BAB II

LANDASAN TEORI

Bab ini berisi tentang teori mengenai permasalahan secara garis besar yang dibahas dalam tugas akhir ini dimulai dari definisi Visi komputer, *OpenCV,* model warna, bahasa *Python,* Bahasa *C,* Raspberry Pi dan mikrokontroler Arduino.

2.1 Visi Komputer (*Computer Vision*)

Computer vision merupakan proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akuisisi citra, pengolahan citra, pengenalan dan membuat keputusan. *Computer vision* mencoba meniru cara kerja sistem visual manusia (*human vision*) yang sesungguhnya sangat kompleks. Untuk itu, *computer vision* diharapkan memiliki kemampuan tingkat tinggi sebagaiman *human visual*. Kemampuan itu di antaranya adalah:

- *Object detection* → Mendeteksi objek
- *Recognition* → Menempatkan label pada objek.
- *Description* → Menugaskan properti kepada objek.
- 3D *Inference* → Menafsirkan adegan 3D dari 2D yang dilihat.
- *Interpreting motion* → Menafsirkan gerakan.

Computer vision terdiri dari teknik-teknik untuk mengestimasi ciri-ciri objek dalam citra, pengukuran ciri yang berkaitan dengan geometri objek, dan menginterpretasi informasi geometri tersebut. Secara ringkas bisa dilihat persamaan berikut:

Vision = Geometry + Measurement + Interpretation

Proses-proses dalam *computer vision* dibagi dalam 3 (tiga) aktifitas :

- Memperoleh atau mengakuisisi citra digital.
- Melakukan teknik komputasi untuk memproses atau memodifikasi data citra. (operasi-operasi pengolahan citra).
- Menganalisis dan menginterpretasi citra menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya memandu robot, mengontrol peralatan, memantau manufaktur dan lain-lain.

Computer Vision sering didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali objek yang diamati / diobservasi. Cabang ilmu ini adalah kecerdasan buatan (*Artificial Intelligence*) yang akan mampu menghasilkan sistem intelijen visual (*Visual Intelligence System*). *Computer Vision* merupakan kombinasi antara Pengolahan Citra dan Pengenalan Pola yang hubungan antara ketiganya dapat dilihat pada **Gambar 2.1.** Pengolahan citra merupakan proses awal dari *computer vision*, sedangkan pengenalan pola merupakan proses menginterpretasikan citra. *Computer Vision* terdiri dari 2 bidang ilmu yaitu.

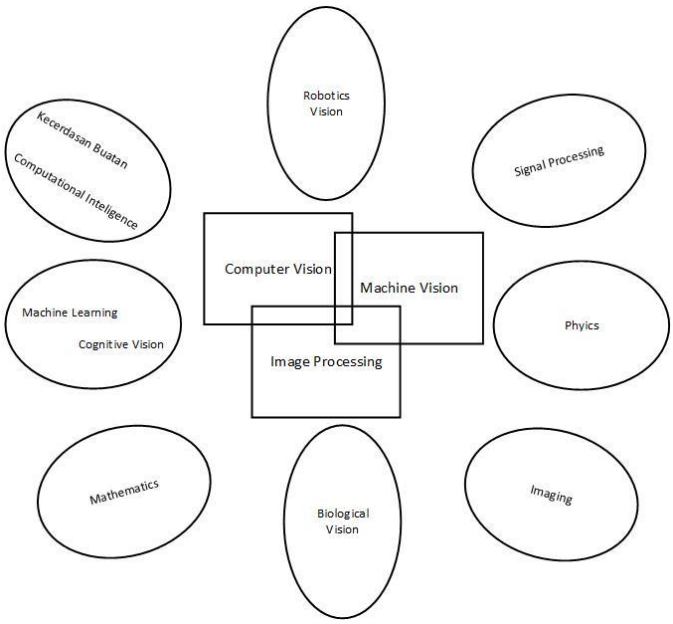
1. Pengolahan Citra (*Image Proessing)*

   Pengolahan citra adalah suatu metode

   yang digunakan untuk mengolah gambar sehingga menghasilkan gambar yang sesuai dengan yang keinginkan kita. Pengambilan gambar bisa dilakukan dengan menggunakan kamera atau alat lain yang bisa digunakan untuk mentransfer gambar misalnya scanner atau kamera digital.

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]

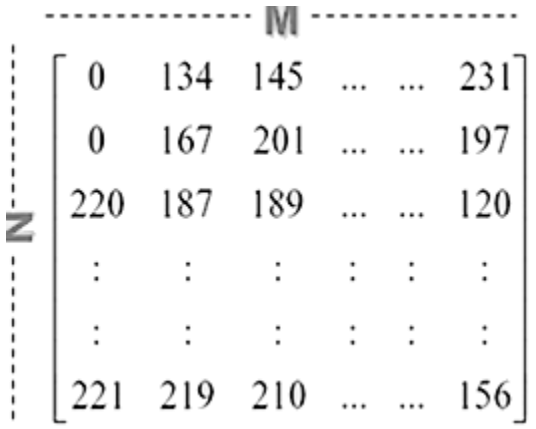


Gambar 2.1 Hubungan *Computer Vision*

2.1.1 Citra Digital

Citra digital adalah kumpulan data diskrit yang berbentuk matriks dua dimensi dimana angka dari matriks tersebut menunjukkan tingkat kecerahan dari titik-titik tersebut. Titik-titik yang membangun citra digital inilah yang sering disebut dengan pixel, tingkat kecerahan dari pixel tersebut berkisar antara 0-255 dimana untuk pixel yang bernilai 0 maka pixel tersebut akan dikatakan gelap atau hitam, sedangkan untuk pixel dengan nilai 255 dapat dapat dikatakan terang.

Suatu citra berukuran N x M pixel yang memiliki intensitas beragam pada tiap pixelnya, direpresentasikan secara numerik dalam bentuk matriks yang berukuran N baris dan M kolom.



Gambar 2.2 Representasi citra secara numerik

2.1.2 Citra Biner

Citra biner adalah citra digital yang hanya memiliki 2 nilai piksel yaitu 1 (putih) dan 0 (hitam), Mode warna ini banyak dipakai dalam proses-proses pengenalan citra yang tidak membutuhkan informasi warna, seperti proses pengenalan *mophology* dari suatu objek, penipisan objek, manipulasi bentuk objek, dan lain-lain.

Terdapat beberapa alasan perlunya dilakukan perubahan dari citra RGB menjadi citra biner, yaitu :

1. Untuk mengenali keberadaan objek, yang divusualisasikan sebagai daerah (*region*) di dalam citra. Contohnya kita akan memisahkan (segmentasi) objek dari gambar latar belakangnya. Piksel-piksel objek dinyatakan dengan nilai 1 (putih) sedangkan piksel lain dengan 0 (hitam).
2. Untuk lebih memfokuskan pada analisis bentuk *morfologi*, yang dalam hal ini intensitas piksel tidak terlalu penting dibandingkan bentuknya.
3. untuk mengkonversikan citra yang telah ditingkatkan kualitas tepinya ke penggambaran garis-garis tepi.

Sebelum mendapatkan citra biner dari warna piksel, harus dilakukan teknik *thresholding*, maka nilai biner dari suatu piksel akan diperoleh dengan melakukan klasifikasi nilai piksel berdasarkan nilai *thresholding* tersebut, seperti yang terlihat pada pernyataan di bawah ini :

$$f(m,n) = \begin{cases} 1, jika\ fb(m,n) \geq th \\ 0, jika\ \ fb(m,n) < th \end{cases}$$

2.1.3 Threshold

Threshold adalah salah satu metode proses pemisahan citra dari nilai derajat keabuan menjadi nilai biner (hitam putih), sehingga dapat membantu memudahkan proses pendeteksian. Dengan menggunakan metode *threshold* maka derajat keabuan suatu citra dapat diubah sesuai dengan keinginan, metode *threshold* ini sangat banyak digunakan untuk melakukan pendeteksian suatu objek terutama pendeteksian objek berdasarkan warna dari objek tersebut.

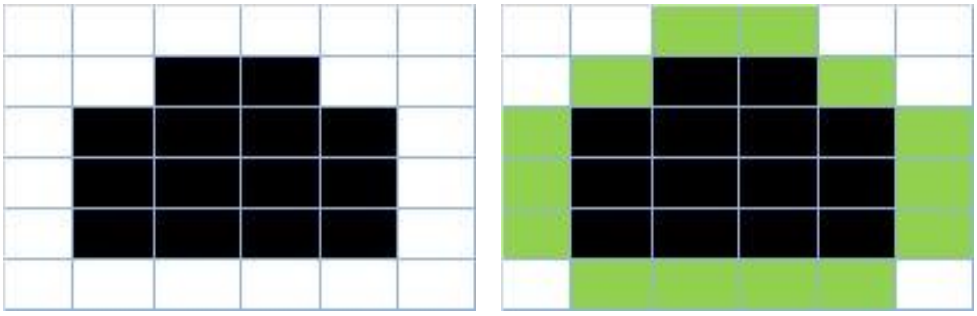
2.1.4 Dilasi

Dilasi adalah operasi morfologi yang akan menambahkan piksel pada batas antar objek dalam suatu citra digital. Atau secara rinci dilasi merupakan suatu proses menambahkan piksel pada batasan dari objek dalam suatu citra sehingga nantinya apabila dilakukan operasi ini maka citra hasilnya lebih besar ukurannya dibandingkan dengan citra aslinya.

Jika A dan B adalah himpunan, dilasi antara A dan B dinyatakan A ⊕ B dan didefinisikan dengan:

$$A \oplus B = \bigcup_{b \in B} A_B \qquad \text{..... (2.1)}$$

dimana himpunan B adalah *strel*, sedangkan A himpunan (objek citra) yang terdilasi.



a b

Gambar 2.3 Proses Dilasi

a. Citra Asli

b. Citra Hasil Dilasi

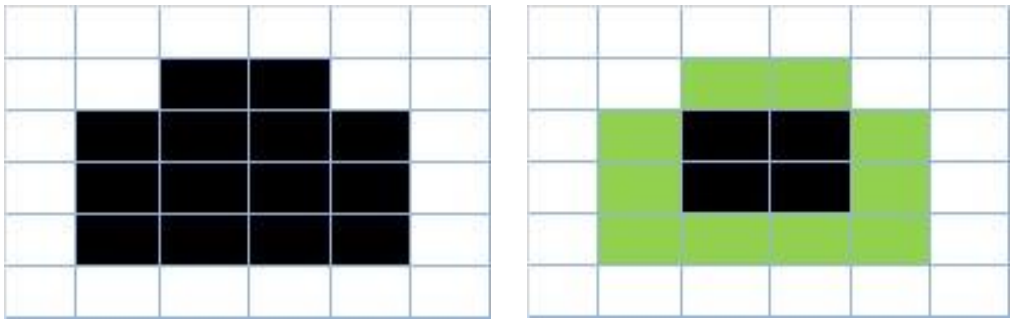
2.1.5 Erosi

Erosi merupakan kebalikkan dari dilasi. Proses ini akan membuat ukuran sebuah citra menjadi lebih kecil. Berbeda dengan dilasi, apabila erosi dilakukan maka yang dikerjakan adalah memindahkan piksel pada batasan-batasan objek yang akan di erosi. Jumlah dari piksel yang ditambah atau dihilangkan bergantung pada ukuran dan bentuk dari *structuring element* yang digunakan untuk memproses citra tersebut.

Jika A dan B adalah himpunan piksel, erosi A oleh B dinyatakan dengan A Θ B, didefinisikan sebagai:

$$A \ominus B = \bigcap_{b \in B} A_{-b} \qquad \text{..... (2.2)}$$

Persamaan di atas menunjukkan bahwa erosi A oleh B adalah kumpulan semua titik di mana B ditranslasikan di dalam isi A



a b

Gambar 2.4 Proses Erosi

a. Citra Asli

b. Citra Hasil Erosi

2.1.6 Pixel

Pixel adalah komponen pembentuk gambar atau representasi sebuah titik terkecil dalam sebuah gambar yang dihitung per inci, semakin banyak jumlah pixel semakin baik pula gambar yang diperoleh, Pada umumnya citra digital yang banyak digunakan adalah citra digital dengan tingkat kuantitas 8 bit, hal ini didasarkan atas pertimbangan besar ukuran berkas dari citra dan kemampuan mata manusia dalam membedakan warna.

2.1.7 Segmentasi dan Pengembangan

Proses segmentasi mempunyai tujuan yang hampir serupa dengan proses klasifikasi tidak terpandu. Istilah segmentasi citra itu sendiri mempunyai arti membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan yang tertentu antara tingkat keabuan suatu piksel dengan tingkat

keabuan piksel-piksel tetangganya. Citra yang diperoleh kemudian akan terdiri atas bagian-bagian objek dan objek dan bagian latar belakang.

Memilih bentuk-bentuk dalam sebuah citra sangat berguna dalam pengukuran atau pemahaman citra. Secara tradisional, pemgambangan didefinisikan sebagai proses pendefinisian jangkauan nilai- nilai gelap-terang pada citra yang sebenarnya, memilih piksel-piksel dalam jangkauan ini sebagai latar depan dan sisanya sebagai latar belakang. Dengan demikian, citra terbagi atas dua bagian, yaitu bagian hitam dan bagian putih, atau warna-warna yang membatasi setiap wilayah.

2.1.8 OpenCV

Open Source Computer Vision (OpenCV) adalah suatu *library computer vision* yang dibuat oleh para *developer Intel Corporation* yang bersifat *Open Source, library* tersebut merupakan kumpulan fungsi-fungsi C dan beberapa kelas C++ yang mengimplementasikan banyak algoritma-algoritma Image Processing dan Computer Vision. Salah satu penerapan program OpenCV yaitu dalam dunia robotika.

Library OpenCV mempunyai lebih dari 2500 algoritma yang telah dioptimalkan dimana meliputi sebuah himpunan menyeluruh dari keduanya yaitu klasik dan seni beberapa algoritma *computer vision* dan *machine learning.* Algoritma-algoritma tersebut dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi obyek, mengklasifikasi tindakan manusia dalam video, mengikuti jejak perpindahan obyek, mengekstrak model-model 3D obyek, menghasilkan titik awan 3D dari kamera stereo, dan lain sebagainya.

OpenCV dapat diterapkan pada pemrograman C++, C. Python, Java dan MATLAB. OpenCV mendukung untuk sistem operasi Windows, Linux, Android dan Mac OS. [2]

2.1.9 *Hough Circle*

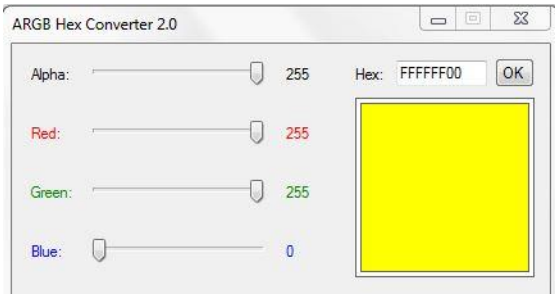
Sebuah lingkaran memiliki matematis sebagai (x-x_{center})^2 + (y - y_{center})^2 = r^2 where (x_{center},y_{center}) sebagai pusat lingkaran. Dari persamaan tersebut memiliki tiga parameter yang membutuhkan akumulator 3D untuk menentukan kontur lingkaran yang tidak efektif untuk menentukan lingkaran di *OpenCV*. *Hough Circle* merupakan metode yang terdapat pada pustaka *OpenCV*, berfungsi untuk mendeteksi kontur lingkaran dengan menentukan parameter dan radius lingkaran dengan cara yang lebih efektif.

2.1.10 *Color Filtering*

Color Filtering adalah metode yang berguna untuk menemukan sebuah warna yang terdapat pada sebuah gambar. Dengan ditemukannya warna yang kita cari pada sebuah gambar, kita dapat menentukan proses apa yang selanjutnya harus dilakukan. Pada dasarnya pencarian ini menggunakan kombinasi dari komponen *Red, Green,* dan *Blue* yang terdapat pada gambar. Nilai dari masing-masing komponen ini didapat dari hasil beberapa kali eksperimen. Hasil dari eksperimen ini adalah sebuah kombinasi dari komponen warna *Red, Green,* dan *Blue*. Kombinasi warna ini yang kemudian dijadikan filter yang merupakan penentu sebuah warna diloloskan atau tidak. Keluaran dari metode ini langsung akan menghasilkan sebuah gambar biner. Gambar biner sendiri adalah sebuah gambar yang hanya memiliki 2 derajat keabuan yaitu hitam dan putih.

2.1.11 Model Warna *Red Green Blue* (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.



Gambar 2.5 Ilustrasi pencampuran RGB

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. [4]

2.1.12 Model Warna *Hue Saturation Value* (HSV)

Model warna HSV merupakan kepanjangan dari *Hue Saturation* dan *Value*. Dari pengertian tersebut memiliki fungsi masing-masing yang berbeda. Model warna HSV ini dipandang lebih dekat dari model warna RGB dalam mendeskripsikan sensasi warna oleh mata manusia. *Hue* merupakan suatu ukuran panjang gelombang dari warna utama, *hue* mempunyai ukuran berkisar antara 0-255. 0 mewakili warna merah hingga melalui suatu spektrum kembali bernilai 256

atau kembali menjadi warna merah kembali. *Saturation* merupakan suatu proses untuk meningkatkan kecerahan warna yang di dasari dari jumlah *hue* murni pada warna akhir. Jika *saturation* bernilai nol maka warna akhir adalah bukan *hue* yang terbentuk hanya cahaya putih saja. Jika *saturation* bernilai 255 maka tidak ada pencahayaan tambahan pada warna akhir. *Value* merupakan sebuah ukuran seberapa besar kecerahan dari suatu warna. Apabila warna itu memiliki ukuran 100% maka akan terlihat sangat cerah, dan ketika 0% maka akan terlihat gelap.

Seperti gambar di atas dapat dijelaskan, untuk nilai *hue* dapat dilihat dari ukuran sudut yang melingkar hingga 360º, dari setiap perbedaan sudut hasil warnanya akan berbeda. *Saturation* dapat dilihat dari ukuran horizontal ketika nilai warna itu di perkecil yang terlihat hanya warna putih. *Value* dapat dilihat dari ukuran vertikal nilai *value* akan berbanding dari nilai *saturation* yaitu ketika di perkecil akan tampak gelap. [5]

Color	Hue	Saturation	Value
Black	0° < H < 360°	0 < S < 1	V < 0.1
White	0° < H < 360°	S < 0.15	V > 0.65
Gray	0° < H < 360°	S < 0.15	0.1 < V < 0.65
Red	H < 11°, H > 351°	S > 0.7	V > 0.1
Pink	H < 11°, H > 351°	S < 0.7	V > 0.1
	310° < H < 351°	S > 0.15	V > 0.1
Orange	11° < H < 45°	S > 0.15	V > 0.75
Brown	11° < H < 45°	S > 0.15	0.1 < V < 0.75
Yellow	45° < H < 64°	S > 0.15	V > 0.1
Green	64° < H < 150°	S > 0.15	V > 0.1
Blue-green	150° < H < 180°	S > 0.15	V > 0.1
Blue	180° < H < 255°	S > 0.15	V > 0.1
Purple	255° < H < 310°	S > 0.5	V > 0.1
Light Purple	255° < H < 310°	0.15 < S< 0.5	V > 0.1

Gambar 2.6 Model Warna HSV [6]

2.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam tugas akhir ini adalah program bahasa Python yang digunakan untuk proses pada Raspberry Pi dan program bahasa C untuk proses pada mikrokontroler Arduino. Berikut adalah uraian secara gari besar dari masing-masing perangkat lunak tersebut.

2.2.1 Bahasa Python

Bahasa pemrograman ini dibuat oleh Guido van Rossum dari Amsterdam, Belanda. Pada awalnya, motivasi pembuatan bahasa pemrograman ini adalah untuk bahasa skrip tingkat tinggi pada sistem operasi terdistribusi Amoeba. Bahasa pemrograman ini menjadi umum digunakan untuk kalangan *engineer* seluruh dunia dalam pembuatan perangkat lunaknya, bahkan beberapa perusahaan menggunakan python sebagai pembuat perangkat lunak komersial.

Python merupakan bahasa pemrograman yang *freeware* atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan *source* codenya, *debugger* dan *profiler*, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, GUI (antarmuka pengguna grafis), dan basis datanya. Python dapat digunakan dalam beberapa sistem operasi, seperti kebanyakan sistem UNIX, PCs (DOS, Windows, OS/2), Macintosh, dan lainnya. Pada kebanyakan sistem operasi linux, bahasa pemrograman ini menjadi standarisasi untuk disertakan dalam paket distribusinya.

1. Tipe Data di Python

   Variabel menyimpan data yang dilakukan selama program dieksekusi dan isinya dapat diubah oleh operasi – operasi tertentu pada program yang

menggunakan variabel tersebut. Di dalam Python, terdapat beberapa tipe data yang cukup unik bila dibandingkan dengan bahasa pemrograman seperti C, Java, dan yang lainnya. Tipe data pada Python adalah sebagai berikut :

- Boolean, contoh *True and False*
- *Complex*, pasangan angka *real* dan imajiner, misalnya 1 + 5j
- *Date*, bilangan yang dapat dikonversi menjadi format tanggal, misalnya 26-09-2013
- *Float*, bilangan yang mempunyai koma, misalnya 3.14, 6.387
- *Hexadecimal*, bilangan dalam format heksa, misalnya 7b, 4d2
- *Integer*, bilangan bulat, misalnya 10, 20, 30, 15, 37
- *Long*, bilangan bulat yang panjang, misal 123456789123456789L
- *None*, data yang tidak terdefinisi tipe data apapun
- *String*, data yang berisi kalimat. Bisa dibentuk dengan diapit tanda ' dan ', atau diapit “ dan“, atau diapit “”” dan “”” untuk membentuk paragraf.
- *List*, sebuah data berupa untaian yang menyimpan berbagai tipe data dan isinya bisa diubah.
- *Tuple*, sebuah data berupa untaian yang menyimpan berbagai tipe data tapi isinya tidak bisa diubah.
- *Dictionary*, sebuah data berupa untaian yang menyimpan berbagai tipe data berupa pasangan penunjuk dan nilai.

Object, sebuah data yang berisi atribut dan metode.

2. Operator – Operator di Python

Selain variabel diatas, terdapat juga beberapa operator untuk pemrosesan data di Python. Agar lebih memahami seperti apa cara kerja operator di python, bukalah *console* **python** dan coba contoh disamping penjelasan tiap operator. Berikut operator yang ada di Python :

A. Aritmatika (*Aritchmetic Operator*)
B. Perbandingan (*Comparison Operator*)
C. Penugasan (*Assignment Operator*)
D. Biner (*Bitwiser Operator*)
E. Logika (*logical Operator*)
F. Keanggotaan (*Membership Operator*)
G. Identitas (*Identity Operator*)

Beberapa teknologi yang menggunakan Python yaitu *Django, PyGame, Panda 3D, SimpleCV, NLTK,* dan *Flask.* [7]

2.2.2 Bahasa C

Pemrograman menggunakan Bahasa C adalah salah satu dari sekian banyak bahasa dasar untuk pemrograman mikrokontroler, misalnya bahasa *Assembly*, *Basic*, dan lain-lain. Bahasa pemrograman Bahasa C dikenal di seluruh dunia sebagai bahasa pemrograman handal, cepat, mudah, dan tergolong ke dalam bahasa pemrograman tingkat menengah. Bahasa C banyak digunakan untuk aplikasi mikrokontroler karena kemudahan dan kompatibel terhadap mikrokontroler jenis AVR. Setiap bahasa pemrograman mempunyai standar penulisan program. Konstruksi dari program bahasa C harus mengikuti aturan sebagai berikut.

1. *Preprocessor*

   Biasanya digunakan untuk menyertakan file header (.h) atau file library. File include berguna untuk memberitahu compiler agar membaca file yang di include- kan lebih dahulu agar mengenali definisi-definisi yang digunakan dalam program sehingga tidak dianggap error.

   Cara penulisan:

   #include <…….> untuk lokasi standar file yang telah disetting oleh tools biasanya pada folder include atau forder directori complier.

   #include"……." Untuk lokasi file yang yang kita tentukan sendiri.

   File header io.h adalah file yang segala informasi/definisi tentang register-register fungsi khusus (SFR) dan bit-bit atau pin-pin mikrokontroler AVR.

2. Komentar

   Komentar merupakan bagian kode program yang tidak dieksekusi saat program dijalankan. Komentar diperlukan untuk memperjelas jalannya program agar lebih mudah dimengerti. Komentar dapat dimanfaatkan untuk penuangan ide yang belum sempat diimpelementasikan.

   Contoh penulisan :

   // hanya satu ini sebelu enter yang dianggap komentar.

   * kata-kata yang berbeda dalam tanda garis miring dengan bintang dianggap komentar berapa pun panjangnya dan berapa pun barisnya*/.

   Komentar adalah bagian yang tidak dieksekusi oleh kompiler. Ini mempertegas bahwa kode program dibuat untuk dibaca dan dimengerti oleh manusia. Biasakan untuk menuliskan komentar dibagian tertentu dari program.

3. Nama

Nama digunakan untuk mengidentifikasi variabel, konstanta, tipe, fungsi dan prosedur. Penamaan memiliki aturan.

- Dalam C, nama membedakan antara huruf besar dan huruf kecil.
- Nama harus dimulai dengan alfabet, tidak boleh angka, spasi atau karakter khusus.
- Karakter penyusun nama hanya huruf alfabet, angka dan underscore.
- Tidak boleh mengandung operator aritmatik, relasional, tanda baca dan karakter khusus.

Setiap nama harus ditulis secara utuh, tidak boleh diberi spasi.

Contohnya penulisan : int x, y, jumlah; // x, y dan jumlah adalah nama dari variabel intnilai(const int angka, int test){} // nilai adalah nama dari fungsi.

4. Operator

Operator adalah karakter-karakter khusus untuk memanipulasi variable. Operand adalah variabel atau konstanta yang merupakan bagian dari pernyataan.

Aritmatika :

- + adalah penjumlahan.
- - adalah pengurangan.
- * adalahperkalian.
- / adalah pembagian.
- ++ adalah increment.

- -- adalah decrement.

Logika :

- == adalah logika sama dengan.
- != adalah logika tidak sama dengan.
- < adalah logika lebih kecil.
- <= adalah logika lebih kecil sama dengan.
- > adalah logika lebih besar.
- > adalah logika lebih besar sama dengan.
- ! adalah logika not.
- && adalah logika AND.
- || adalah logika OR

Manipulasi Bit :

- ~ adalah mengkomplementkan.
- & adalah mengANDkan.
- | adalah mengORkan.
- ^ adalah mengXORkan.
- << adalah shift left.
- >> adalah shift right.

5. Kontrol Aliran Program

Untuk mencerdaskan sebuah program, maka diperlukan algoritma dimana terdiri dari berbagai pengendalian aliran program.

- If (...){…}

Digunakan untuk mengecek satu kondisi satu blok jawaban.

- If (…) {…} else {…}

  Digunakan untuk mengecek satu kondisi dua blok jawaban.

- If (…) {…} else if (…) {…} else {…}

  Digunakan untuk mengecek beberapa kondisi yang berkaitan.

- While (…) {…}

  Digunakan untuk perulangan/looping/iterasi jika kondisi yang diuji bernilai benar.

- Do {…} while (…)

  Digunakan untuk perulangan/looping/iterasi jika kondisi yang diuji bernilai benar.Perbedaanya dengan while tanpa do adalah blok di eksekusi terlebih dahulu baru diuji, hal ini dapat terjadi kemungkinan yang diuji salah namun blok tetap di eksekusi.

- For (… ; … ; … ;…) {…}

  Digunakan untuk perulangan/looping/iterasi jika kondisi dan syarat yang ditentukan. [8]

2.3 Perangkat Keras

Berikut akan dijelaskan secara garis besar mengenai landasan teori perangkat keras dan komponen yang digunakan pada perangkat yang di rancang.

2.3.1 Raspberry Pi

Raspberry Pi, sering juga disingkat dengan nama Raspi, adalah komputer papan tunggal (*Single Board Computer*) yang memiliki ukuran sebesar kartu kredit. Raspberry Pi bisa digunakan untuk berbagai keperluan, seperti spreadsheet, game,

bahkan bisa digunakan sebagai media *player* karena kemampuannya dalam memutar video *high definition*. Raspberry Pi terlihat seperti *motherboard*, dengan berbagai *chip* dan *port* yang dipasang sama seperti komputer atau laptop. Di bawah ini akan disebutkan berbagai spesifikasi *hardware* pada Raspberry Pi.

2.3.2 Bagian-Bagian Raspberry Pi

Berikut akan dijelaskan bagian-bagian penting yang terdapat pada papan Raspberry Pi.

1. ARM CPU/GPU

   Raspberry Pi menggunakan Broadcom BCM2836 *System on Chip* yang terdiri dari unit ARM CPU dan VideoCore denga 4 pengolahan grafis GPU.

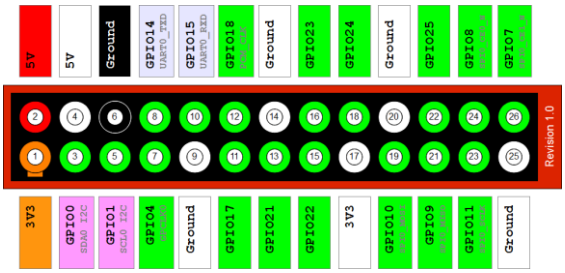


Gambar 2.7 Broadcom BCM2836

2. *General Purpose Input/Output* (GPIO)

   GPIO adalah perangkat yang memungkinkan Raspberry Pi berinteraksi sesungguhnya dengan dunia luar. Berbentuk pin layaknya *header* yang dikenal pada dunia *hardware*, Raspberry Pi memiliki pin GPIO sebanyak 26 buah dan tambahan hingga 40 pin pada Raspberry Pi model 2. Masing-masing pin memiliki fungsi yang berbeda. Namun, pin-pin tersebut bisa diatur, baik sebagai input maupun output. Itulah alasan disebut sebagai *General Purpose*. Ada beberapa pin yang memiliki fungsi khusus, seperti antarmuka serial I2C, SPI, dan UART. Jika menggunakan fungsi pin GPIO

tersebut, maka jumlah pin GPIO (yang bisa digunakan sebagai input dan output) ada sebanyak 8 pin, namun masih dapat mengubah fungsi pin tersebut sehingga menghasilkan 17 pin GPIO. Sisanya digunakan untuk Vcc dan Ground.



Gambar 2.8 pin GPIO Raspberry Pi

3. *Synchronous Dynamic Random Access Memory* (SDRAM)

   Sama seperti memori pada komputer, SDRAM berfungsi untuk mengoptimalkan pemrosesan pada Raspberry Pi

4. USB 2.0 Port

   Raspberry Pi memiliki port USB 2.0 untuk proses transfer data dan komunikasi/input keyboard, mouse dan perangkat lainnya yang menggunakan port USB.



Gambar 2.9 USB Port Raspberry Pi

5. Video Output

   Video output digunakan untuk menampilkan display Raspberry Pi di layar/monitor. Raspberry Pi 2 memiliki Port untuk output video, yaitu Port HDMI.



Gambar 2.10 Port HDMI Raspberry Pi 2

6. Audio input Output

   File media seperti musik, video, dan media lainnya dapat dijalankan di Raspberry Pi. Raspberry Pi memiliki port audio output untuk dihubungkan ke speaker atau headset.Untuk audio intput, Raspberry Pi dapat dihubungkan/dipasangkan dengan sound card atau USB mic.



Gambar 2.11 Port Audio Raspberry Pi 2

7. Penyimpanan

   Raspberry Pi menggunakan memori SD sebagai *storage* penyimpanan sekaligus sebagai memori internal. Memori internal digunakan sebagai memori utama untuk menyimpan sistem operasi Raspberry Pi, selain itu juga bisa digunakan untuk menyimpan data lainnya. Ukuran minimum memori internal adalah 4 GB, karena untuk menyimpan sistem operasi membutuhkan sekitar 1,8 GB.



Gambar 2.12 *Board Storage* Raspberry Pi

8. Jaringan

   Papan jaringan hanya terdapat pada model B saja. Papan jaringan yang digunakan adalah single RJ45 *Ethernet Jack*, dengan kecepatan 10/100 Mbps.



Gambar 2.13 RJ45 Ethernet Raspberry Pi

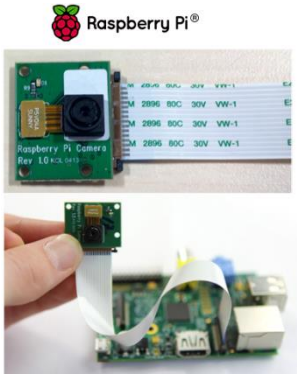
9. Power

   Raspberry Pi menggunakan tegangan sebesar 5V dan arus sekitar 700 – 1000 mA, tergantung periferal apa yang terhubung pada Raspberry Pi, misal keyboard atau mouse. Kebutuhan daya yang dibutuhkan untuk antarmuka yang terdapat pada Raspberry Pi, yaitu pin GPIO membutuhkan daya 50 mA (yang didistribusikan ke semua pin), port HDMI membutuhkan daya 50 mA, modul kamera membutuhkan daya 250 mA, dan untuk mouse dan keyboard membutuhkan sekita 100 mA hingga 1000 mA. [9]

2.3.3 Kamera Raspberry Pi

Kamera yang digunakan adalah model Omnivision 5647, model ini berukuran sama dengan kamera yang biasa digunakan di handphone. Kamera ini menggunakan PCB kecil dan terhubung ke prosesor BC2836 melalui port *Camera Serial Interface-2* (CSI-2) pada Raspberry Pi menggunakan *Ribbon Cable* atau kabel fleksibel yang mampu mengambil gambar atau merekam video, port CSI ini memiliki kecepatan data transfer yang tinggi dan sangat baik mengambil data *pixel*. Kamera terhubung ke *Image System Pipeline* (ISP) pada Raspberry Pi *SoC* di mana data kamera yang masuk diproses dan dikonversi ke gambar atau video ke penyimpanan (*Micro* SD).

Modul kamera ini mampu mengambil foto hingga mencapai resolusi 5 *Megapixels* atau 2592 x 1944 Pixels dan merekam video hingga mencapai resolusi 1080p30 atau 1920 x 1080 pixels x 30 fps. Modul kamera mendukung untuk mengambil gambar mentah atau *RAW Capturing* (data langsung dari sensor) dan format gambar lainnya seperti, JPEG, PNG, GIF dan BMP, YUV yang tidak dikompresi, atau foto RGB yang terkompresi. Hal ini mampu merekam video H.264 dengan format dasar, format utama atau *High-profile formats*. [10]



Gambar 2.14 Modul Kamera Raspberry Pi [11]

2.3.4 Mikrokontroler Arduino Uno

Arduino Uno adalah papan mikrokontroler berbasis ATmega328. Arduino Uno memiliki 14 digital pin input/output, dimana 6 pin digunakan sebagai output PWM, 6 pin input analog, 16 MHz resonator keramik, koneksi USB, jack catu daya eksternal, header ICSP, dan tombol reset. Ini semua berisi hal-hal yang diperlukan untuk mendukung mikrokontroler; sederhana saja, hanya dengan menghubungkannya ke komputer dengan kabel USB atau sumber tegangan dengan adaptor AC-DC dan atau baterai untuk memulai menggunakan papan arduino.

Arduino Uno R3 berbeda dari semua papan Uno sebelumnya yang sudah tidak menggunakan chip driver FTDI USB-to-serial. Sekarang, Arduino Uno menggunakan fitur Atmega16U2 (Atmega8U2 sampai dengan versi R2) yang diprogram sebagai konverter USB-to-serial. [12]



Gambar 2.15 Arduino Uno

2.3.5 Motor Servo

Motor Servo adalah sebuah motor dengan sistem umpan balik tertutup dimana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. [13]



Gambar 2.16 Motor Servo [14]

2.3.6 Motor DC

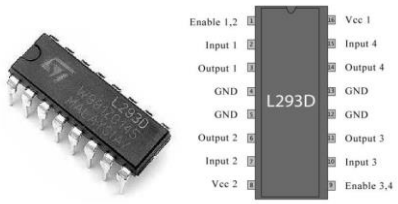
Motor DC adalah salah satu komponen yang dapat difungsikan sebagai penggerak roda-roda pada robot tank. Motor DC memiliki fungsi untuk mengubah energi listrik menjadi energi mekanik. Pada aplikasinya pergerakan pada sebuah mobil listrik umumnya menggunakan motor DC sebagai komponen penggeraknya, karena jenis motor ini lebih mudah untuk dikendalikan. Kecepatan yang dihasilkan oleh motor DC berbanding lurus dengan potensial yang diberikan. [15]



Gambar 2.17 Motor DC

2.3.7 Motor Driver

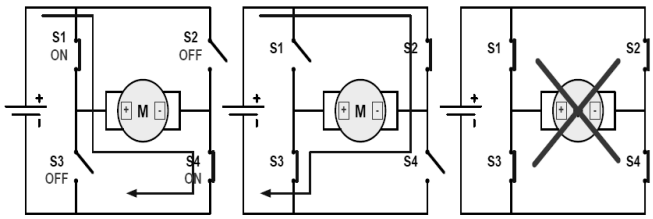
Driver motor merupakan pilihan alternatif yang dapat digunakan karena dengan adanya driver motor ini dapat membantu untuk mengontrol / mengendalikan motor DC pada robot ini. Ada beberapa jenis driver motor DC yang sering digunakan, yaitu menggunakan rangkaian *H-bridge* transistor, *H-bridge* MOSFET dan IC driver motor. Driver motor yang mudah digunakan yaitu menggunakan IC driver motor seperti jenis L293D dan L298N.



Gambar 2.18 Driver Motor L293D [15]

L293D diproduksi oleh SGS-Thomson merupakan sebuah *chip* yang di dalamnya terdapat dua rangkaian *H-Bridge*. Komponen ini bisa mengendalikan dua buah motor DC sampai tegangan 4,5 VDC, arus sebesar 1A untuk tiap kanal dengan arus puncak 1,2A. Untuk satu bagian motor DC dapat dikontrol dengan 3 pin seperti 1 (*enable*) dan 2 masukan. Kemudian pin *output* yang menuju ke motor DC dihubungkan dengan pin keluaran.

Secara konsep rangkaian *H-bridge* ini dapat diilustrasikan sebagai 4 saklar yang tersusun sedemikian rupa sehingga memungkinkan motor dapat teraliri arus dengan arah yang berkebalikan. Cara kerja dari rangkaian *H-bridge* terdapat pada Gambar 2.4 berikut ini:



Gambar 2.19 Cara Kerja *H-Bridge* [15]

Dari gambar di atas, berikut *H-Bridge* bekerja :

1. Ketika S1 dan S4 tertutup (diagonal) dan S2 dan S3 terbuka, maka arus akan mengalir dari baterai menuju ke kutub positif motor kemudian keluar ke kutub negatif motor, maka motor akan berputar ke arah kanan.

2. Ketika S2 dan S3 tertutup (diagonal) dan S1 dan S4 terbuka, maka arus akan mengalir sebaliknya yaitu mengalir dari baterai menuju ke kutub negatif motor kemudian keluar ke kutub positif motor, maka motor akan berputar ke arah kiri. Jika semua saklar tertutup, maka motor akan berhenti, dan jika diteruskan maka akan menyebabkan "*short circuit*".
3. Jika semua saklar tertutup, maka motor akan berhenti, dan jika diteruskan maka akan menyebabkan "*short circuit*".

2.3.8 Bluetooth

Bluetooth adalah sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz yang memungkinkan untuk melakukan komunikasi data dan suara seraca *real time* antara *host to host bluetooth* dengan jangkauan yang terbatas. Dengan menggunakan , kita dapat dengan mudah menghubungkan perangkat yang ingin kita hubungkan. Contoh, penggunaan *bluetooth* dalam bidang percetakan. Misalnya, kita dapat mencetak hasil foto dengan mengambil file dari kamera digital kemudian di pindah ke PC dengan menggunakan *bluetooth* dan kita tinggal mencetaknya menggunakan printer.



Gambar 2.20 Modul Bluetooth HC-05

2.3.9 Servo Driver PCA9685

PCA9685 merupakan I2C-Bus yang mengendalikan 16 Kanal LED dengan warna *Red, Green, Blue, Amber* (RGBA). Setiap output LED memiliki resolusi sendiri 12-bit (4096 *steps*) frekuensi tetap individu PWM yang beroperasi pada frekuensi 24 Hz sampai 1526 Hz dengan *duty cycle* yang disesuaikan dari 0% sampai 100%, ini memungkinkan LED dapat diatur ke nilai kecerahan tertentu. Semua output ditetapkan untuk frekuensi PWM yang sama.

Motor servo bergerak menggunakan output PWM yang terdapat di sebagian besar mikrokontroler *embedded.* Sementara Raspberry Pi hanya memiliki satu PWM yang terdapat pada pin GPIO18, oleh karena itu dibutuhkan *driver* yang dapat mengendalikan motor servo lebih dari satu, yaitu menggunakan PCA9685.



Gambar 2.21 Driver PCA9685

2.4 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 perangkatperangkat lainnya. Pada dasarnya untuk terjadinya komunikasi dibutuhkan minimal dua perangkat alat dan terdapat media transmisinya.



Gambar 2.22 Alur komunikasi sederhana

Keterangan untuk gambar diatas yaitu :

- Pengirim

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

- Media Transmisi

  Media transmisi berfungsi sebagai alat/ media pembawa untuk menyampaikannya informasi yang dituju.

- Penerima

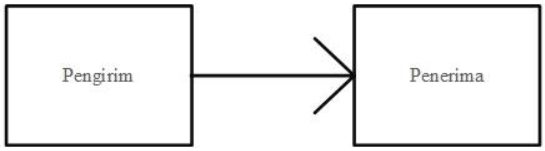
  Penerima berfungsi sebagai alat/ media yang dikirim/ ditujuan. [16]

2.4.1 Aliran Komunikasi Data

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

1. Simplex

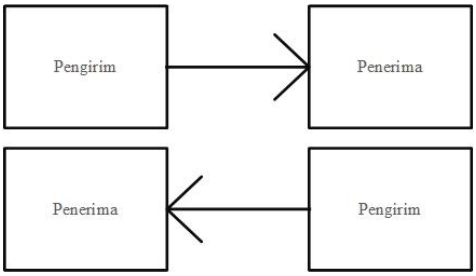
   Simlex merupakan komunikasi yang mengirimkan sinyal transmisinya hanya satu arah.



Gambar 2.23 Komunikasi satu arah

2. Half Duplex

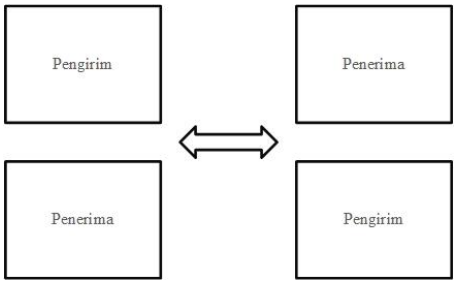
   Half Duplex merupakan komunikasi yang dapat mengirimkan data / informasi dari dua arah namun dilakukan secara bergantian.



Gambar 2.24 Komunikasi dua arah bergantian

3. Full Duplex

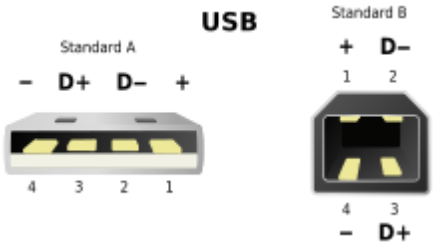
   Full Duplex merupakan komunikasi yang dapat mengirimkan data / informasi dari dua arah secara bersamaan. [11]



Gambar 2.25 Komunikasi dua arah bersamaan

2.4.2 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 komputer menggunakan konektor tipe A.



Gambar 2.26 Konektor USB

Berikut merupakan konfigurasi pin USB :

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 +.

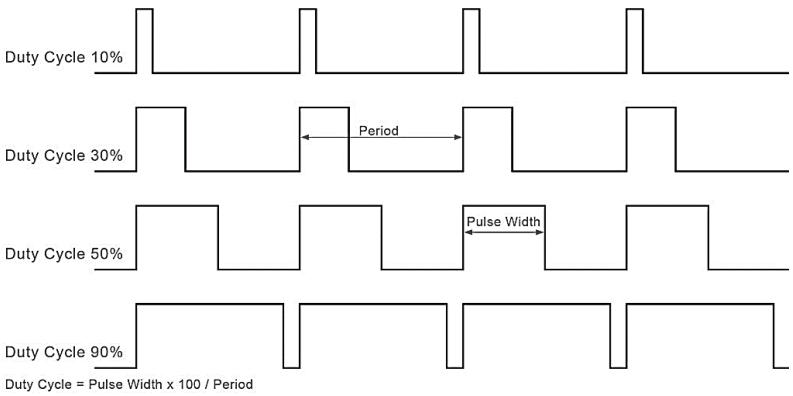
3. Data + = Saluran data, data akan aktif apabila data + > data –.

4. Ground = Saluran *Ground*.[17]

2.5 *Pulse Width Modulation* (PWM)

Pulse Width Modulation (PWM) adalah salah satu teknik modulasi dengan mengubah lebar pulsa (*duty cylce*) dengan nilai amplitudo dan frekuensi yang tetap. Satu siklus pulsa merupakan kondisi high kemudian berada di zona transisi ke kondisi low. Lebar pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. *Duty Cycle* merupakan representasi dari kondisi logika high dalam suatu periode sinyal dan di nyatakan dalam bentuk (%) dengan range 0% sampai 100%, sebagai contoh jika sinyal berada dalam kondisi high terus menerus artinya memiliki *duty cycle* sebesar 100%. Jika waktu sinyal keadaan high sama dengan keadaan low maka sinyal mempunyai *duty cycle* sebesar 50%.

Aplikasi penggunaan PWM biasanya ditemui untuk pengaturan kecepatan motor dc, pengaturan cerah/redup LED, dan pengendalian sudut pada motor servo. Contoh penggunaan PWM pada pengaturan kecepatan motor dc semakin besar nilai *duty cycle* yang diberikan maka akan berpengaruh terhadap cepatnya putaran motor. Apabila nilai *duty cylce*-nya kecil maka motor akan bergerak lambat.

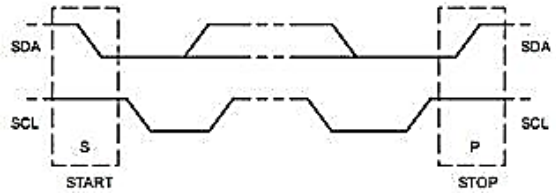


Gambar 2.27 Bentuk Sinyal Duty Cycle [18]

Ketika nilai *duty cycle* dirubah maka untuk perioda waktu pergerakan sistem akan mengalami perubahan.[19]

2.6 *Inter Intergrated Circuit* (I2C)

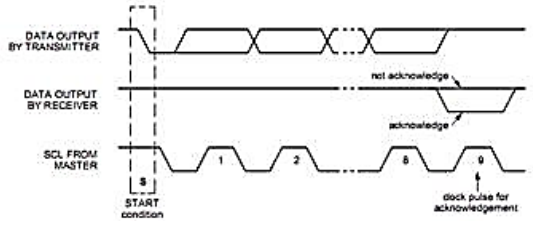
Inter Integrated Circuit atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (Serial Clock) dan SDA (Serial Data) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai Master dan Slave. Master adalah piranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal Start, mengakhiri transfer data dengan membentuk sinyal Stop, dan membangkitkan sinyal clock. Slave adalah piranti yang dialamati master.Sinyal Start merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari "1" menjadi "0" pada saat SCL "1". Sinyal Stop merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari "0" menjadi "1" pada saat SCL "1". Kondisi sinyal Start dan sinyal Stop seperti tampak pada **Gambar 2.26.**



Gambar 2.28 Kondisi sinyal start dan stop

Sinyal dasar yang lain dalam I2C Bus adalah sinyal acknowledge yang disimbolkan dengan ACK Setelah transfer data oleh master berhasil diterima slave, slave akan menjawabnya dengan mengirim sinyal acknowledge, yaitu dengan membuat SDA menjadi "0" selama siklus clock ke 9. Ini menunjukkan bahwa Slave telah

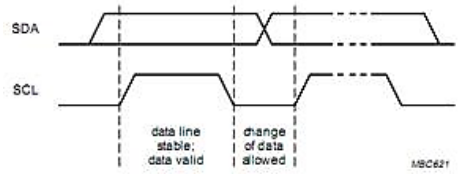
menerima 8 bit data dari Master. Kondisi sinyal acknowledge seperti tampak pada **Gambar 2.27.**



Gambar 2.29 Sinyal ACK dan NACK

Dalam melakukan transfer data pada I2C Bus, kita harus mengikuti tata cara yang telah ditetapkan yaitu:

- Transfer data hanya dapat dilakukan ketikan Bus tidak dalam keadaan sibuk.
- Selama proses transfer data, keadaan data pada SDA harus stabil selama SCL dalam keadan tinggi. Keadaan perubahan “1” atau “0” pada SDA hanya dapat dilakukan selama SCL dalam keadaan rendah. Jika terjadi perubahan keadaan SDA pada saat SCL dalam keadaan tinggi, maka perubahan itu dianggap sebagai sinyal Start atau sinyal Stop.



Gambar 2.30 Transfer Bit pada I2C Bus