BAB II

DASAR TEORI

2.1 Visi Komputer (*Computer Vision*)

Visi komputer adalah ilmu yang mempelajari bagaimana komputer dapat mengenali objek yang akan diamati / diobservasi.Hal ini dilakukan bertujuan untuk dapat meniru visualisasi dari manusia yang diaplikasikan ke dalam komputer. Visi komputer terdiri dari 2 bidang ilmu diantaranya adalah:

1. Pengolahan Citra (*image Processing*)

   Pengolahan Citra merupakan salah satu ilmu yang mempelajari bagaimana pemperbaiki kualitas citra agar mendapatkan hasil citra yang baik dan mudah dikenali oleh manusia atau mesin.[2]

2. Pengenalan pola (*Pattern recognition*)

   Pengenalan pola merupakan proses pengelompokkan data numerik dan simbolik (termasuk citra) secara otomatis,yang bertujuan untuk dapat diidentifikasi objek pada citra.[2]

2.1.1 Model Warna RGB

Model warna RGB merupakan komponen yang terdiri dari warna dasar merah (*Red*), hijau (*Green*), biru (*Blue*) yang dapat dimodelkan atau di aplikasikan dengan cara mereproduksi susunan warna untuk mendapatkan warna yang bervariasi. Rentang nilai untuk warna RGB yaitu antara 0-255. Untuk dapat menghitung nilai warna dan menampilkannya dalam model RGB maka dapat dilakukan normalisasi dari ke tiga komponen warna.[3]

Di bawah ini adalah ilustrasi hasil pencampuran warna RGB :

Warna	Kombinasi
	R : 255, G : 255, B : 255
	R : 0, G : 0, B : 0
	R : 255, G : 0, B : 0
	R : 237, G : 255, B : 0
	R : 138, G : 233, B : 23
	R : 15, G : 0, B : 255

Gambar 2. 1 Model pencampuran RGB

2.1.2 Metode Threshold

Threshold merupakan salah satu metode proses pemisahan citra dari nilai derajat keabuan menjadi nilai biner (hitam putih), sehingga akan memudahkan proses identifikasi. Dengan menggunakan motode *threshold* yang semula hanya memisahkan derajat keabuan, dengan menggunakan software MATLAB maka didapatkan pemisahan citra yang beragam warna. Citra yang di *threshold* g(x,y) dapat didefinisikan seperti contoh di bawah :

$$g(x,y) = \begin{cases} 1\ Jika\ f(x,y) \geq T \\ 0\ Jika\ f\ (x,y) < T \end{cases}$$

T = *Threshold*

f(x,y)≥T disebut *object poin*[3]

2.1.3 Ektrasi Warna

Ekstraksi ciri citra merupakan tahapan mengekstrak ciri/informasi dari objek di dalam citra yang ingin dikenali/dibedakan dengan objek lainnya. Ciri yang telah diekstrak kemudian digunakan sebagai parameter untuk membedakan antara objek satu dengan lainnya pada tahapan identifikasi/ klasifikasi. Dan contoh untuk memilih warna dalam setiap RGB adalah:[4]

1. *red=gambar(:,:,1); %* memanggil matriks gambar yang hanya berisi piksel warna merah.
2. *green=gambar(:,:,2);%* memanggil matriks gambar yang hanya berisi piksel warna merah.
3. *blue=gambar(:,:,3); %* memanggil matriks gambar yang hanya berisi piksel warna merah.

2.1.4 *Morphology*

Morphology adalah teknik pengolahan citra digital dengan menggunakan bentuk sebagai pedoman dalam pengolahan. Nilai dari setiap pixel dalam citra digital hasil diperoleh melalui proses perbandingan antara pixel yang bersesuaian pada citra digital masukan dengan pixel tetangganya. Operasi morphologi bergantung pada urutan kemunculan dari pixel, tidak memperhatikan nilai numeric dari pixel sehingga teknik morphologi sesuai apabila digunakan untuk melakukan pengolahan *binary image* dan *grayscale image*.[4] Ada beberapa operasi *morphology* yang dilakukan pada sistem pendeteksian ini diantaranya :

1. Penutupan (*Closing*)

   Fungsi dari proses ini adalah untuk menghilangkan *noise* yang terdapat di sekitar target, selain menghilangkan *noise* proses ini juga berfungsi untuk menghaluskan kontur atau tepian target sehingga membuat target yang terdeteksi lebih halus.

2. Pengisian (*Filling*)

Fungsi dari operasi ini adalah melakukan pengisian pada bagian dalam atau tengah target sehingga menghasilkan segmen target yang tidak berlubang dan solid.

2.2 Komunikasi Data

Komunikasi merupakan suatu proses penyampaian berupa informasi dari satu tempat ke tempat lain. Sedangkan pengertian data merupakan suatu informasi yang disajikan dalam bentuk digital. Jadi, komunikasi data merupakan suatu proses pengiriman informasi berbentuk data digital dari dua tempat yang berbeda melalui saluran transmisi, bisa antara perangkat komputer dan komputer atau perangkat lainnya. Pada dasarnya untuk terjadinya sebuah komunikasi dibutuhkan minimal dua perangkat alat dan terdapat media transmisinya.



Gambar 2. 2 Blok diagram komunikasi sederhana

Keterangan untuk gambar diatas adalah :

a. Pengirim

Pengirim berfungsi sebagai alat/ media yang nantinya akan mengirimkan informasi berupa data ke media transmisi.

b. Media Transmisi

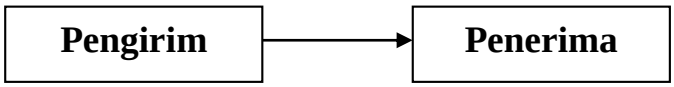
Media transmisi berfungsi sebagai alat/ media pembawa untuk menyam-paikannya informasi yang dituju.

c. Penerima

Penerima berfungsi sebagai alat/ media yang dikirim/ tujuan.

2.2.1 Aliran Komunikasi Data

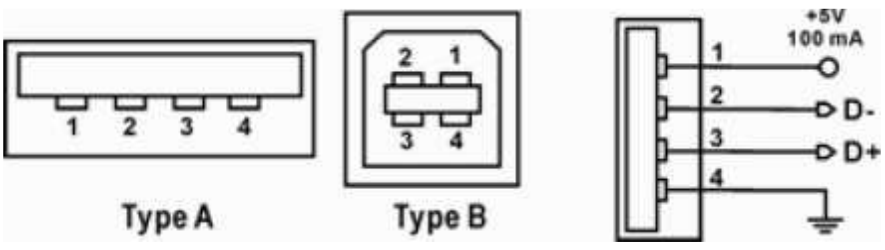
Aliran komunikasi data yang dipakai pada robot navigasi ialah komunikasi data *simplex* dimana hanya terjadi komunikasi satu arah.Disini yang berperan sebagai media pengirim ialah program matlab sedangkan mikrokontroler bisa menjadi media pengirim dan penerima karena mikrokontroler nantinya akan meniram data dari program matlab dan akan kembali mengirimkan data ke driver motor.



Gambar 2.3 Komunikasi satu arah

2.2.2 Komunikasi Serial *Universal Serial Bus* (USB)

Universal Serial Bus (USB) adalah perangkat yang digunakan sebagai perangkat penghubung antara piranti *hardware* dengan komputer. Perkembangan perangkat USB saat ini sudah ada 3 versi. Setiap versi memiliki kecepatan transfer yang berbeda-beda, yang bertujan untuk meningkatkan efisien waktu pengiriman datanya. Untuk konektor USB ada 2 jenis konektor yaitu konektor tipe A dan konektor tipe B, anya konektor yang sering digunakan untuk mengkoneksikan perangkat hardware dengan komputermenggunakan konektor tipe A.



Gambar 2.4 Konektor USB tipe A dan tipe B

Standarisasi pin USB tipe A dan tipe B ialah sebagai berikut:

Tabel 2. 1 Konfigurasi saluran data konektor USB

Pin	Nama	Warna Kabel	Keterangan
1	VBus	Merah	+5Vdc
2	Data-	Putih	Data -
3	Data+	Hijau	Data +
4	Ground	Hitam	Ground

Keterangan mengenai saluran konektor USB ialah sebagai berikut:

1. VBus = Saluran *input* tegangan sebesar +5 VDC/ 100 mA.
2. Data - = Saluran data, data akan aktif ketika data - > data +.
3. Data + = Saluran data, data akan aktif apabila data + > data –.
4. Ground = Saluran *Ground.*

2.2.3 Komunikasi Serial *Inter Integrated Circuit* (I2C)

Komunikasi *Inter Integrated Circuit* (I2C) merupakan komunikasi serial antara dua arah yang menggunakan dua saluran terdiri dari *Serial Data* (SDA) dan *Serial Clock* (SCL) perangkat yang terhubung dengan sistem I2C bus dapat di operasikan sebagai *Master* dan *Slave*. *Master* ini yang berfungsi sebagai pengendalisaat mentransfer, untuk *slave* berfungsi sebagai yang dikendalikan oleh *master*.

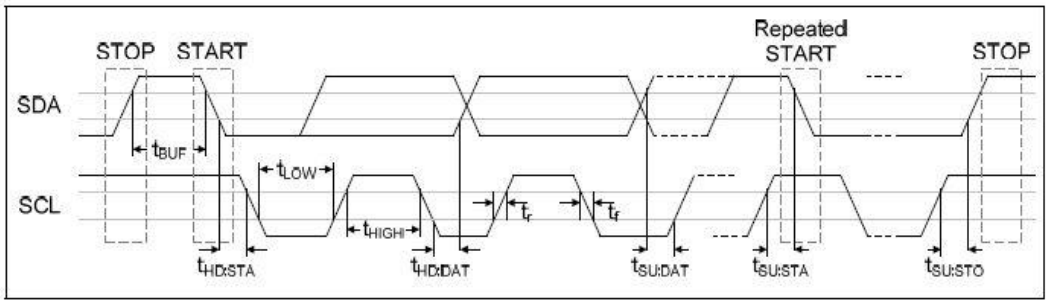


Figure 5-18 I²C Bus Timing

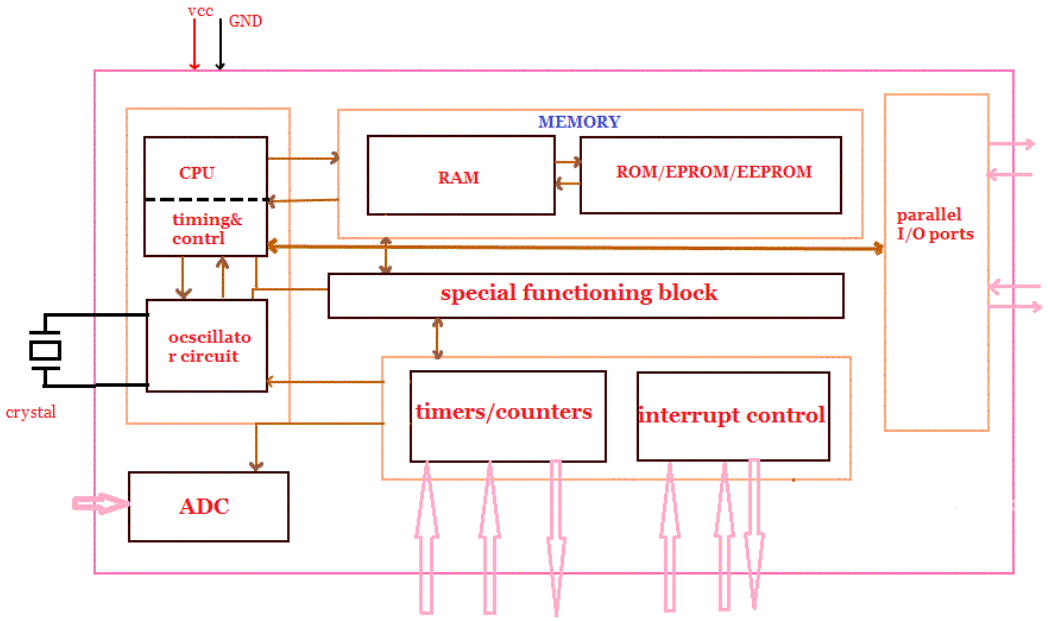
Gambar 2.5 Prinsip kerja protokol bus I2C

Penjelasan gambar di atas dapat di jelaskan sebagai berikut :

Bit pertama *master* akan mengirimkan berupa alamat *slave*, indikator akan mengecek apakah ada bit untuk baca/ tulis. Setelah itu *slave* akan mengirimkan bit *acknowledge*, dan diteruskan untuk mengirim data bit dari *slave* ke *master*. *Master* akan menerima bit dari *slave*, kemudian mengirimkannya kembali bit *acknowledge*. Bit kedua dari terakhir *master* akan mengirimkan sinyal *not acknowledge*, kemudian *master* mengirimkan sinyal bit berhenti.[5]

2.3 Mikrokontroler

Mikrokontroler adalah sebuah komputer kecil yang terdapat di dalam sebuah rangkaian terintegrasi. Mikrokontroler pada dasarnya terdiri CPU (*Central Processing Unit*), *memory*, *input*/*output ports*, *timers and counters*, *interrupt controls*, *analog to digital converters*, *serial interfacing ports* dan *oscillatory circuits*.[5]



Gambar 2.6 Blok Diagram *Microcontroller* [5]

Penjelasan singkat mengenai blok diagram *microcontroller* :

a. CPU (*Central Processing Unit*)

CPU adalah otak dari sebuah *microcontroller*. CPU bertugas untuk mengambil setiap intruksi dalam bentuk kode dan melakukan *decode* (menterjemahkan intruksi) ke dalam bahasa mesin untuk selanjutnya dilakukan eksekusi. CPU juga bertugas untuk menghubungkan setiap bagian dari *microcontroller* ke dalam sebuah sistem.

b. *Memory*

Fungsi dari sebuah *memory* adalah menyimpan setiap intruksi dan data dari sebuah program. *Microcontroller* biasanya memiliki sejumlah memori seperti RAM, ROM/EPROM/EEPROM dan *flash memory*.

c. *Parallel Input/Output Ports*

Parallel input/output ports digunakan untuk melakukan *interface* dengan perangkat lain yang dibutuhkan untuk keperluan sebuah sistem seperti LED, LCD, motor servo, *fingerprint module* dan lain-lain.

d. *Serial Ports*

Serial port digunakan untuk keperluan berbagai *interface* serial antara mikrokontroler dan perangkat lain seperti halnya *parallel port*.

e. *Timers / Counters*

Timers / counters merupakan salah satu fungsi yang sangat berguna dari mikrokontroler. Mikrokontroler memiliki lebih dari satu *timer* dan *counter*. Operasi utama dari *timer* dan *counter* adalah melakukan *clock function* , modulasi, *pulse generations*, pengukuran frekuensi, membuat osilasi, dan lain-lain.

f. ADC (*Analog to Digital Converter*)

ADC berfungsi untuk mengkonversi sinyal analog menjadi digital. Konversi ini dilakukan dengan tujuan mendapatkan output berupa sinyal digital, salah satu contohnya adalah pengukuran suhu yang hasilnya ditampilkan pada sebuah LCD.

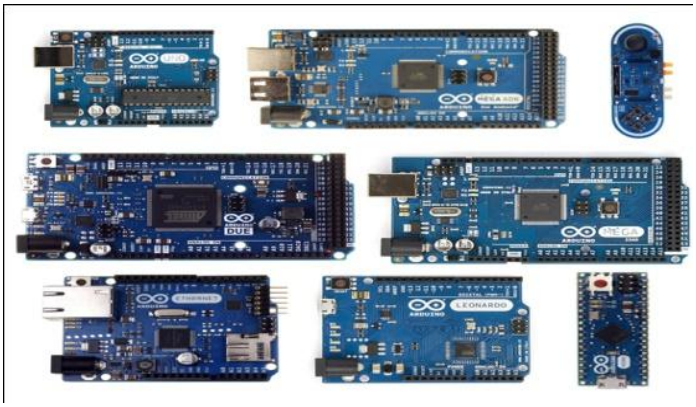
g. *Interrupt Control*

Interrupt control digunakan untuk melakukan interupsi pada sebuah program ketika dieksekusi. *Interrupt control* memiliki dua jenis, yaitu *interrupt control internal* (memakai intruksi interupsi) dan *interrupt control eksternal* (memakai akses pin interupsi).

h. *Special Functioning Block*

Special functioning block merupakan bagian tambahan pada sebuah *mikrokontroller* untuk kebutuhan tertentu. Tidak semua *mikrokontroller* menggunakan *special functioning block.*

Salah satu *microcontroller* yang paling banyak digunakan pada jaman sekarang adalah *arduino. Arduino* adalah sebuah platform komputasi fisik yang bersifat *open-source* berbasis papan/board mikrokontroler sederhana. Hardwarenya menggunakan prosesor *atmel* AVR dan softwarenya memiliki bahasa pemrograman sendiri yaitu C++ *arduino*. Salah satu alasan kenapa *arduino* banyak dipakai adalah karena sistem pengoperasiannya yang cukup mudah. Bahasa pemograman yang mudah dipahami dan perangkat tambahan yang mudah didapatkan menjadi daya tarik tersendiri. Arduino memiliki jenis-jenis yang beragam. Jenis-jenis ini memiliki perbedaan dalam segi bentuk, spesifikasi dan performansi, secara singkatnya dijelaskan pada gambar 2.7 di bawah ini



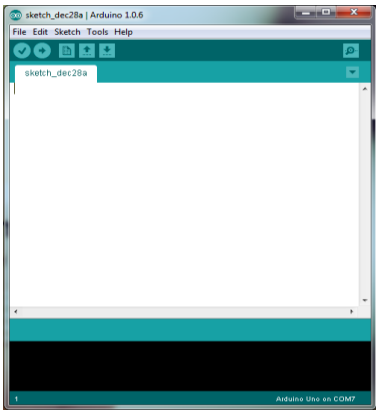
Gambar 2.7 Bentuk Fisik *Arduino* [6]

	Processor	Processor Voltage	Supply Voltage	Flash	SRAM	Digital I/O Pins	PWM Pins	Analog Inputs	Hardware Serial Ports	Dimensions
Uno	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm
Uno Ethernet	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm
Mega	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm
Mega ADK	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm
Leonardo	16MHz Atmega 32U4	5v	7-12v	32Kb	2.5Kb	20*	7	12*	1	2.1"x2.7" 53x75mm
Due	84MHz ARM SAM3X8E	3.3v	7-12v	512Kb	96Kb	54	12	12	4	2.1"x4" 53x102mm
Micro	16MHz Atmega 32U4	5v	5v	32Kb	2.5Kb	20*	7	12*	1	0.7"x1.9" 18x49mm
Flora	8MHz Atmega 32U4	3.3v	3.5-16v	32Kb	2.5Kb	8*	4	4*	1	1.75" dia 44.5mm dia

Gambar 2.8 Tabel Spesifikasi *Arduino* [6]

2.3.1 *Arduino* IDE (Integrated Development Environment)

Arduino IDE adalah sebuah software untuk menulis program, mengedit program sampai mengirim program tersebut ke *arduino.* Program *arduino* sering disebut *sketch.*



Gambar 2.9 *Arduino* IDE

2.3.2 Bahasa Pemograman *Arduino*

Bahasa Pemograman arduino sering disebut C++ *arduino* karena struktur pemograman dan tipe datanya mirip dengan C++ pada umumnya. C++ *arduino* terbagi menjadi 3 bagian utama, *structure, variables* dan *function*.[5]

1) *Structure*

   a. *Main Structure*

   - *void setup*( ), nama fungsi yang telah disediakan oleh *arduino uno* untuk menyatakan fungsi yang akan dijalankan pertama kali. Fungsi ini berisi kode-kode untuk kepentingan inisialisasi.
   - *void loop*( ) , fungsi yang secara otomatis dijalankan oleh *arduino uno* setelah fungsi *setup* ( ) dieksekusi. Seluruh kode yang ada di fungsi dengan sendirinya akan diulang terus-menerus. Satu-satunya yang bisa menghentikan eksekusi *loop* ( ) adalah berhentinya input catu daya ke papan *arduino uno.*

   b. *Control Structures*

   - *If….else*, format pemogramannya seperti di bawah ini :

     if (kondisi)

     {aksi A}

     else

     {aksi B}

     Jika kondisi *true* maka output akan melakukan aksi A, jika kondisi *false* output akan melakukan aksi B.

   - *for*, contoh format pemogramannya seperti di bawah ini :

     for (inisialisasi ; kondisi ; penambahan / pengurangan)

contohnya : for (int i=0; i <= 255; i++){ analogWrite(PWMpin, i); delay(10); }

loop akan terus berlangsung sampai kondisi terpenuhi, selama loop berlangsung kondisi akan terus dilakukan penambahan/pengurangan.

c. Aturan *Syntax*

Tabel 2.2 Aturan *Syntaq*

Syntax	Keterangan
; (titik koma)	Akhir dari setiap *statement*
{} (kurung kurawal)	Batas awal dan akhir dari sebuah program.
// (dua garis miring)	Memberi komentar pendek pada sebuah program.
/* */ (garis miring bintang)	Memberi komentar pendek pada sebuah program.
#define	Mendeskripsikan sebuah variabel sehingga dikenali oleh arduino.
#include	Memasukan *library* pada sebuah program.

d. Operator Aritmatika

Tabel 2.3 Operator Aritmatika

Operator	Keterangan
+	Penjumlahan dan tanda positif.
-	Pengurangan dan tanda negatif
/	Pembagian.
*	Perkalian.

e. Operator Pembanding

Tabel 2.4 Operator Pembanding

Operator	Keterangan
==	Operator “sama dengan”
!=	Operator “tidak sama dengan”
>	Operator “lebih dari”
>=	Operator “lebih dari atau sama dengan
<	Operator “kurang dari”

f. Operator *Boolean*

Tabel 2.5 Operator *Boolean*

Operator	Keterangan
&&	Operator “AND”
||	Operator “OR”
!	Operator “NOT”

2) *Variable*

a. Tipe Data

Tabel 2.6 Tipe Data C++ *Arduino*

Tipe Data	Keterangan	Kebutuhan Memori
Boolean	Tipe Boolean hanya dapat digunakan untuk menamp-ung dua nilai saja *true* atau *false*	1 byte

Char	Tipe *char* berguna untuk menyimpan sebuah nilai karakter seperti ‘A’,’9’ dan ‘*’.	1 byte
Unsigned Char	Sama dengan tipe *char*, tetapi bilangan akan dikodekan dalam bentuk bilangan positif antara 0 sampai dengan 255.	1 byte
byte	Tipe data ini berguna untuk menampung bilang- an bulat yang berkisar antara 0 sampai 255.	1 byte
Int	Tipe data ini berguna untuk menampung bilang-an bulat yang berkisar antara -32768 sampai 32767.	2 byte
Unsigned Int	Tipe data ini berguna untuk menampung bilang- an bulat yang berkisar antara 0 sampai 65535.	2 byte
Word	Tipe *word* identik dengan *usigned int.*	2 byte
Long	Tipe data ini berguna untuk menampung bilang-an bulat yang berkisar antara -2,147,483, 648 sampai 2,147,483,647	4 byte
Unsigned Long	Tipe data ini berguna untuk menampung bilang-an bulat yang berkisar antara 0 sampai 4,294,967,295 (2^{23-1})	4 byte
Float	Tipe data ini berguna untuk menampung bilang-an real. Angka yang disimpan dari -3,4028235 E+38 sampai 3,4028235 E+38.	4 byte

b. Konversi Tipe Data

Tabel 2.7 Fungsi-fungsi untuk konversi data

Konversi	Keterangan

char( )	Konversi nilai argument tipe *char*( ) menjadi tipe *char.*
byte( )	Konversi nilai argument tipe *byte*( ) menjadi tipe *byte.*
int( )	Konversi nilai argument tipe *int*( ) menjadi tipe *int.*
word( )	Konversi nilai argument tipe *word*( ) menjadi tipe *word.*
long( )	Konversi nilai argument tipe *long*( ) menjadi tipe *long.*
float( )	Konversi nilai argument tipe *float*( ) menjadi tipe *float.*

3) *Function*

a. Digital I/O

Tabel 2.8 Konfigurasi Digital I/O

Digital I/O	Keterangan	Format Program
pinMode( )	Mengkonfigurasi mode pin tertentu pada *arduino.* Mode yang dipilih bisa *imput, output* dan	pinMode(Pin,Mode)
digitalWrite( )	Mengkondisikan *high* atau *low* pada sebuah pin digital.	digitalWrite(Pin,Value)
digitalRead( )	Membaca *high* atau *low* pada sebuah pin digital.	digitalWrite(Pin)

b. Analog I/O

Tabel 2.9 Konfigurasi Analog I/O

Analog I/O	Keterangan	Format Program
analogReference( )	Mengkonfigurasi mode tegangan referensi untuk input analog. Mode yang bisa dipilih adalah	analogReference(Type)

	DEFAULT, INTERNAL, INTERNAL1V1, INTERNAL2V56 dan EXTERNAL	
analogWrite()	Menulis nilai analog pada sebuah pin analog.	analogWrite(Pin,Value)
analogRead()	Membaca nilai pada sebuah pin analog.	analogWrite(Pin)

c. Fungsi Matematika

Tabel 2.10 Fungsi Matematika

Fungsi	Keterangan
min(x,y)	Menghitung nilai minimum dari x dan y.
max(x,y)	Menghitung nilai maximum dari x dan y.
abs(x)	Memberikan nilai absolut x.
pow(x,y)	Memberikan nilai balik berupa x^y. Hasilnya mempunyai tipe *float.*
sin(x)	Memberika nilai balik berupa sinus x. Argumen x dinyatakan dalam radian. Hasilnya mempunyai tipe *float.*
cos(x)	Memberika nilai balik berupa cosinus x. Argumen x dinyatakan dalam radian. Hasilnya mempunyai tipe *float.*
tan(x)	Memberika nilai balik berupa tangent x. Argumen x dinyatakan dalam radian. Hasilnya mempunyai tipe *float.*

2.4 Perangkat Keras

Berikut dibawah ini akan dijelaskan mengenai landasan teori dan fungsi komponen-komponen yang digunakan pada perangkat yang akan di rancang :

2.4.1 *Modul Adjustable DC-DC Step Up XL6009*

Modul step up atau penaik tegangan *DC (boost) XL6009* ini berfungsi sebagai penyelesaian masalah perbedaan tegangan yang dibutuhkan dengan yang tersedia. Seringkali dalam pembuatan rangkaian elektronika atau modul-modul mikrokontroler terdapat perbedaan tegangan kerja antar modul sehingga memerlukan sebuah modul regulator untuk menyesuaikan tegangan.

Modul step up *DC to DC XL6009* ini membantu untuk menaikkan tegangan ke tegangan yang lebih tinggi.

Adapun karakteristik dari modul *DC to DC XL6009* yaitu:

Input voltage : DC 3.5 V - 18 V

Output voltage : DC 4 V - 24 V (tegangan output harus lebih tinggi dari input)

Arus *input max* : 4 A

Ukuran *board* : 43 mm x 30 mm x 12 mm

Modul *regulator* penaik tegangan ini mengunakan bahan *solid capacitor* dan PCB berkualitas untuk menjamin kualitas tegangan yang dibutuhkan.



Gambar 2.10 *DC-DC Step Up XL6009*

2.4.2 Kamera *Webcam*

Kamera *Webcam* merupakan komponen utama yang digunakan sebagai menangkap objek gambar. Resolusi dari kamera merupakan bagian yang menentukan dari kualitas gambar yang akan dihasilkan. Kamera ini bisa dihubungkan dengan perangkat komputer karena memiliki komunikasi *serial* USB.



Gambar 2. 11 Kamera *Webcam* C170

Dengan menggunakan perangkat komputer, kamera *webcam* ini dapat mengakses gambar nyata lalu ditampilkan ke dalam monitor, dengan kamera *webcam* ini digunakan untuk keperluan media sosial internet dengan beragam jenis dari kamera dan resolusinya tinggi, maka gambar yang dihasilkan akan semakin baik.

2.4.3 *Motor DC*

Motor Listrik DC atau DC Motor adalah suatu perangkat yang mengubah energi listrik menjadi energi kinetik atau gerakan (motion). Motor DC ini juga dapat disebut sebagai Motor Arus Searah. Seperti namanya, DC Motor memiliki dua terminal dan memerlukan tegangan arus searah atau DC (Direct Current) untuk dapat menggerakannya. Motor Listrik DC ini biasanya digunakan pada perangkat-perangkat Elektronik dan listrik yang menggunakan sumber listrik DC seperti Vibrator Ponsel, Kipas DC dan Bor Listrik DC.



Gambar 2.12 *Motor DC*

Motor Listrik DC atau DC Motor ini menghasilkan sejumlah putaran per menit atau biasanya dikenal dengan istilah RPM (Revolutions per minute) dan dapat dibuat berputar searah jarum jam maupun berlawanan arah jarum jam apabila polaritas listrik yang diberikan pada Motor DC tersebut dibalikan. Motor Listrik DC tersedia dalam berbagai ukuran rpm dan bentuk. Kebanyakan Motor Listrik DC memberikan kecepatan rotasi sekitar 3000 rpm hingga 8000 rpm dengan tegangan operasional dari 1,5V hingga 24V.

Apabila tegangan yang diberikan ke Motor Listrik DC lebih rendah dari tegangan operasionalnya maka akan dapat memperlambat rotasi motor DC tersebut sedangkan tegangan yang lebih tinggi dari tegangan operasional akan membuat rotasi motor DC menjadi lebih cepat. Namun ketika tegangan yang diberikan ke Motor DC tersebut turun menjadi dibawah 50% dari tegangan operasional yang ditentukan maka Motor DC tersebut tidak dapat berputar atau terhenti. Sebaliknya, jika tegangan yang diberikan ke Motor DC tersebut lebih tinggi sekitar 30% dari tegangan operasional yang ditentukan, maka motor DC tersebut akan menjadi sangat panas dan akhirnya akan menjadi rusak.

2.4.4 *Driver Motor L298N*

Driver motor digunakan untuk mengontrol arah putaran dan kecepatan motor DC yang merupakan penggerak utama dari rangkaian tugas akhir ini.IC driver motor L298 yang didalamnya terdapat rangkaian *H-Bridge* akan mengontrol putaran motor sesuai data masukan digital yang berasal dari PLC Zelio SR2 B201 BD,dan pada IC L298 ini juga terdapat pin untuk pengaturan aplikasi PWM *(Pulse Width Modulator)* yang akan mengatur kecepatan motor dc yang dikendalikannya.L298 memiliki rangkaian dual *H-Bridge*, sehingga mampu mengendalikan dua buah motor DC sekaligus.



Gambar 2. 13 *Driver Motor L298N*

Berikut adalah Karakteristik dari driver motor L298N :

1. Tegangan operasi supply sampai dengan 36 Volt.
2. Total arus DC sampai dengan 1A.
3. Tegangan logic ”0” sampai dengan 1,5 Volt.
4. Memiliki dua Enable input.

Fungsi dari tiap-tiap pin driver motor L298 adalah sebagai berikut:

1. Output 1 dan Output 2 (pin 3 dan pin 6)Pin ini merupakan output untuk bridge-A
2. Vs (pin 8)Merupakan pin supply tegangan untuk output.
3. Input 1 dan Input 2 (pin 2 dan pin 7)Pin ini digunakan untuk mengontrol bridge-A
4. Enable 1 dan Enable 2 (pin 1 dan pin 9)Pin ini berfungsi untuk mengaktifkan dan menonaktifkan bridge A dan bridge B.
5. Ground (pin 4, 5, 12, dan 13)Berfungsi sebagai grounding rangkaian driver.
6. Vss (pin 16)Pin ini berfungsi sebagai supply logic untuk driver.
7. Input 3 dan Input 4 (pin 10 dan 15)Berfungsi sebagai masukan pada bridge B.
8. Output 3 dan Output 4 (11 dan 14)

2.4.5 *Battry Lipo*

Baterai Lithium Polimer atau biasa disebut dengan LiPo merupakan salah satu jenis baterai yang sering di gunakan dalam dunia RC. Baterai LiPo tidak menggunakan cairan sebagai elektrolit melainkan menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastik film tipis. Lapisan film ini disusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. Dengan metode ini baterai LiPo dapat dibuat dalam berbagai bentuk dan ukuran. Diluar dari kelebihan arsitektur baterai LiPo, terdapat juga kekurangan yaitu lemahnya aliran pertukaran ion yang terjadi melalui elektrolit polimer kering. Hal ini menyebabkan penurunan pada charging dan discharging rate. Masalah ini sebenarnya bisa diatasi dengan memanaskan baterai sehingga menyebabkan pertukaran ion menjadi lebih cepat, namun metode ini dianggap tidak dapat untuk diaplikasikan pada keadaan sehari-hari. Seandainya para ilmuwan dapat

memecahkan masalah ini maka risiko keamanan pada batera jenis lithium akan sangat berkurang.

Pada baterai jenis NiCad atau NiMH tiap sel memiliki 1,2 volt sedangkan pada baterai Lipo memiliki rating 3,7 volt per sel. Keuntungannya adalah tegangan baterai yang tinggi dapat dicapai dengan menggunakan jumlah sel yang lebih sedikit.

Pada setiap paket baterai LiPo selain tegangan ada label yang disimbolkan dengan “S”. Disini “S” berarti sel yang dimiliki sebuah paket baterai (battery pack). Sementara bilangan yang berada didepan simbol menandakan jumlah sel dan biasanya berkisar antar 2-6S (meskipun kadang ada yang mencapai 10S). Berikut adalah beberapa contoh notasi baterai LiPo.

- 3.7 volt battery = 1 cell x 3.7 volts
- 7.4 volt battery = 2 cells x 3.7 volts (2S)
- 11.1 volt battery = 3 cells x 3.7 volts (3S)
- 14.8 volt battery = 4 cells x 3.7 volts (4S)
- 18.5 volt battery = 5 cells x 3.7 volts (5S)
- 22.2 volt battery = 6 cells x 3.7 volts (6S)

Kapasitas baterai menunjukkan seberapa banyak energi yang dapat disimpan oleh sebuah baterai dan diindikasikan dalam miliampere hours (mAh). Notasi ini adalah cara lain untuk mengatakan seberapa banyak beban yang dapan 34 diberikan kepada sebuah baterai selama 1 jam, dimana setelah 1 jam baterai akan benar-benar habis.

Sebagai contoh sebuah baterai RC LiPo yang memiliki rating 1000 mAh akan benar-benar habis apabila diberi beban sebesar 1000 miliampere selama 1 jam.

Apabila baterai yang sama diberi beban 500 miliampere, maka baterai akan benar-benar habis setelah selama 2 jam. Begitu pun apabila beban ditingkatkan menjadi 15.000 miliampere (15 Amps) maka energi di dalam baterai akah habis terpakai setelah selama 4 menit saja. (15 Amp merupakan jumlah beban yang umum digunakan pada RC kelas 400).



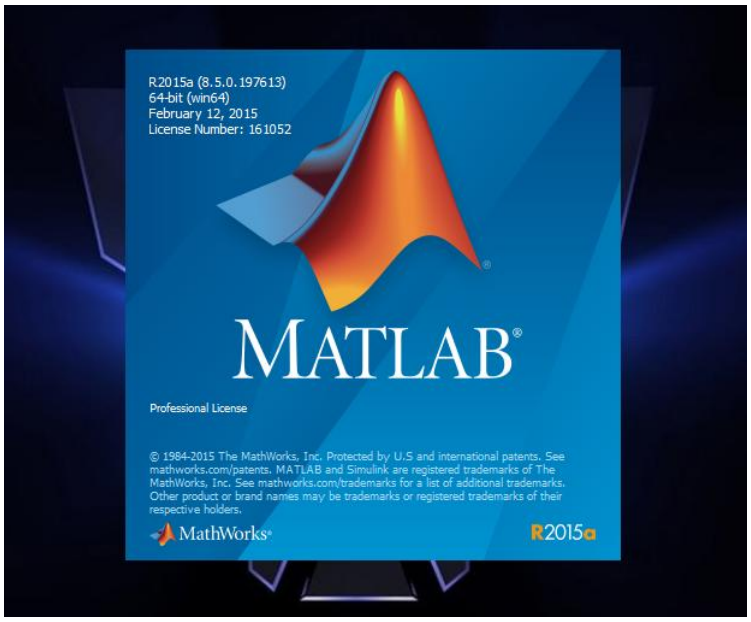
Gambar 2. 14 Baterai Lipo

Seperti yang telah dijelaskan, dengan beban arus yang begitu besar maka merupakan sebuah keuntungan apabila menggunakan baterai dengan kapasitas yang lebih besar (misal 2000 mAh). Dengan begitu maka waktu discharge akan meningkat menjadi 8 menit.

2.5 Perangkat Lunak MATLAB R2015a (*Software*)

Dalam proses pendeteksian target pada tugas akhir ini menggunakan sebuah *Graphical User Interface* (GUI).[7] GUI adalah sebuah aplikasi dari MATLAB yang mengandung tugas, perintah, atau komponen program yang berfungsi sebagai media bantu pengontrol untuk lebih mempermudah *user* atau pengguna dalam mengoprasikan sebuah program dalam MATLAB.[8] GUI pada

tugas akhir ini dibuat dengan MATLAB R2015a seperti yang terlihat pada gambar di bawah ini.



Gambar 2. 15 MATLAB R2015a