BAB II

DASAR TEORI

2.1 Sampah

Sampah merupakan material sisa yang tidak diinginkan setelah berakhirnya suatu proses. Sampah didefinisikan oleh manusia menurut derajat keterpakaiannya, dalam proses-proses alam sebenarnya tidak ada konsep sampah, yang ada hanya produk-produk yang dihasilkan setelah dan selama proses alam tersebut berlangsung. Akan tetapi karena dalam kehidupan manusia didefinisikan konsep lingkungan maka sampah dapat dibagi menurut jenis-jenisnya.

1. Sampah organik - dapat diurai *(degradable)*

   Sampah Organik, yaitu sampah yang mudah membusuk seperti sisa makanan, sayuran, daun-daun kering, dan sebagainya. Sampah ini dapat diolah lebih lanjut menjadi kompos.

2. Sampah anorganik - tidak terurai *(undegradable)*

   Sampah Anorganik, yaitu sampah yang tidak mudah membusuk, seperti plastik wadah pembungkus makanan, kertas, plastik mainan, botol dan gelas minuman, kaleng, kayu, dan sebagainya. Sampah ini dapat dijadikan sampah komersil atau sampah yang laku dijual untuk dijadikan produk laiannya. Beberapa sampah anorganik yang dapat dijual adalah plastik wadah pembungkus makanan, botol dan gelas bekas minuman, kaleng, kaca, dan kertas, baik kertas koran, HVS, maupun karton.

2.2 Visi Komputer (*Computer Vision*)

Visi komputer merupakan 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 yaitu.

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.[1]

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.[1]

2.2.1 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.

Berikut 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 [1]

Dapat dilihat gambar dari pencampuran warna diatas setiap warna memiliki rentang nilai dari 0-255, tingkat kehalusan dari sebuah citra dapat dilihat dari besar kecilnya satuan bit yang ditampilkan. Sebuah citra dengan ukuran 8 Bit memiliki jumlah warna sebesar 2^8= 256 warna.

2.2.2 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 MATLAB maka didapatkan pemisahan citra yang beragam warna. Citra yang di *threshold* g(x,y) dapat didefinisikan sebagai berikut :

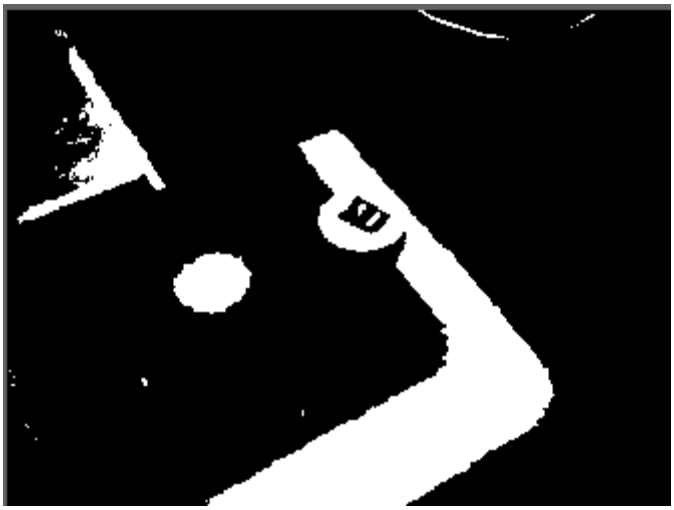
$$g(x,y) = \begin{cases} 1\ Jika\ f(x,y)\ \geq T \\ 0\ Jika\ f\ (x,y) < T \end{cases} \qquad \text{…….. (2.1)}$$

T = *Threshold*

f(x,y)≥T disebut *object point*



Gambar 2.2 Citra asli



Gambar 2.3 Citra hasil thresholding

Gambar 2.2 merupakan gambar asli yang belum di *threshold* dan Gambar 2.3 merupakan gambar hasil dari citra RGB yang di rubah didalam *image processing* pada MATLAB.

2.2.3 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

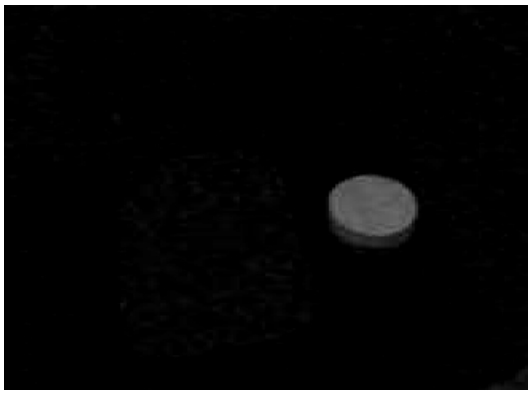
antara objek satu dengan lainnya pada tahapan identifikasi/ klasifikasi. Dan contoh untuk memilih warna dalam setiap RGB ialah :

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.

Contoh ektrasi warna pada warna biru :



Gambar 2.4 Citra asli untuk ektrasi



Gambar 2.5 Hasil Ektrasi warna biru

2.2.4 *Morphology*

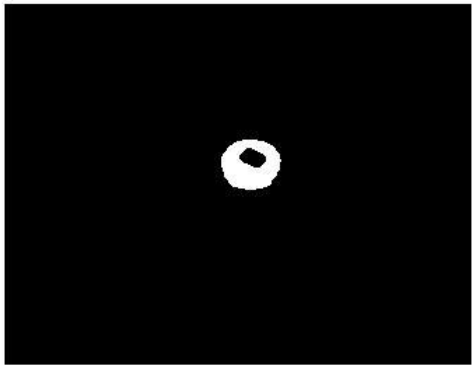
Morphology adalah teknik pengolahan citra digital dengan menggunakan bentuk sebagai pedoman dalam pengolahan. Nilai dari setiap pixel dalam citra digital hasil diperoleh melalui proses perbandingan antara pixel yang bersesuaian pada citra digital masukan dengan pixel tetangganya. Operasi morphologi bergantung 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*. 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.



Gambar 2.6a Citra hasil *threshold* sebelum proses *closing*



Gambar 2.7b Hasil proses *closing*

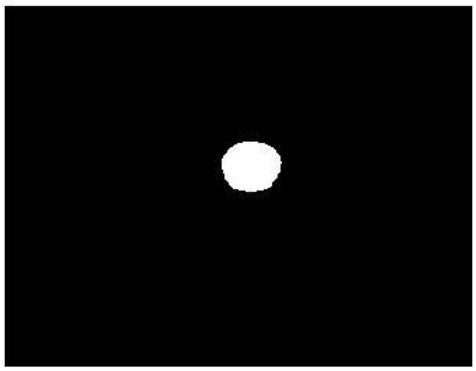
Dari gambar 2.6a dan 2.7b di atas dapat dilihat perbedaan sebelum dan sesudah proses *closing*, pada gambar 2.6a sebelum proses *closing* begitu banyak *noise* yang terdeteksi tetapi setelah proses *closing, noise* yang ada dihilangkan sehingga membuat target menjadi lebih baik dimana penutupan satu set (citra biner) A oleh penataan elemen B adalah erosi pelebaran set itu :

$$A \bullet B = (A \oplus B) \ominus B,$$

Dimana $\oplus$ dan $\ominus$ menunjukkan pelebaran dan erosi , masing-masing. Dalam pengolahan citra , penutupan , bersama-sama dengan pembukaan , pekerja keras dasar penghapusan kebisingan morfologi . Pembukaan menghilangkan benda-benda kecil , sedangkan penutupan menghilangkan lubang-lubang kecil .[3]

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. Pada gambar 2.7b akan dilakkan pengisian atau *filling.*



Gambar 2.8 Citra hasil proses *filling*

Hasil pengisian dimana sebelmnya pada gambar 2.8b titik tengah memiliki lubang sehingga pada proses *filling* ini menghasilkan bentuk citra lebih baik sehingga objek dapat terlihat sepenuhnya. Dan rumus *filling* ialah :

$$f_m(x,y) = \begin{cases} 1 - f(x,y) & jika\ (x,y) adalah\ border\ f \\ 0 & lainya \end{cases} \quad \ldots\ldots\ldots \quad (2.2)$$

Dimana setiap aplikasi ditentukan oleh pemilihan marker dan mask citra. Dipilih citra f_m yang dilakukan adalah menjadikan 0 disetiap tempat kecuali border citra,dimana posisi tersebut diset menjadi 1 – *f*. [3]

2.3 Komunikasi Data

Komunikasi merupakan proses penyampaian berupa informasi dari satu tempat ke tempat lain. Sedangkan pengertian data merupakan informasi yang disajikan dalam bentuk digital. Jadi, komunikasi data merupakan 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 komunikasi dibutuhkan minimal dua perangkat alat dan terdapat media transmisinya.[1]



Gambar 2. 9 Blok diagram komunikasi sederhana

Keterangan untuk gambar diatas yaitu :

a. Pengirim

   Pengirim berfungsi sebagai alat/ media yang 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

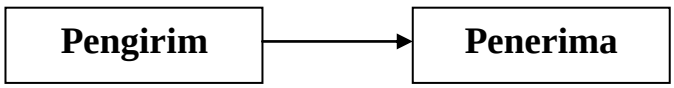
   Penerima berfungsi sebagai alat/ media yang dikirim/ ditujuan.

2.3.1 Aliran Komunikasi Data

Aliran komunikasi data terbagi menjadi tiga kategori, yaitu sebagai berikut :

1. Simplex

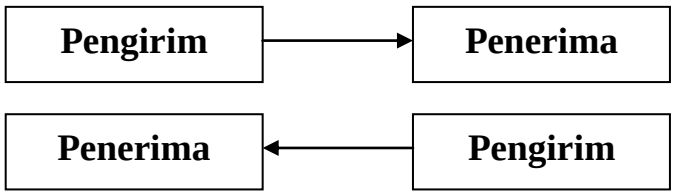
   Simplex merupakan komunikasi yang mengirimkan sinyal transmisinya hanya satu arah



Gambar 2.10 Komunikasi satu arah

2. Half Duplex

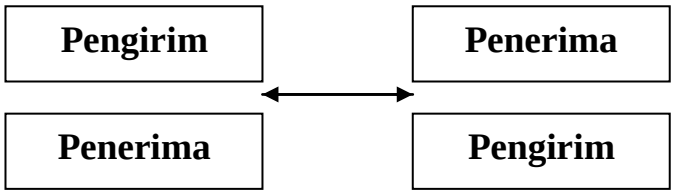
   Half Duplex merupakan komunikasi yang dapat mengirimkan data/ informasi dari dua arah namun dilakukan secara bergantian.



Gambar 2.11 Komunikasi dua arah secara bergantian

3. Full Duplex

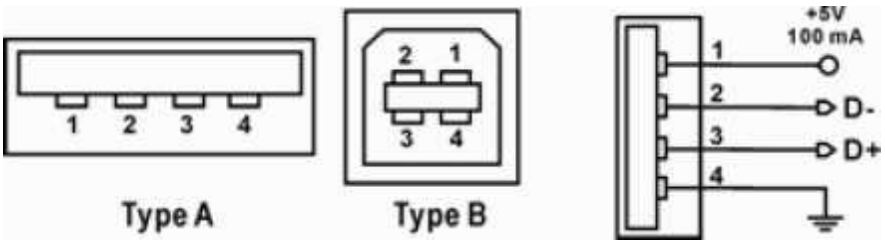
Full Duplex merupakan komunikasi yang dapat mengirimkan data/informasi dari dua arah secara bersamaan.



Gambar 2.12 Komunikasi dua arah secara bersamaan

2.3.2 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.[1]



Gambar 2.13 Konektor USB tipe A dan tipe B [1]

Standarisasi pin USB tipe A dan tipe B adalah 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 adalah 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.*

2.3.3 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*.[4]

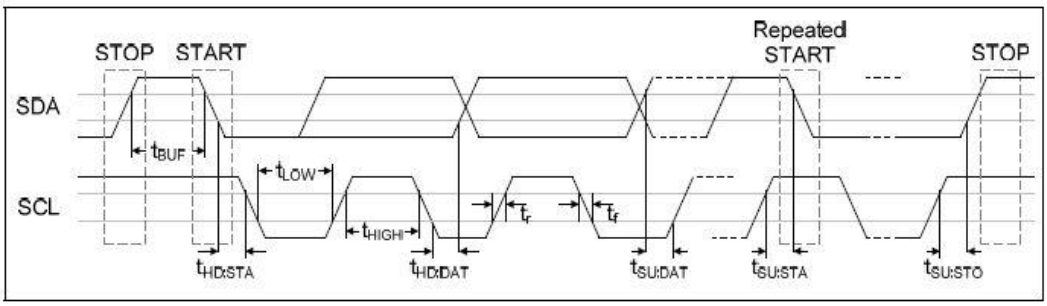


Figure 5-18 I²C Bus Timing

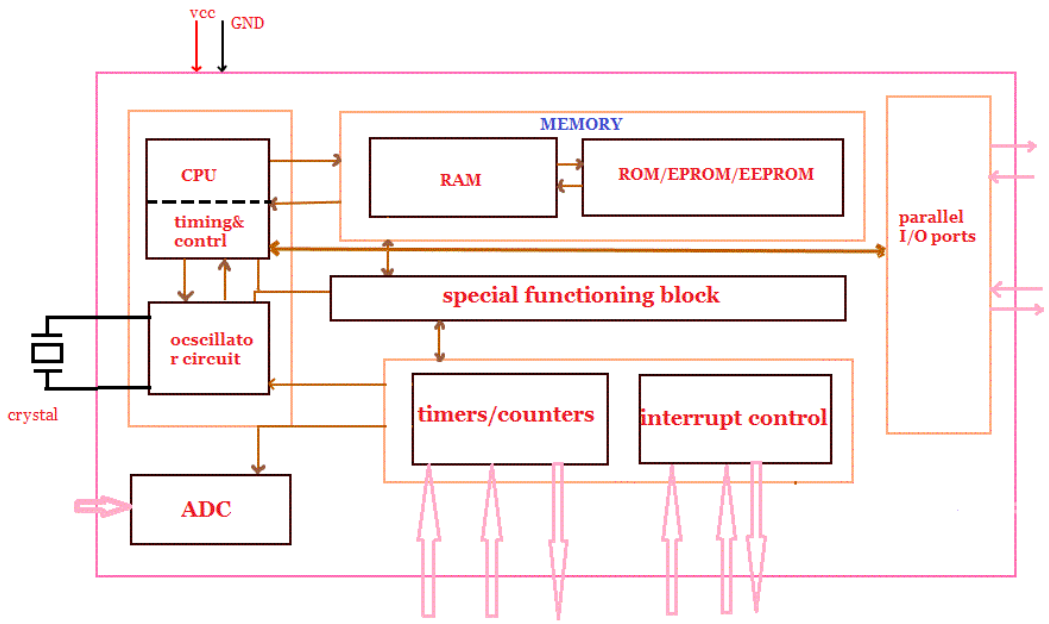
Gambar 2.14 Prinsip kerja protokol bus I2C [4]

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.[4]

2.4 *Microcontroller*

Microcontroller adalah sebuah komputer kecil yang terdapat di dalam sebuah rangkaian terintegrasi. *Microcontroller* 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 2.15 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.

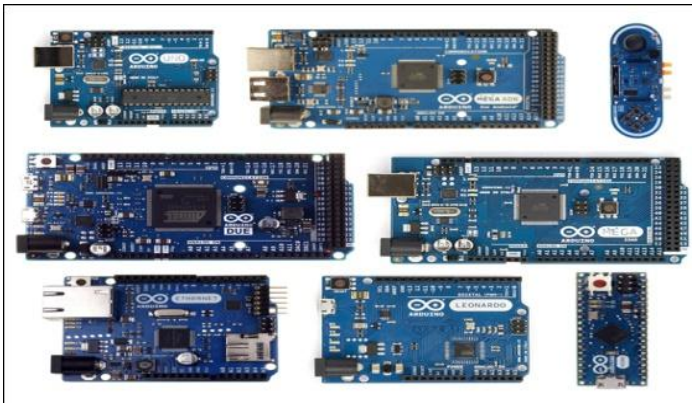
g. *Interrupt Control*

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 2.16 di bawah ini



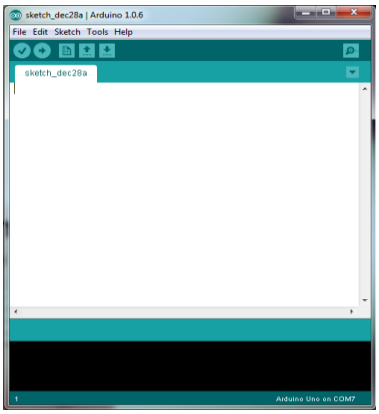
Gambar 2.16 Bentuk Fisik *Arduino [6]*

	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 2.17 Tabel Spesifikasi *Arduino [7]*

2.4.1 *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 2.18 *Arduino* IDE

2.4.2 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.[6]*

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 menampung 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 bilangan bulat yang berkisar antara 0 sampai 255.	1 byte
Int	Tipe data ini berguna untuk menampung bilangan bulat yang berkisar antara -32768 sampai 32767.	2 byte
Unsigned Int	Tipe data ini berguna untuk menampung bilangan 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 bilangan bulat yang berkisar antara -2,147,483, 648 sampai 2,147,483,647	4 byte
Unsigned Long	Tipe data ini berguna	4 byte

	untuk menampung bilang-an bulat yang berkisar antara 0 sampai 4,294,967,295 (2^{23-1})	
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 DEFAULT, INTERNAL, INTERNAL1V1, INTERNAL2V56 dan EXTERNAL	analogReference(Type)
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.*

2.5 Perangkat Keras

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

2.5.1 Aktuator (penggerak)

Aktuator merupakan perangkat mekanis yang berfungsi sebagai alat untuk menggerakkan suatu sistem. Aktuator akan berfungsi ketika diberi masukan berupa sumber tegangan. Terdapat 2 jenis aktuator berdasarkan sumber masukannya yaitu aktuator AC dan Aktuator DC. Aktuator AC yaitu aktuator yang bekerja apabila diberi sumber masukannya berupa tegangan *Alternating Current* (AC), sedangkan aktuator DC yaitu aktuator yang bekerja apabila diberi sumber masukannya berupa tegangan *Direct Current* (DC). Aktuator yang digunakan pada *robot kapal* sebagai penghasil gerakan yaitu *brushless motor* yang diberi masukan tegangan DC berupa sinyal pulsa yang dikirim melalui mikrokontroler.

2.5.2 Kamera *Webcam*



Gambar 2. 19 Kamera *Webcam* C310 [10]

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. 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.

2.5.3 *Licuid Crystal Diode* (LCD) 2x16



Gambar 2. 20 LCD 2X16

LCD 2x16 merupakan sebuah komponen elektronika yang dapat menampilkan suatu karakter tertentu dengan tampilan seperti *dot matrix,*dengan

ukuran *dot matrix* 16 baris dan 2 kolom. LCD ini memiliki sumber daya yang kecil dengan dilengkapi panel LCD serta memiliki kontroler LCD *Complementary Metal Oxide Semiconductor* (CMOS) yang telah terpasang dalam modul tersebut. LCD 2x16 ini memiliki sebuah memori yang dapat diakses yaitu *Random Access Memory* (RAM) dan *display* data RAM, memori tersebut berfungsi sebagai mempermudah pengguna untuk membuat suatu instruksi-instruksi yang akan ditampilkan dengan dihubungkan ke mikrokontroler.[5]

2.5.4 *ESC (Electronic Speed Control)*



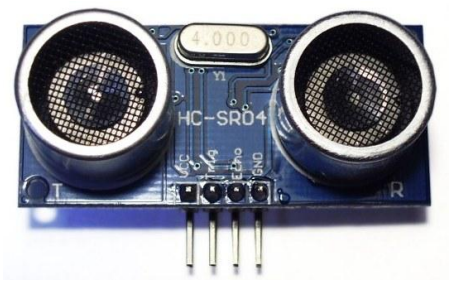
Gambar 2. 21 *ESC (Electronic Speed Control)*

Electronic Speed Control (ESC) adalah rangkaian elektronik yang berfungsi sebagai pengatur kecepatan putaran motor pada pesawat, helicopter dan kapal dengan cara menterjemahkan sinyal yang diterima *receiver* dari *transmitter*. Seperti halnya motor, *ESC* juga dibedakan dua jenis, brushed *ESC* untuk brushed motor dan brushless *ESC* untuk brushless motor. Di pasaran terdapat berbagai merk *ESC* dengan kekuatan arus (*current rating*) dan kekuatan voltase *(voltage rating)* serta *feature* yang ditawarkan. [2]

Untuk menentukan *ESC* yang akan digunakan sangatlah penting untuk mengetahui kekuatan *(peak current)* dari motor. Pilihlah *ESC* yang kekuatannya

melebihi kekuatan motor. Misalnya, dari data kita dapatkan kekuatan motor adalah 12A (ampere) pada saat *throttle* terbuka penuh. sebaiknya *ESC* yang akan kita gunakan adalah *ESC* yang berkekuatan 18A atau 20A. Jika kita paksakan menggunakan *ESC* 10A kemungkinan pada saat *throttle* dibuka penuh, *ESC* akan panas bahkan terbakar. Pada perancangan robot kapal ini menggunakan menggunakan *ESC* 30A.

2.5.5 Ultrasonik



Gambar 2. 22 Ultrasonik HC-SR04

Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek tertentu di depannya, frekuensi kerjanya pada daerah diatas gelombang suara dari 40 KHz hingga 400 KHz.Sensor ultrasonik terdiri dari dari dua unit, yaitu unit pemancar dan unit penerima. Struktur unit pemancar dan penerima sangatlah sederhana, sebuah kristal piezoelectric dihubungkan dengan mekanik jangkar dan hanya dihubungkan dengan diafragma penggetar.

Besar amplitudo sinyal elekrik yang dihasilkan unit sensor penerima tergantung dari jauh dekatnya objek yang dideteksi serta kualitas dari sensor pemancar dan sensor penerima. Proses sensing yang dilakukan pada sensor ini menggunakan metode pantulan untuk menghitung jarak antara sensor dengan

obyek sasaran. Jarak antara sensor tersebut dihitung dengan cara mengalikan setengah waktu yang digunakan oleh sinyal ultrasonik dalam perjalanannya dari rangkaian *Tx* sampai diterima oleh rangkaian *Rx*, dengan kecepatan rambat dari sinyal ultrasonik tersebut pada media rambat yang digunakannya. [2]

2.5.6 *Brushless*



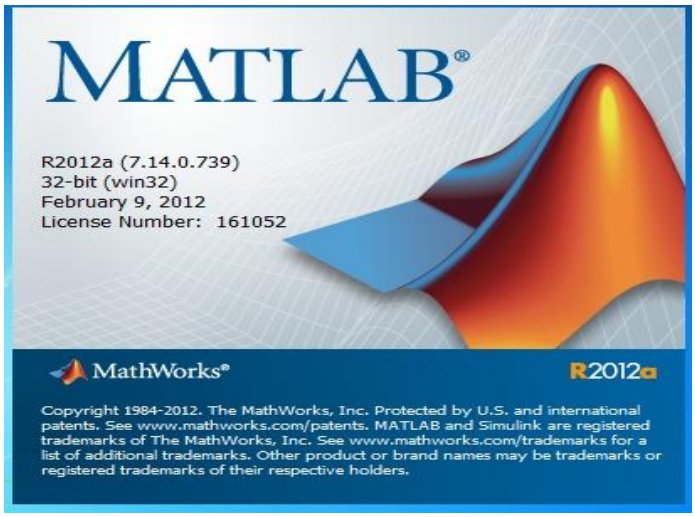
Gambar 2. 23 Motor *Brushless*

Brushless motor merupakan motor yang mempunyai permanen magnet pada bagian "rotor" sedangkan elektron-magnet pada bagian "*stator*"-nya. Secara umum, kecepatan putaran *brushless* motor yang keluar dari *ESC* diatur oleh pulsa dari mikrokontroler, sehingga berbeda dengan *brushed*. Berikut adalah gambaran fisik dari motor brushless.

2.6 Perangkat Lunak MATLAB R2012a (*Software*)

Dalam proses pendeteksian target pada tugas akhir ini menggunakan sebuah *Graphical User Interface* (GUI). 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. GUI pada tugas akhir ini dibuat dengan MATLAB R2012a seperti yang terlihat pada gambar di bawah ini.



Gambar 2. 24 MATLAB R2012a