

BAB II

TINJAUAN PUSTAKA

2.1. Modul ESP8266

Menurut Heri dan Aan (2015:169), WiFi adalah sebuah teknologi yang digunakan pada peralatan elektronik untuk bertukar data secara *nircable* (menggunakan gelombang radio) melalui sebuah jaringan komputer. Peralatan dapat terhubung ke jaringan komputer melalui sebuah *access point*. Sebuah *access point* memiliki jangkauan kurang lebih 20 meter di dalam ruangan, dan lebih luas lagi jika di luar ruangan.

Modul ESP8266 merupakan SoC (*System on Chip*) dengan *stack* protokol TCP/IP yang telah terintegrasi, sehingga dan mudah diakses menggunakan mikrokontroler melalui komunikasi *serial* 802.11 WiFi *Direct* (P2P). Modul WiFi Esp8266 dapat berfungsi sebagai *host* maupun sebagai modul *transfer* data dalam jaringan WiFi. Modul ini memiliki kemampuan pengolahan dan penyimpanan data yang baik sehingga memungkinkan untuk diintegrasikan dengan sensor dan perangkat khusus lainnya melalui GPIO.



Gambar 2.1. Modul ESP8266

2.2. Motor DC

Motor DC adalah motor yang bergerak berputar 360 derajat, biasanya disebut *dynamo* dan biasanya digunakan sebagai penggerak roda. Apabila kutub *positive* dan *negative* sumber yang dipasang ditukar maka motor DC akan berputar berlawanan arah dari arah putar sebelumnya (Heri dan Aan, 2015:131).



Gambar 2.2. Motor DC

2.3. Motor Servo

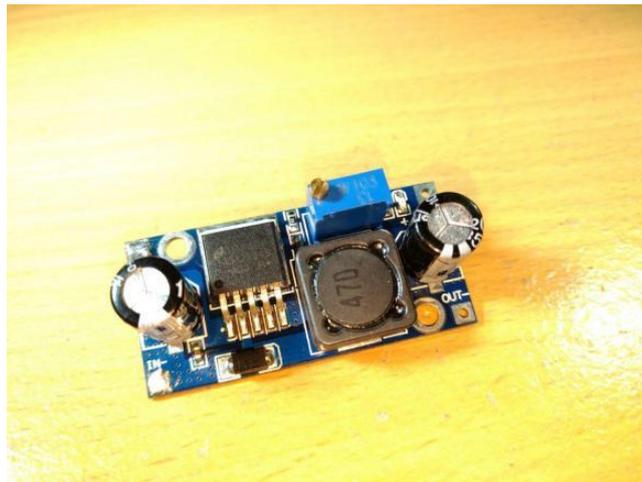
Motor servo adalah sebuah motor dengan sistem umpan balik tertutup, posisi dari motor akan diinformasikan kembali kerangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian roda gigi (*gear*), potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan putaran sudut dari sumbu motor servo diatur (dengan sinyal PWM) berdasarkan lebar pulsa (berkisar antara 0.5 ms sampai dengan 2 ms) yang dikirimkan melalui kaki sinyal dari motor servo. Secara umum terdapat 2 jenis motor servo, yaitu motor servo *standard* (dapat berputar sebesar 180 derajat) dan motor servo *continous* (dapat berputar sebesar 360 derajat) (Heri dan Aan, 2015:127).



Gambar 2.3. Motor Servo

2.4. Module LM2596

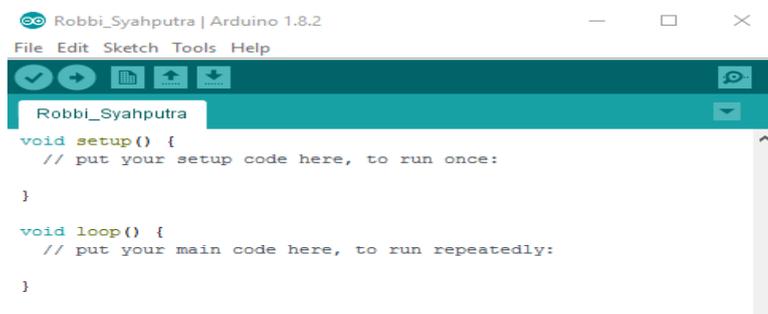
LM2596 merupakan sebuah modul pengkonversi tegangan DC ke DC yang dilengkapi dengan IC penurun dan penaik tegangan. Pada LM2596 tegangan di konversi dan bisa disesuaikan dengan cara memutar bagian pengaturan untuk menurunkan atau menaikkan tegangan.



Gambar 2.4. Module LM2596

2.5. Software IDE Arduino

Menurut Heri dan Aan (2015:31-38), *software* IDE Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *platform Wiring*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang, *hardware* menggunakan *processor* Atmel AVR dan *software* memiliki bahasa pemrograman C++ yang sederhana dan fungsi-fungsinya yang lengkap.



```
Robbi_Syahputra | Arduino 1.8.2
File Edit Sketch Tools Help
Robbi_Syahputra
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Gambar 2.5. IDE Arduino

Berikut komponen-komponen dari *Software IDE Arduino*:

1) **File**

Menu *file* adalah menu bar yang digunakan untuk berinteraksi dengan *file*.

Tabel 2.1. Pilihan pada Menu *File*

<i>Sub Menu</i>	Keterangan
<i>New</i>	Membuat <i>sketch</i> * baru
<i>Open</i>	Membuka <i>file sketch</i> yang sudah disimpan
<i>Sketchbook</i>	Membuka <i>file sketch</i> yang pernah dibuat
<i>Example</i>	Membuka contoh-contoh <i>file sketch</i> yang berisi berbagai macam aplikasi yang disediakan oleh arduino
<i>Close</i>	Menutup <i>sketch</i>
<i>Save</i>	Menyimpan <i>sketch</i>
<i>Save as</i>	Menyimpan <i>sketch</i> dengan nama lain
<i>Upload to I/O Board</i>	Mengunggah program ke <i>board</i>
<i>Page Setup</i>	Mengatur ukuran halaman pada pencetak
<i>Print</i>	Mencetak <i>sketch</i>
<i>Preference</i>	Mengatur <i>setting</i> IDE Arduino
<i>Quit</i>	Keluar dari IDE Arduino

2) **Menu *Edit***

Menu *edit* digunakan untuk mengedit program yang ditulis di *sketch editor*. Menu *edit* memiliki beberapa *sub menu* yaitu *Cut*, *Copy*, *Copy for Forum*, *Copy as HTML*, *Paste*, *Select All*, *Comment*, *Increase Indent*, *Decrease Indent*, *Find*, *Find Next*.

3) **Menu *Sketch***

Menu *sketch* digunakan untuk memeriksa, dan mengunggah *sketch*.

Tabel 2.2. Pilihan pada Menu *Sketch*

<i>Sub Menu</i>	Keterangan
<i>Stop</i>	Menghentikan kompilasi (apabila ' <i>Hang</i> ')
<i>Show sketch folder</i>	Menampilkan <i>folder</i> dari <i>sketch</i> yang sedang dibuka
<i>Import Library</i>	Mengambil <i>hader library</i> dari fungsi-fungsi tambahan
<i>Add File</i>	Menambah buka <i>file sketch</i> pada jendela yang sama

4) Menu Tools

Menu *tools* digunakan untuk memanggil *tools* pendukung *software* arduino.

Tabel 2.3. Pilihan pada Menu Tools

<i>Sub Menu</i>	Keterangan
<i>Auto Format</i>	Mengatur <i>format sketch</i> secara otomatis
<i>Archive Sketch</i>	Menyimpan <i>sketch</i> dalam bentuk <i>Zip file</i> (kompresi)
<i>Fix Encoding & Reload</i>	Membatalkan perubahan <i>sketch</i> dan mengambil ulang <i>sketch</i> sebelumnya yang telah disimpan
<i>Serial Monitor</i>	Mengaktifkan jendela tampilan komunikasi <i>serial</i> pada komputer
<i>Board</i>	Menentukan jenis <i>board</i> arduino yang digunakan
<i>Serial Port</i>	Menentukan <i>port serial</i> yang digunakan untuk mengunggah program dan tersambung pada <i>board</i> arduino
<i>Burn Bootloader</i>	Memasukkan <i>bootloader</i> pada mikrokontroler yang ada pada <i>board</i> arduino melalui ICSP

5) Menu Help

Menu *Help* berisi bantuan/catatan/keterangan yang mungkin dibutuhkan jika ada pertanyaan.

Tabel 2.4. Pilihan pada Menu Help

<i>Sub Menu</i>	Keterangan
<i>Verify</i>	Untuk mengkompilasi program artinya mengkonversi program pada arduino menjadi informasi/data yang dapat dieksekusi/dibaca oleh mikrokontroler
<i>Upload</i>	Untuk mengunggah program ke dalam <i>board</i> arduino
<i>New</i>	Untuk membuat <i>file sketch</i> baru
<i>Open</i>	Untuk membuka <i>file sketch</i> yang sudah pernah dibuat
<i>Save</i>	Untuk menyimpan <i>sketch (list program)</i> yang sedang dibuat
<i>Serial Monitor</i>	Untuk mengaktifkan jendela komunikasi <i>serial</i> , dan <i>transfer data</i> (kirim/terima) antara <i>board</i> arduino dan komputer

2.6. Bahasa Pemrograman Arduino Berbasis C

Menurut Heri dan Aan (2015:45-57), Struktur dasar dalam pemrograman arduino sangatlah *simple* dan terdiri dari dua bagian, terdiri dari dua fungsi, yaitu fungsi persiapan (*setup()*) dan fungsi utama (*loop()*).

- *Setup()* adalah persiapan sebelum eksekusi program.
- *Loop()* adalah tempat menulis program utama yang akan dieksekusi.

Fungsi *setup()* digunakan untuk mendefinisikan variabel-variabel yang digunakan dalam program. Fungsi ini berjalan pertama kali ketika program dijalankan. Selanjutnya terdapat *loop()* adalah program inti/utama dari arduino yang dijalankan secara terus menerus baik pembacaan *input* maupun pengaktifan *output*. Program ini adalah inti dari semua program dalam arduino. Berikut adalah fungsi-fungsi dasar pada bahasa pemrograman arduino:

Tabel 2.5. Fungsi-fungsi dasar bahasa pemrograman arduino

No.	Fungsi-fungsi Dasar	Keterangan
1.	<i>Setup()</i>	Inisialisasi <i>mode pin</i> dan <i>serial</i>
2.	<i>Loop()</i>	Mengeksekusi fungsi terus menerus
3.	<i>Function</i>	Sekumpulan blok instruksi
4.	{ } kurung kurawal	Mengawali dan mengakhiri fungsi
5.	; (Titik koma)	Tanda akhir dari instruksi
6.	/*.....*/ Blok komentar	Memberi komentar program > 1 baris
7.	// Komentar	Memberi komentar per baris program
8.	Variabel	<i>Byte</i> : menyimpan data numerik 8 bit <i>Int</i> : menyimpan data angka 16 bit <i>Long</i> : menyimpan data angka 32 bit <i>Float</i> : data numerik memiliki nilai desimal
9.	<i>Array</i>	Kumpulan nilai-nilai yang diakses dengan nomor <i>indeks</i>
10.	Aritmatika	Penambahan, pengurangan, perkalian, dan pembagian
11.	Operasi Gabungan	Operasi aritmatika gabungan

12.	Operator Perbandingan	Membandingkan 2 konstanta atau variabel
13.	Operator Logika	Membandingkan 2 ekspresi <i>TRUE</i> atau <i>FALSE</i>
14.	Konstanta	Nilai-nilai yang telah ditetapkan
15.	<i>TRUE/FALSE</i>	Konstanta Boolean yang mendefinisikan nilai logika
16.	<i>HIGH/LOW</i>	Menentukan nilai <i>pin HIGH</i> atau <i>LOW</i>
17.	<i>Input/Output</i>	Menentukan <i>mode pin digital</i> sebagai <i>input</i> atau <i>output</i>
18.	<i>If</i>	Instruksi untuk menguji suatu kondisi telah tercapai
19.	<i>If....else</i>	Mengeksekusi instruksi lain jika suatu kondisi tidak terpenuhi
20.	<i>For</i>	Untuk mengulang suatu blok instruksi di dalam kurung kurawal
21.	<i>While</i>	Menjalan program secara terus menerus sehingga suatu kondisi salah atau <i>false</i>
22.	<i>Do....write</i>	Perintah melakukan terus menerus hingga mencapai suatu kondisi yang tidak memenuhi kondisi yang diinginkan
23.	<i>pinMode(pin,Mode)</i>	Inisialisasi <i>pin</i> sebagai <i>input</i> atau <i>output</i>
24.	<i>digitalRead(pin)</i>	Membaca <i>input</i> dari suatu <i>pin</i>
25.	<i>digitalWrite(pin,value)</i>	Memberi nilai <i>output HIGH</i> (1) atau <i>LOW</i> (0) pada <i>pin digital</i>
26.	<i>analogRead(pin)</i>	<i>Read</i> nilai <i>input analog</i> dengan resolusi 10 bit.
27.	<i>analogWrite(pin,value)</i>	Memberi nilai PWM (<i>pulse width modulation</i>) pada <i>output</i>
28.	<i>delay(ms)</i>	Memberi jeda sebelum lanjut ke program selanjutnya
29.	<i>millis()</i>	Mengambil nilai waktu dari program berjalan sampai berhenti
30.	<i>Tone</i>	Menghasilkan nada frekuensi
31.	<i>noTone(pin)</i>	Menghentikan nada frekuensi
32.	<i>randomSeed(seed)</i>	Mengambil nilai acak dengan <i>seed</i>

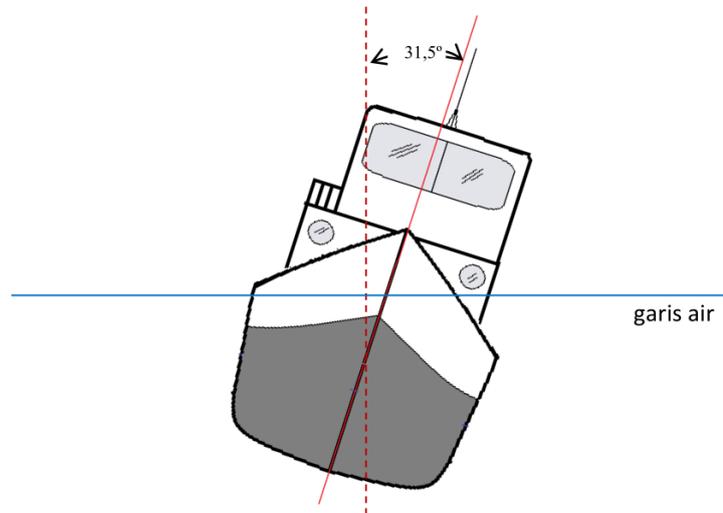
33.	<i>Random(min,max)</i>	Mengambil nilai acak <i>min</i> dan <i>max</i>
34.	<i>Serial.begin(rate)</i>	Membuka <i>port</i> data <i>serial</i>
35.	<i>Serial.print()</i>	Untuk mengirimkan data ke <i>serial port</i>
36.	<i>Serial.read()</i>	Menerima data dari <i>serial port</i>
37.	<i>Serial.available()</i>	Instruksi untuk mendeteksi apakah menerima data dari <i>serial port</i>
38.	<i>attachInterrupt(noInt, function, mode)</i>	Instruksi untuk mengaktifkan interupsi dari luar

2.7. APP Invertor

App *Inventor* adalah alat pengembangan yang digunakan untuk membangun aplikasi di *android*. Peranti ini diciptakan di MIT (*Massachusetts Institute of Tachnology*) dengan tujuan untuk memudahkan pembuatan aplikasi di *Android*. Bahasa pemrograman yang secara bawaan digunakan di *Android* adalah bahasa Java. Bahasa Java banyak digunakan untuk kepentingan pembuatan aplikasi di perangkat *Android*. MIT menciptakan alat pengembangan yang mudah digunakan dengan menggunakan pendekatan blok. (Abdul Kadir, 2017:5)

2.8. Stabilitas Kapal

Pada penelitian yang telah dilakukan oleh Yopi Novita (2011) berjudul Pengaruh *Free Surface* Terhadap Stabilitas Kapal Pengangkut Ikan Hidup menjelaskan bahwa jika kapal terdapat nilai $FA = 31,5^\circ$. Nilai ini menunjukkan bahwa KPIH (Kapal Pengangkut Ikan Hidup) memiliki *Floading Angle* (FA) sebesar $31,5^\circ$. *Floading angle* adalah sudut kemiringan kapal yang menjadi batas saat masuknya air laut ke atas dek kapal. Ilustrasi *floading angle* disajikan pada Gambar 2.9. Apabila KPIH mengalami kemiringan lebih dari $31,5^\circ$, maka dapat dipastikan lantai dek kapal akan mulai kebanjiran air laut, dan jika kondisi ini dibiarkan lambat laun kapal akan terbalik.

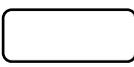
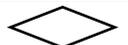
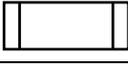
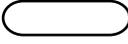
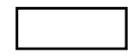


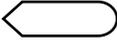
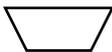
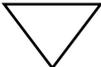
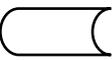
Gambar 2.6. Ilustrasi kondisi *loading angle* KPIH

2.9. Flowchart

Menurut I Gusti Nguah Suryantara (2009), badan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. (Hendra, 2009, Analisis dan Perancangan Sisfo; Dahliar Ananda, 2009, Algoritma dan Pemrograman)

Tabel 2.6. Simbol *Flowchart*

No.	Simbol	Nama Simbol	Keterangan
1.		<i>Alternate Process</i>	Menyatakan segala jenis operasi yang diproses dengan menggunakan mesin yang memiliki <i>keyboard</i>
2.		<i>Decision</i>	suatu penyelesaian kondisi dalam program
3.		<i>Data</i>	Mewakilik data <i>input</i> atau <i>output</i>
4.		<i>Predefined Process</i>	Suatu operasi yang rinciannya di tunjukkan di tempat lain
5.		<i>Document</i>	<i>Document input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer
6.		<i>Terminator</i>	Untuk menunjukkan awal dan akhir dari suatu proses
7.		<i>Process</i>	Kegiatan proses dari operasi program komputer

8.		<i>Manual Input</i>	<i>Input yang menggunakan online keyboard</i>
9.		<i>Conector</i>	Penghubung ke halaman yang masih sama
10.		<i>Off-Page Connector</i>	Penghubung ke halaman lain
11.		<i>Display</i>	<i>Output yang ditampilkan di monitor</i>
12.		<i>Delay</i>	Menunjukkan penundaan
13.		<i>Preparation</i>	Memberi nilai awal suatu besaran
14.		<i>Manual Operation</i>	Pekerjaan <i>manual</i>
15.		<i>Card</i>	<i>Input atau output yang menggunakan kartu</i>
16.		<i>Punch Tape</i>	<i>Input atau output menggunakan pita kertas berlubang</i>
17.		<i>Merge</i>	Penggabungan atau penyimpanan beberapa proses atau informasi sebagai salah satu
18.		<i>Dirrect Access Storage</i>	<i>Input atau output menggunakan drum magnetic</i>
19.		<i>Magnetic Disk</i>	<i>Input atau output menggunakan hard disk</i>
20.		<i>Sequential Access Storage</i>	<i>Input atau output menggunakan pita magnetic</i>
21.		<i>Sort</i>	Proses pengurutan data di luar komputer
22.		<i>Stored Data</i>	<i>Input atau output menggunakan diskette</i>
23.		<i>Extract</i>	Proses dalam jalur <i>paralel</i>
24.		<i>Arrow</i>	Menyatakan jalan atau arus suatu proses
25.		<i>Summing Junction</i>	Untuk berkumpul beberapa cabang sebagai proses tunggal
26.		<i>Or</i>	Proses menyimpang dalam dua proses