan nilai *variabel output*.

Defuzzifier : untuk melakukan proses yang sebaliknya yaitu mengubah nilai *linguistik* hasil inferensi *fuzzy* ke nilai numerik yang siap dikirimkan ke *output*.

Menurut Dwi Ana Ratna Wati. Sistem Kendali Cerdas. 2011 : 4. Dengan menggunakan tipe Mamdani dapat digambarkan Blok Diagram Kendali Logika Fuzzy (*Fuzzy Logic Control*) sebagai berikut:



Gambar 2.1 Blok Diagram Control Fuzzy Logic

Keputusan yang akan dibuat pengendali *fuzzy* didasarkan pada aturan yang tersedia.

Aturan himpunan yang digunakan disini untuk menghasilkan output adalah:

1. Jika *sensor Ultrasonic* terkena halangan kendaraan pada jarak jangkauan, maka pintu gerbang terbuka.
2. Jika *sensor Ultrasonic* tidak terkena halangan kendaraan pada jarak jangkauan, maka pintu gerbang tertutup.
3. Jika *sensor Bluetooth* terkena pada jarak jangkauan dari pengaturan *Android*, maka pintu gerbang terbuka.
4. Jika *sensor Bluetooth* terkena pada jarak jangkauan dari penaturan *Android*, maka pintu gerbang tertutup.

2.3 Perangkat (*Tools*) / *Software* Pintu Gerbang Otomatis

Ada beberapa *hardware* utama yang perlu dikenal untuk menggerakkan pintu gerbang otomatis ini, yaitu:

- a. Pintu Gerbang Besi : sebagai benda yang akan digerakkan buka dan tutup oleh proses *mikrokontroller* pada *servomotor*.
- b. *Arduino Uno* : sebagai rangkaian *mikrokontroller*.
- c. *Sensor Ultrasonic* : sebagai pemberi input guna proses.
- d. *Sensor Bluetooth* : sebagai pemberi input guna proses dari *Android*.
- e. *Timer Omron* : sebagai pengatur waktu otomatis pengalihan *switch* penggunaan sensor *Ultrasonic* atau *Bluetooth*.
- f. *Servo motor* : sebagai *output* dari proses *input* (*aktuator*).
- g. *Adaptor* : sebagai catu daya Arus Searah (*Direct Current*) untuk peralatan elektronik yang digunakan setelah *Arduino Uno* dimasukkan program terlebih dahulu yang terhubung dengan *port* USB menggunakan bahasa C pada komputer / *laptop*.

2.3.1 Pintu Gerbang Besi

Untuk pintu gerbangnya dibuat dari bahan besi yang telah dilas dengan panjang ± 50 Cm dan tinggi ± 30 Cm yang kerjanya dapat membuka dan menutup secara otomatis dengan cara bergeser horizontal kekiri dan kekanan pada relnya mengikuti pergerakan arah putaran motor servo berdasarkan input dari *sensor ultrasonic* atau *bluetooth*.

Untuk bentuk peralatan tersebut dapat dilihat seperti gambar dibawah ini:



Gambar 2.2 Pintu Gerbang Otomatis

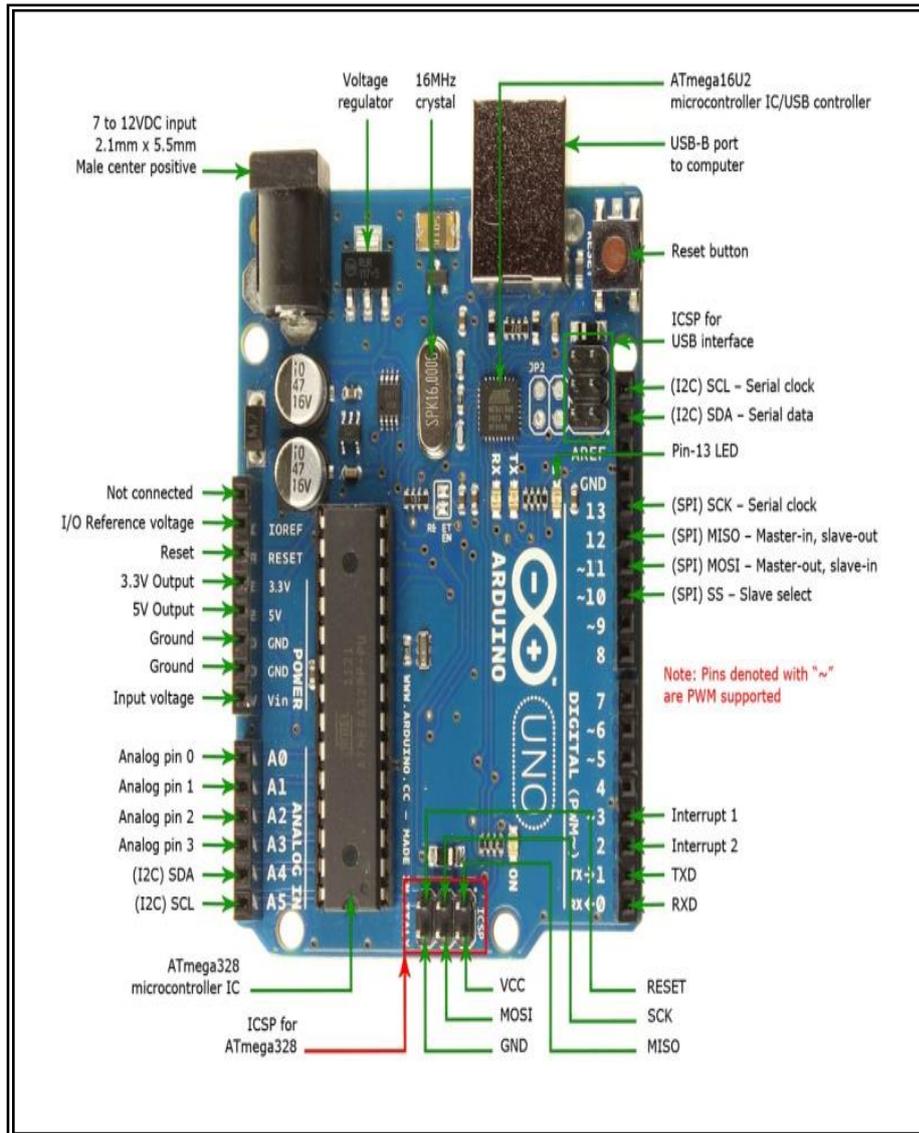
2.3.2 *Arduino Uno*

A. *Board Arduino Uno*

Arduino Uno merupakan salah satu papan (*board*) mikrokontroler *Atmega328* yang bersifat *open source* penemuan Massimo Banzi dan David Cuartielles pada tahun 2005 di Ivre, Italia dengan tujuan pengembangan awal yaitu untuk membantu para siswa dapat membuat perangkat desain dan interaksi dengan harga yang murah / terjangkau dibandingkan dengan perangkat lain yang tersedia. Secara umum, pada *Arduino Uno* dapat kita bagi atas dua bagian, yaitu:

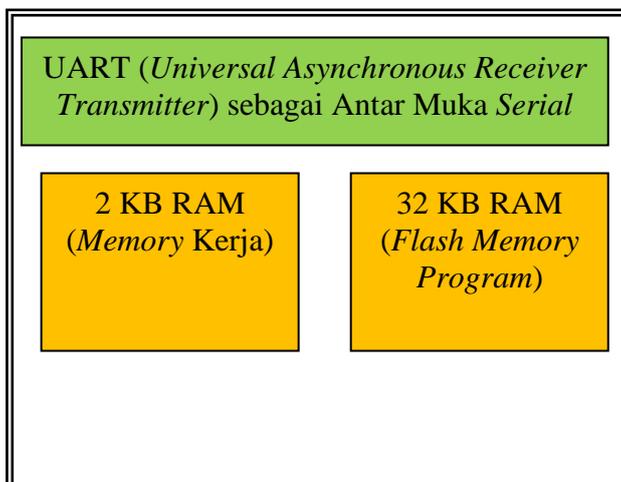
13

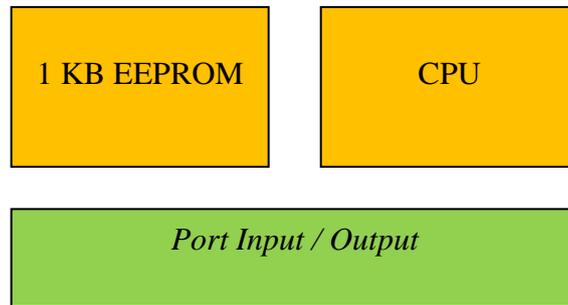
1. *Hardware* merupakan bentuk fisik papan (*board*) *input/output (I/O)*.
2. *Software* merupakan non fisik untuk membuat, mengkompilasi dan meng-*upload* program yaitu *Arduino IDE* yang bisa diprogram menggunakan bahasa C.



Gambar 2.3 Nama-nama bagian pada board *Arduino Uno*.

Didalam sebuah IC (*Integrated Circuit*) mikrokontroller biasanya ada terdapat CPU (*Central Processing Unit*), memori, timer, saluran komunikasi serial dan port input / output, ADC (*Analog Digital Converter*).





Gambar 2.4 Diagram blok sederhana dari *Arduino Uno* ATmega328

Struktur *Memory* pada *Arduino Uno* ATmega328 mempunyai 32 KB (dengan 0,5 KB digunakan untuk *bootloader*). ATmega328 juga mempunyai 2 KB SRAM dan 1 KB EEPROM (yang dapat di *Read* dan *Write* dengan *EEPROM Library*).

B. Bahasa Pemrograman *Arduino Uno*

Menurut Andrianto dan Aan Darmawan. *Arduino Belajar Cepat dan Pemrograman* 2016:45. Setiap program *Arduino* yang biasa disebut dengan sebutan *sketch* mempunyai dua buah fungsi yang harus ada. Fungsi yang harus ada tersebut yaitu:

15

1. `void setup() { }`

Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program *Arduino* dijalankan untuk pertama kalinya. Jadi disini `setup()` adalah persiapan sebelum *eksekusi* program.

2. `void loop() { }`

Fungsi ini akan dijalankan setelah setup (fungsi *void setup*) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (*power*) dilepaskan. Jadi disini *loop()* adalah tempat menulis program utama yang akan dieksekusi.

Ada beberapa fungsi-fungsi dasar pada bahasa pemrograman *Arduino* diantaranya yaitu:

a. *Syntax*

Berikut ini adalah elemen bahasa C yang dibutuhkan untuk format penulisan.

Tabel 2.1 Penamaan fungsi *Syntax* pada bahasa C

Tanda	Keterangan
//	(komentar satu baris) Digunakan untuk memberi catatan pada diri sendiri apa arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang kita ketikkan dibelakangnya akan diabaikan oleh program.
/* */	(komentar banyak baris) 16 Jika kita punya banyak catatan, maka hal itu dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.
{ }	(kurung kurawal)

	Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).
;	(titik koma) Setiap baris kode harus diakhiri dengan tanda titik koma (jika ada titik koma yang hilang maka program tidak akan bisa dijalankan).

b. *Variabel*

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang tepat. Jadi disini *Variabel* adalah ekspresi yang digunakan untuk mewakili suatu nilai yang digunakan dalam program.

Tabel 2.2 Penamaan fungsi *Variabel* pada bahasa C

Nama	Keterangan
int	(<i>integer</i>) Digunakan untuk menyimpan angka dalam 2 byte (16 bit). Tidak mempunyai angka ecimal dan menyimpan nilai dari -32,768 dan 32,767.
Long	(<i>long</i>) Digunakan ketika <i>integer</i> tidak mencukupi lagi. Memakai 4 <i>byte</i> (32 bit) dari <i>memori</i> (RAM) dan mempunyai rentang dari -2,147,483,648 dan 2,147,483,647.
<i>decimal</i>	(<i>boolean</i>)

	Variabel sederhana yang digunakan untuk menyimpan nilai <i>TRUE</i> (benar) atau <i>FALSE</i> (salah). Sangat berguna karena hanya menggunakan 1 <i>bit</i> dari RAM. ¹⁷
<i>Float</i>	<i>(float)</i> Digunakan untuk angka desimal (<i>floating point</i>). Memakai 4 <i>byte</i> (32 bit) dari RAM dan mempunyai rentang dari -3.4028235E+38 dan 3.4028235E+38.
Char	<i>(character)</i> Menyimpan 1 karakter menggunakan kode ASCII (misalnya 'A' = 65). Hanya memakai 1 <i>byte</i> (8 bit) dari RAM.
<i>Byte</i>	Digunakan menyimpan data <i>numerik</i> bernilai 8 <i>bit</i> , tidak memiliki nilai desimal, data bertipe <i>byte</i> nilainya berkisar 0 – 255.

c. Operator Matematika

Operator

Operator yang digunakan untuk memanipulasi angka (bekerja seperti matematika yang sederhana).

Tabel 2.3 Penamaan fungsi Operator pada bahasa C

Tanda	Keterangan
=	Membuat sesuatu menjadi sama dengan nilai yang lain (misalnya: $x = 10 * 2$, x sekarang sama dengan 20).
%	Menghasilkan sisa dari hasil pembagian suatu angka dengan angka yang lain (misalnya: $12 \% 10$, ini akan menghasilkan angka 2).
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian

Operator Pembandingan

Digunakan untuk membandingkan nilai logika.

Tabel 2.4 Penamaan fungsi Operator Pembandingan pada bahasa C

Tanda	Keterangan
==	Sama dengan (misalnya: $12 == 10$ adalah <i>FALSE</i> (salah) atau $12 == 12$ adalah <i>TRUE</i> (benar).
!=	Tidak sama dengan (misalnya: $12 != 10$ adalah <i>TRUE</i> (benar) atau $12 != 12$ adalah <i>FALSE</i> (salah).
<	Lebih kecil dari (misalnya: $12 < 10$ adalah <i>FALSE</i> (salah) atau $12 < 12$ adalah <i>FALSE</i> (salah) atau

	12 < 14 adalah <i>TRUE</i> (benar).
>	Lebih besar dari (misalnya: 12 > 10 adalah <i>TRUE</i> (benar) atau 12 > 12 adalah <i>FALSE</i> (salah) atau 12 > 14 adalah <i>FALSE</i> (salah).

d. Struktur Pengaturan

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan.

Tabel 2.5 Penamaan fungsi Struktur Pengaturan pada bahasa C

Instruksi	Keterangan
<p>if...else, dengan format seperti berikut ini:</p> <pre>if (inputPin == HIGH) { digitalWrite (0,HIGH); } else { digitalWrite (0,LOW);}</pre>	<p>Memungkinkan untuk mengeksekusi instruksi yang lain jika suatu kondisi tidak terpenuhi.</p> <p style="text-align: right;">20</p>
<p>For dengan format seperti berikut ini:</p> <pre>for (i = 0; i <10; i++) { digitalWrite (0,HIGH); }</pre>	<p>Digunakan bila ingin melakukan pengulangan kode di dalam kurung kurawal beberapa kali, ganti pengulangan dengan jumlah pengulangan yang diinginkan. Melakukan penghitungan ke atas dengan <i>i++</i> atau ke bawah dengan <i>i--</i>.</p>

e. *Digital*

Tabel 2.6 Penamaan fungsi *Digital* pada bahasa C

Instruksi	Keterangan
<p>pinMode(pinMode)</p> <p>contoh:</p> <pre>pinMode(3,OUTPUT); //pindigital3</pre> <p>diinisiasi sebagai <i>output</i></p>	<p>Instruksi yang digunakan pada fungsi <i>void setup()</i> untuk menginisialisasi suatu <i>pin</i> sebagai <i>input</i> atau <i>output</i>.</p>
<p>digitalWrite(pin,value)</p> <p>contoh:</p> <pre>digitalWrite(pin, HIGH);</pre>	<p>Instruksi yang digunakan untuk memberi nilai output <i>HIGH</i> (1) atau <i>LOW</i> (0) pada <i>pin digital</i>. <i>Pin digital</i> dapat diartikan sebagai suatu <i>variabel</i> atau konstanta 0 – 13 yang mewakili <i>input</i> dan <i>output</i> dari <i>board arduino</i>.</p>
<p>digitalRead(pin)</p> <p>contoh:</p> <pre>value = digitalRead(pin)</pre>	<p>Instruksi yang digunakan untuk membaca <i>input</i> dari suatu <i>pin</i> yang hasilnya berupa logika <i>HIGH</i> atau <i>LOW</i>. <i>Pin</i> dapat diartikan sebagai suatu <i>variabel</i> atau konstanta 0 – 13 yang mewakili <i>input</i> dan <i>output</i> dari <i>board arduino</i>.</p>

f. *Analog*

Arduino adalah mesin digital tetapi mempunyai kemampuan untuk beroperasi di dalam *analog*.

Tabel 2.7 Penamaan fungsi *Analog* pada bahasa C

Instruksi	Keterangan
<i>analogWrite(pin,value)</i> contoh: <code>analogWrite(pin, HIGH);</code>	Instruksi yang berfungsi untuk memberi nilai PWM (<i>pulse width modulation</i>) pada <i>output</i> .
<i>analogRead(pin)</i> contoh: <code>value = analogRead(pin)</code>	Instruksi untuk membaca nilai <i>input analog</i> dengan resolusi 10 <i>bit</i> . Instruksi ini hanya berlaku untuk pin A0-A5 yang mampu membaca nilai <i>analog</i> . Karena beresolusi 10 <i>bit</i> maka hasil pembacaan nilai <i>digital</i> antara 0 sampai 1023. 0 (untuk 0 <i>volts</i>) dan 1024 (untuk 5 <i>volts</i>).

2.3.3 *Sensor Jarak Ultrasonic*

Sensor adalah suatu komponen yang digunakan untuk memberi masukan data / *value* ke *arduino* untuk kemudian diproses. Ada banyak sekali macam *sensor*, seperti sensor gerak PIR (*Passive Infra Red*), *sensor* gas, *sensor* cahaya, *sensor* suara, *sensor* suhu, *sensor* kelembaban, *sensor* sentuh. Pada pintu gerbang otomatis ini, penulis menggunakan *sensor* jarak *Ultrasonic* HC-SR04 sebagai *inputnya*.

Sensor Ultrasonic bekerja dengan cara memancarkan gelombang suara dengan besar frekuensi 20 KHz – 20 MHz. *Sensor Ultrasonic* terdiri dari *transmitter* dan *receiver*. *Sinyal ultrasonik* yang dibangkitkan akan dipancarkan dari *transmitter ultrasonik*. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh *receiver ultrasonik*. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian *mikrokontroler Arduino Uno* untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul).



Gambar 2.5 Sensor Ultrasonic HC-SR04

23

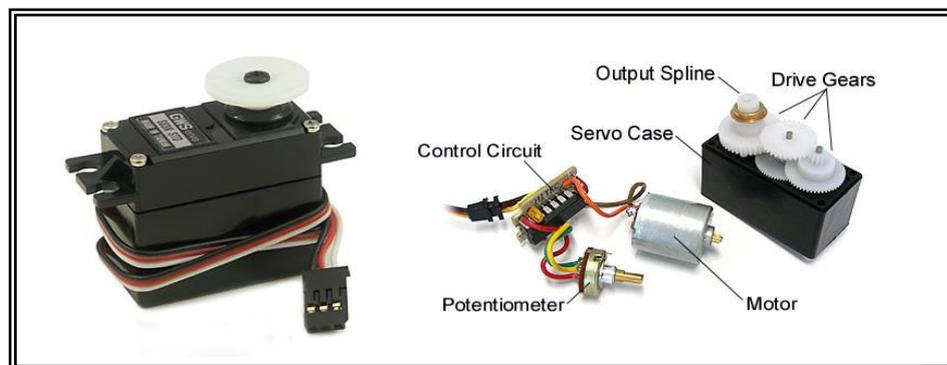
2.3.4 Motor Servo sebagai salah satu Aktuator

Motor Servo adalah aktuator berputar (*rotary*) yang memungkinkan untuk dikontrol secara presisi dari posisi sudut, kecepatan dan percepatannya. *Motor servo* merupakan salah satu dari *aktuator* (komponen *output* dari *Arduino Uno*).

Motor servo sama dengan motor listrik, namun didalamnya terdapat *positionable shaft* (poros) yg dilengkapi roda gigi (*gear*). *Motor servo* dikendalikan dengan mengirimkan pulsa melalui kabel *control* dengan *variable* lebar pulsa terkirim (*Pulse Width Modulation / PWM*).

Berdasarkan arus/tegangan inputnya ada 2 tipe motor servo:

- a. *Motor Servo AC (Alternating Current)*: motor servo ini menggunakan sumber listrik arus bolak-balik (*Alternating Current*) yang biasanya dirancang untuk lonjakan arus yang cukup tinggi, sehingga banyak digunakan pada industri-industri.
- b. *Motor Servo DC (Direct Current)*: motor servo ini menggunakan sumber listrik arus searah (*Direct Current*) yang biasanya dirancang untuk lonjakan arus yang cukup rendah, sehingga biasanya cocok untuk aplikasi lebih sederhana dan murah dibandingkan *motor servo AC*.



Gambar 2.6 *Motor Servo* dan bagian-bagian didalamnya

24

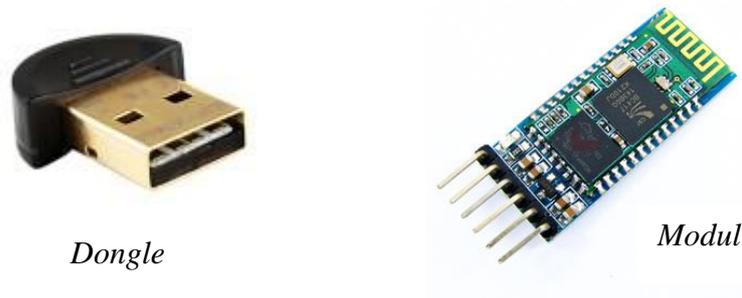
2.3.5 *Bluetooth* sebagai salah satu dari *Modul*

Komunikasi *nirkabel* dapat dilakukan dengan menggunakan *Bluetooth*, sehingga dua peralatan atau lebih dapat saling berkomunikasi tanpa kabel.

Secara umum ada dikenal *Bluetooth Dongle* yang dipasangkan pada *port* USB komputer dan *Modul Bluetooth* yang dipasang pada sisi *Arduino*. *Bluetooth* adalah sebuah teknologi komunikasi *nirkabel* yang beroperasi pada pita frekuensi 2,4 GHz. *Bluetooth* dapat dipakai untuk melakukan komunikasi data diantara peralatan dengan



jarak jangkauan yang cukup jauh. Beberapa jenis *modul bluetooth* diantaranya tipe HC-04, HC-05, HC-06.



Dongle

Modul

Gambar 2.7 *Bluetooth dongle dan Modul Bluetooth.*

2.3.6 Alat Ukur *Multitester*

Multitester merupakan salah satu alat ukur listrik / elektronik yang dapat mengukur besaran tegangan AC/DC, arus AC/DC dan tahanan. Dalam pembuatan / perakitan komponen pintu gerbang, *multitester* sangat diperlukan untuk mengetahui apakah komponen yang digunakan dalam keadaan bagus atau tidak, karena 25 an menggunakan *multitester* kita dapat mengukur besaran-besaan listrik tersebut. Besaran listrik tersebut tidak bisa dilihat dengan pandangan mata saja namun kita bisa mengetahuinya dengan mengukur besarnya menggunakan alat ukur tersebut.



Gambar 2.8 Alat Ukur *Multitester*

2.3.7 Pengenalan *Software*

Untuk dapat menjalankan *hardware board Arduino Uno* ada beberapa hal yang perlu dilakukan seperti dibawah ini.

a. Instalasi *Arduino*

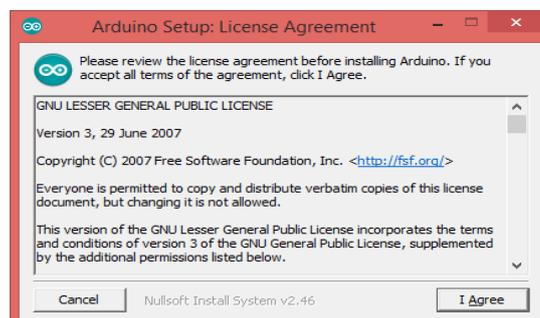
Software IDE (Integrated Development Environment) Arduino merupakan *software* yang digunakan untuk menulis program (pada *Arduino* disebut *Sketch*), kemudian mengkompilasinya menjadi kode bahasa mesin untuk dieksekusi, dan melakukan *debug* jika ada kesalahan serta meng-*Upload sketch* tersebut. Tampilan *software* ini sangat *user friendly*. *Software IDE Arduino* mutlak harus : 26 at melakukan *programming*.

Menginstall *software IDE Arduino* begitu sangat mudah pada *operating system windows*, kita hanya perlu menekan tombol *next-next* pada perintahnya tersebut.

Langkah-langkahnya adalah sebagai berikut:

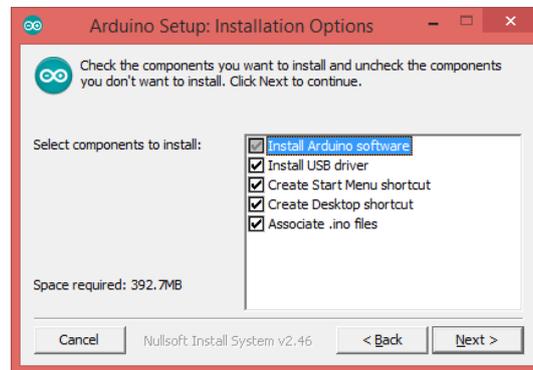
1. Jalankan *Software install*,

Pada Pilihan pertama untuk pilihan license, pilih “*I Agree*”



Gambar 2.9 Konfirmasi persetujuan memasang program *Arduino*

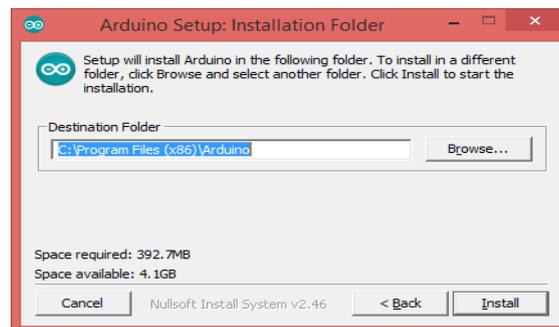
2. Tentukan *installation options*, pilih default kemudian *next*



Gambar 2.10 Pilihan untuk instalasi

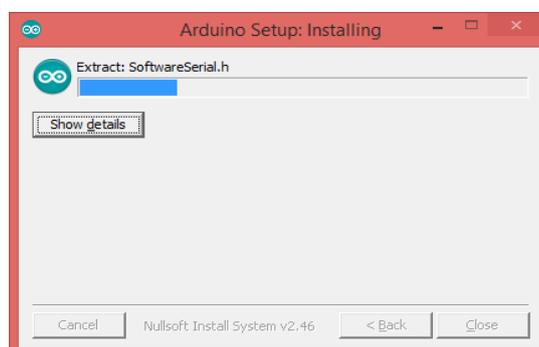
3. Tentukan *destinasi folder* tempat *Arduino* akan diinstall, pilih *default* kemudian *install*

27



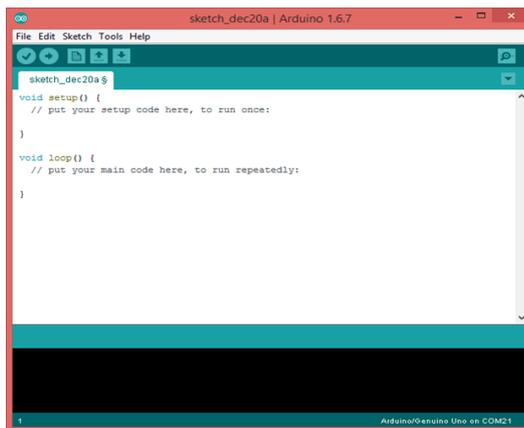
Gambar 2.11 Pilihan *folder*

4. Tunggu hingga *instalasi* selesai dan pilih *close*



Gambar 2.12 Proses instalasi *driver Arduino Uno*

5. Pilih *Arduino* , biasa *default* akan ada *shortcut* di *desktop*, jika berhasil akan tampil seperti berikut



28

Gambar 2.13 Proses instalasi yang telah berhasil

Keterangan:

Pada *tab* / baris pertama ada (*File* , *Edit* , *Sketch*, *Tools*, *Help*), dengan penjelasan sebagai berikut:

- a. *File* berfungsi untuk membuat *sketch* / *coding* baru , simpan , *print*.
- b. *Edit* berfungsi untuk merubah *sketch* / *coding* , seperti *copy*, *paste*, *find*, *undo*.
- c. *Sketch* adalah isian *coding* yang sebelum di upload kedalam *Arduino* kita harus melakukan verifikasi agar tidak ada kesalahan alur / pemrograman . untuk menambah *library* baru yang tidak terinstall *default* di dalam *Arduino* IDE, fiturnya ada didalam *sketch* ini, di *Sketch* > *Include Library* > *Add .Zip Library*.

- d. *Tools* berfungsi untuk menentukan pilihan *board Arduino* yang dipakai, dan *port* yang digunakan, harap memastikan pilihan *board Arduino* yang dipakai dan *port* yang digunakan saat ingin mengupload / menggunakan Arduino, karena jika tidak proses *upload* akan *error* karena tidak sesuai dengan *Arduino / port* yang digunakan.
- e. *Help* adalah informasi bantuan yang dibutuhkan ketika kesulitan dalam menggunakan *Arduino IDE*.

Pada *tab / baris* kedua, ada beberapa *icon* unik yang memiliki fungsi fungsi tersendiri yaitu :



: tanda lambang untuk *verifikasi coding* agar tidak ada kesalahan.

29



: tanda lambang untuk mengupload *coding* kedalam *board Arduino*.



: tanda lambang untuk membuat *sketch* baru / *coding* baru.



: tanda lambang untuk membuka *sketch / coding* yang sudah dibuat sebelumnya dan disimpan didalam lokasi tertentu.



: tanda lambang untuk menyimpan *sketch / coding* yang sudah dibuat.

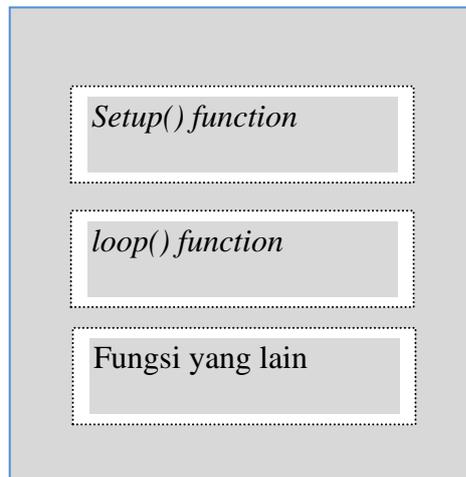


: tanda lambang untuk *serial monitoring*, biasanya digunakan untuk memeriksa alur program dan hasil program dalam bentuk *text* , jadi kita bisa tahu program berhasil dijalankan.

Dan yang *tab / baris* ketiga , yang ditandai dengan *window* berwarna putih , itu adalah tempat *code* dibuat dan dimanipulasi agar *board Arduino* dapat berfungsi sesuai yang dibutuhkan.

2.3.8 Bahasa pemrograman *Arduino* berbasis C

Struktur Program C *Arduino* minimal terdiri dari dua fungsi yaitu *setup()* dan *loop()*.



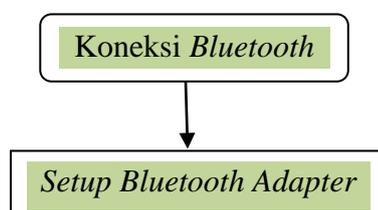
30

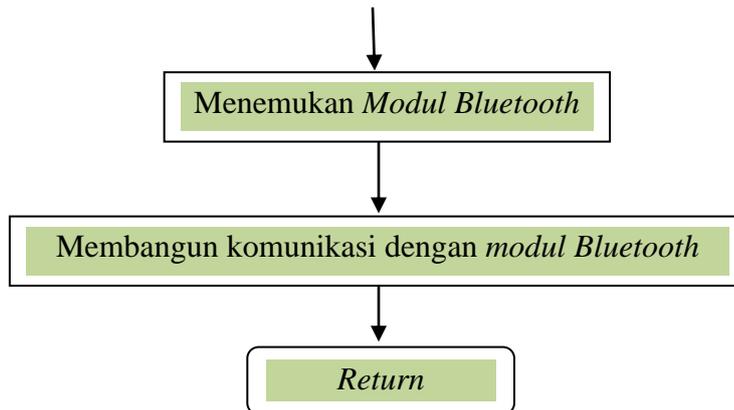
Gambar 2.14 Struktur Program C pada *Arduino Uno*

Fungsi *setup()* dijalankan pertama kali setiap *board arduino* dihidupkan sedangkan fungsi *loop()* dijalankan terus menerus selama *board arduino* hidup.

2.3.9 Aplikasi *Android* untuk koneksi *control jarak jauh*

Android merupakan sistem operasi open source pada perangkat *mobile*. Aplikasinya bisa kita *download* melalui *Play Store* secara gratis. *Android* adalah suatu *software* (perangkat lunak) yang berbasis *Linux* untuk telepon seluler dan komputer *tablet*. Pada umumnya *mobilephone android* sudah tersedia saluran komunikasi data secara nirkabel, salah satunya *Bluetooth*. Saluran komunikasi data tersebut dapat digunakan untuk mengirim / menerima data dari / ke perangkat lain.





Gambar 2.15 Diagram alir prose koneksi *Bluetooth*

Sumber: Andrianto dan Darmawan. (2015:143).

2.4 Penelitian Terdahulu

Guna mendukung dalam penyelesaian penelitian ini, penulis mengambil beberapa bahan pertimbangan dari beberapa jurnal dalam negeri yang terkait.

1. Judul Jurnal : Perancangan dan Implementasi Pengendali Pintu Pagar Otomatis Berbasis *Arduino*

Penulis Jurnal : Wilfrid Sahputra Girsang, Fakhruddin Rizal Batubara, ST. MTI.

ISSN/Vol/Tahun: Vol.7 No.2/Mei 2014

Pembahasan :

Wilfrid Sahputra Girsang, Fakhruddin Rizal Batubara, ST.MTI. 2014. Perancangan dan Implementasi Pengendali Pintu Pagar Otomatis Berbasis *Arduino*. Dari hasil jurnal tersebut disimpulkan bahwasanya Pintu pagar Otomatis yang dirancang bangun tersebut digerakkan oleh suatu rangkaian *mikrokontroller Arduino UNO* dan *Arduino Ethernet Shield*. Pada saat berkomunikasi antara pagar

dan *android* dalam saat akses membuka dan menutup dipengaruhi oleh jenis antenna dan panjang antenna serta jaringan saat mengakses IP.

2. Judul Jurnal : Sistem Keamanan Pintu Pagar Otomatis Menggunakan *Voice Recognition*

Penulis Jurnal : Ashar Seppiawan N, Nurussa'adah, Ir., MT., Ponco Siwindarto, M.Eng.Sc.

ISSN/Vol/Tahun: -

32

Pembahasan :

Ashar Seppiawan N, Nurussa'adah, Ir. MT. dan Ir. Ponco Siwindarto, M.Eng.Sc. Sistem Keamanan Pint Pagar Otomatis Menggunakan *Voice Recognition*. Dari hasil jurnal tersebut disimpulkan bahwasanya Pintu Pagar Otomatis yang dirancang bangun tersebut digerakkan oleh suatu rangkaian *mikrokontroller Arduino Uno* dan *Modul Easy Voice Recognition (Easy VR)*. *Modul Easy VR* ini dapat digunakan sebagai sistem keamanan pintu otomatis berpassword suara yang menggantikan peran kunci *konvensional*.

3. Nama Jurnal : Jurnal Teknik Elektro Institut Teknologi Padang

Judul Jurnal : Pengendalian Pintu Pagar Geser Menggunakan Aplikasi *Smartphone Android* dan *Mikrokontroler Arduino* melalui *Bluetooth*

Penulis Jurnal : Andi Syofian

ISSN/Vol/Tahun: No.2252-3472/Voluem 5, No.1/Januari 2016

Pembahasan :

Andi Syofian. 2016. Pengendalian Pintu Pagar Geser menggunakan Aplikasi *Smartphone Android* dan *Mikrokontroller Arduino* melalui *Bluetooth*. Dari hasil

jurnal tersebut disimpulkan bahwasanya Pintu Pagar Otomatis yang dirancang bangun tersebut digerakkan oleh suatu rangkaian *mikrokontroller Arduino Uno* dan *Bluetooth* yang dikendalikan *App Inventor* pada *smarphone android*.

4. Nama Jurnal : Jurnal LINK

Judul Jurnal : Pintu Pagar Otomatis dengan Kontrol Suara E³³ is
Smartphone Android

Penulis Jurnal : Mohammad Rizal Saifuddin, Slamet Winardi

ISSN/Vol./Tahun : 1858-4667/Vol.22 No.1/Februari 2015

Pembahasan :

Mohammad Rizal Saifuddin. Slamet Winardi. 2015. Pintu Pagar Otomatis dengan Kontrol Suara Berbasis Smartphone Android. Dari hasil jurnal tersebut disimpulkan bahwasanya Pintu Pagar Otomatis yang dirancang bangun tersebut digerakkan oleh suatu rangkaian *mikrokontroller Arduino Uno* dan *Bluetooth* yang dikendalikan *Aplikasi Speech Recognition* (perintah suara) dari *smartphone android* dalam melaksanakan perintah proses tutup agak ada kesulitan karena perintah tutup hampir sama dengan ucapan *youtube*.

5. Nama Jurnal : Jurnal Sistem Komputer

Judul Jurnal : Implementasi *Prototype* Sistem Kendali Kunci Pintu dengan
Smartphone Android Berbasis *Microcontroller AVR*
ATMEGA 328 dan *Fuzy Logic*

Penulis Jurnal : Desmira, M.Natsir, Maya Selvia L, Hamdi Rahman

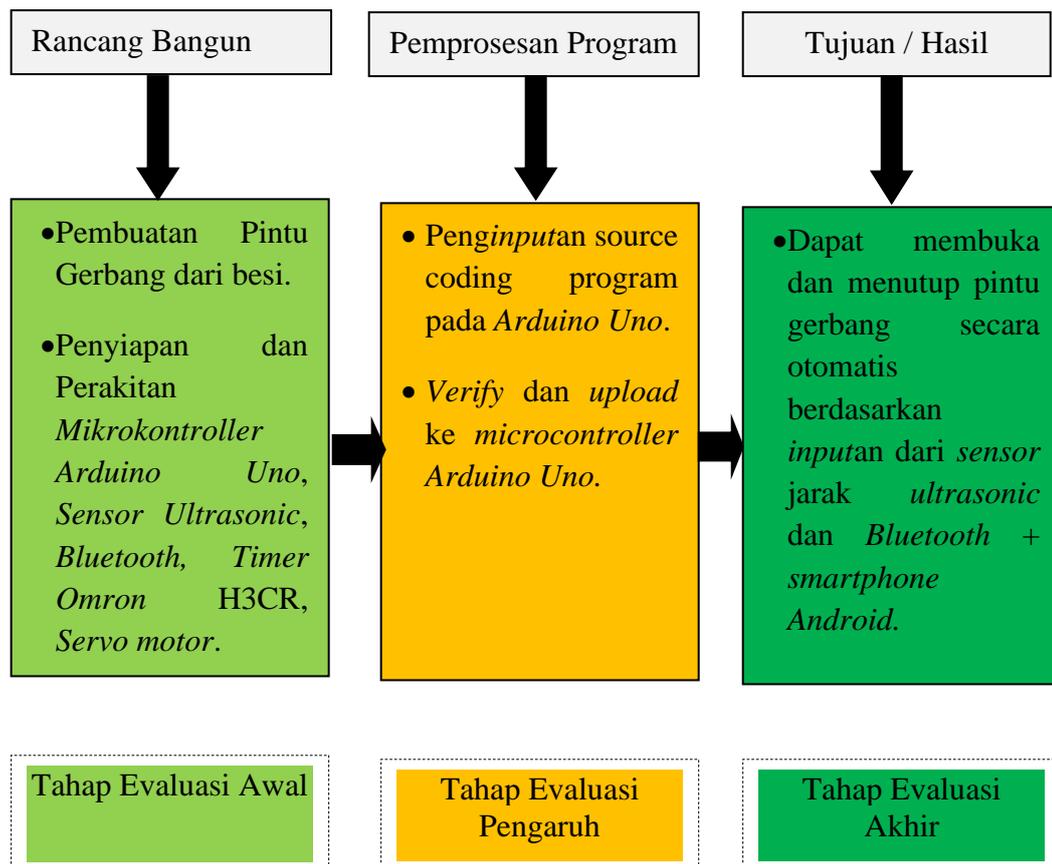
ISSN/Vol./Tahun : 2406-7733/Vol.2 No.1/2015

Pembahasan :

Desmira. M.Natsir. Maya Selvia L. Hamdi Rahman. 2015. Implementasi *Prototype* Sistem Kendali Kunci Pintu dengan *Smartphone Android* Berbasis *Microcontroller AVR ATMEGA 328* dan *Fuzzy Logic*. Dari hasil jurnal tersebut disimpulkan bahwasanya Pintu Pagar Otomatis yang dirancang bangun terdigerakkan oleh suatu rangkaian *mikrokontroller Arduino Uno* dan *Bluetooth* yang dikendalikan Aplikasi *DoorLock* dengan cara menekan tombol *On - Off* dari *smartphone android*.

2.5 Kerangka Pemikiran

Dalam pelaksanaan penelitian ini dapat dibuat kerangka pemikiran sebagai berikut:



Gambar 2.15 Kerangka Pemikiran Pintu Gerbang Otomatis