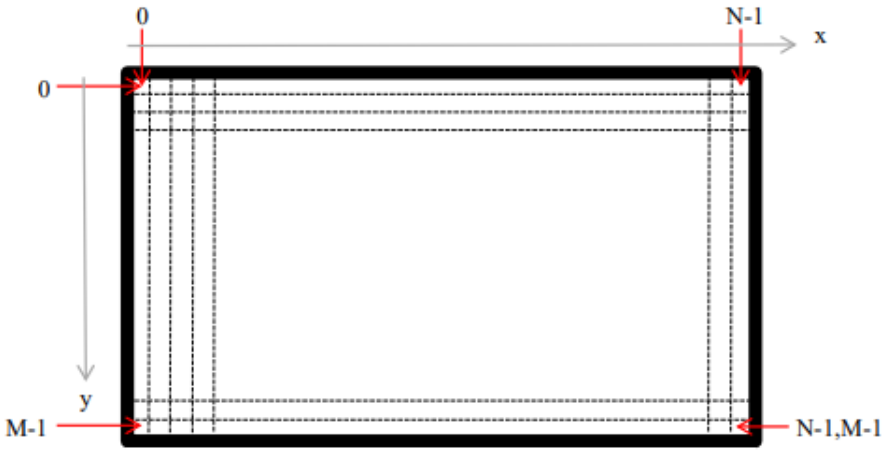
BAB 2
TINJAUAN PUSTAKA

2.1 Citra

Citra merupakan suatu matriks dari suatu *pixel* (*picture elements*) yang diatur dalam suatu kolom dan baris. Citra dapat didefinisikan sebagai fungsi $f(x,y)$ di mana x dan y merupakan koordinat bidang cartesius. Terdapat dua jenis citra, yaitu citra analog dan citra digital. Citra analog merupakan citra yang dihasilkan dari sistem optik yang menerima sinyal analog dan tidak dapat diolah langsung oleh komputer. Seperti citra yang tercetak di kertas atau kanvas. Sedangkan citra digital merupakan representasi gambar dua dimensi yang terdiri dari kumpulan titik-titik elemen gambar[6]. Pemrosesan citra digital dapat didefinisikan sebagai pemrosesan citra dengan menggunakan sebuah komputer digital untuk mendapatkan informasi yang dibutuhkan pada sebuah citra[6].

Setiap *pixel* digambarkan sebagai satu kotak kecil dan setiap elemen gambar mempunyai koordinat posisi. Berikut adalah sistem koordinat yang digunakan untuk menyatakan citra digital dapat dilihat pada Gambar 2.1.



Gambar 2.1 Sistem Koordinat Citra Berukuran M x N

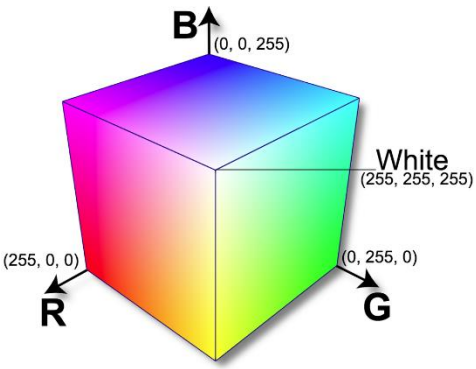
Dengan sistem koordinat yang mengikuti asas pemindaian pada layar standar, sebuah elemen gambar memiliki koordinat berupa sumbu x dan sumbu y, dimana sumbu x menyatakan posisi kolom sedangkan sumbu y menyatakan posisi baris.

Elemen gambar pada pojok kiri atas memiliki koordinat (0, 0) dan elemen gambar pada pojok kanan bawah memiliki koordinat (N-1, M-1).

2.1.1 Citra Berwarna

Citra berwarna merupakan jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Komponen RGB merupakan warna dasar yang dapat diterima oleh mata manusia. Setiap elemen gambar pada citra berwarna membutuhkan data sebesar 3 byte. Setiap warna memiliki intensitas sebesar 8 bit data yaitu dengan nilai minimum 0 dan nilai maksimal 255. Dengan demikian, kemungkinan warna yang dapat disajikan mencapai 255 x 255 x 255 atau 16.581.375 warna.

Ruang warna RGB biasa digunakan pada sistem grafik komputer dan kamera digital. Ruang warna ini dimodelkan dalam bentuk kubus tiga dimensi dengan pusat warna merah, hijau dan biru berada pada ujung sumbu R, G dan B. Warna hitam berada di titik asal dan warna putih berada pada titik sudut yang berseberangan dengan titik asal. Berikut adalah model ruang RGB dapat dilihat pada Gambar 2.2.



Gambar 2.2 Model Ruang RGB

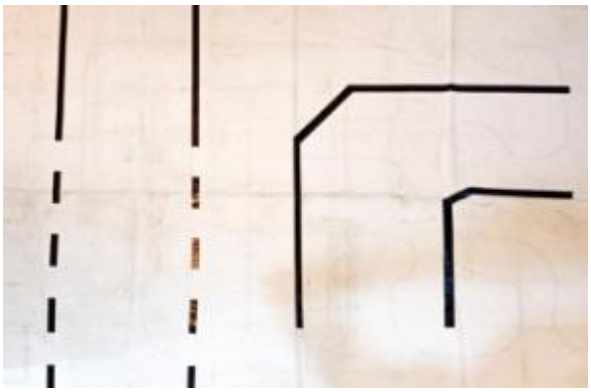
Berikut adalah tabel warna dan kode-kode atau nilai-nilai RGB dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tabel Warna RGB

Warna	R	G	B
Merah	255	0	0
Hijau	0	255	0

Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255

Berikut adalah gambar berwarna dapat dilihat pada Gambar 2.3.



Gambar 2.3 Gambar Berwarna

2.1.2 Citra Berskala Keabuan (*Grayscale*)

Citra dengan skala keabuan (*grayscale*) merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, artinya nilai dari *Red* + *Green* + *Blue*. Nilai-nilai tersebut digunakan untuk menunjukkan intensitas warna. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *grayscale* berbeda dengan citra biner (hitam putih), dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu “hitam” dan “putih” saja. Pada citra *grayscale* warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak.

Citra *grayscale* disimpan dalam format 8 bit untuk setiap elemen gambar, sehingga kemungkinan warna yang dapat disajikan sebanyak 256 warna. Secara umum citra berwarna dapat dikonversi ke citra berskala keabuan dapat dilihat pada rumus :

$$f(x,y) = \mathrm{a \times R} + \mathrm{b \times G} + \mathrm{c \times B} \ldots\ldots\ldots (1)$$

Dimana :

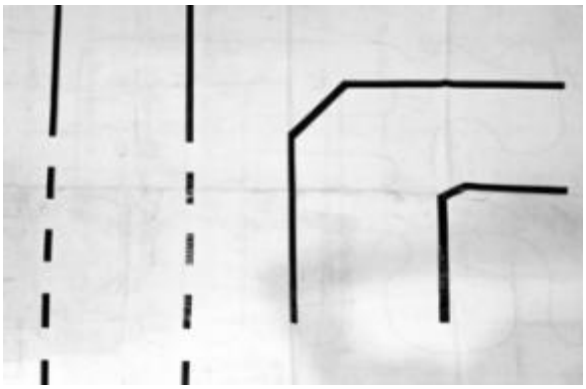
- $f(x, y)$ = fungsi gambar
- $\mathrm{a} + \mathrm{b} + \mathrm{c} = 0.2989 + 0.5870 + 0.1140 = 1$
- $\mathrm{R} =$ merepresentasikan nilai warna merah pada citra

- G = merepresentasikan nilai warna hijau pada citra
- B = merepresentasikan nilai warna biru pada citra

Bobot nilai pada koefisien a, b dan c ditetapkan secara proporsional dengan respon persepsi mata manusia untuk masing-masing warna merah, hijau dan biru, bobot standar yang telah ditentukan oleh NTSC (*National Television System Committee*) pada koefisien a = 0.2989, b = 0.5870, dan c = 0.1140. Dimana rumus matematis tersebut diadopsi dalam fungsi rgb2gray pada Python dan Matlab untuk konversi citra RGB ke citra *grayscale*. Berikut adalah visualisasi 256 aras keabuan dapat dilihat pada Gambar 2.4.

Gambar 2.4 Visualisasi 256 Aras Keabuan

Berikut adalah gambar skala keabuan (*grayscale*) dapat dilihat pada Gambar 2.5.



Gambar 2.5 Gambar Skala Keabuan (*Grayscale*)

2.1.3 Citra Biner

Citra biner (*binary image*) merupakan citra digital yang hanya memiliki dua kemungkinan warna, yaitu hitam dan putih. Citra biner disebut juga dengan citra W & B (*white & black*) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel dari citra biner. Pembentukan citra biner memerlukan nilai batas keabuan yang akan digunakan sebagai nilai patokan. *Pixel* dengan derajat keabuan lebih besar dari nilai batas akan diberi nilai 1 dan sebaliknya *pixel* dengan derajat keabuan lebih kecil dari nilai batas akan diberi

nilai 0. Fungsi dari binerisasi sendiri adalah untuk mempermudah proses pengenalan pola, karena pola akan lebih mudah terdeteksi pada citra yang mengandung lebih sedikit warna. Persamaan untuk binerisasi dapat dilihat pada rumus :

$$f(x,y) = \begin{cases} a1\ jika\ f(x,y) < T \\ a2\ jika\ f(x,y) \geq T \end{cases} \dots\dots\dots (2)$$

Dimana :

- $f(x, y)$ = fungsi gambar
- T = nilai ambang atau patokan (*threshold*)
- nilai $a1$ biasanya = 0 dan nilai $a2$ = 1

Dengan f(x,y) adalah citra biner dari citra *grayscale* f(x,y) dan T menyatakan nilai ambang (*threshold*). Nilai T memegang peranan yang sangat penting dalam proses *thresholding*. Kualitas citra biner sangat bergantung pada nilai T yang digunakan. Berikut adalah gambar biner dapat dilihat pada Gambar 2.6.

Gambar 2.6 Gambar Biner

2.2 *Computer Vision*

Computer Vision mempunyai tujuan ingin meniru cara kerja visual manusia (*human vision*). Pekerjaan yang biasa dilakukan oleh manusia seperti melihat melihat, mengontrol dan lain-lain. *Computer vision* merupakan suatu sistem yang mempunyai kemampuan untuk menganalisis sebuah objek secara visual, setelah data suatu objek yang bersangkutan dimasukkan dalam bentuk citra. Dalam

melakukan atau mengambil suatu keputusan *computer vision* memperoleh data dari hasil pengolahan citra[7].

Tahapan-tahapan dalam *computer vision* terbagi 3 tahapan yaitu : tahap pertama akuisisi citra digital, pada tahap ini dapat juga disebut sebagai tahap awal atau proses *image pre-processing*. Selanjutnya tahap kedua teknik komputasi di mana *pixel* dari sebuah data citra diolah dan dilakukan modifikasi. Tahap ketiga menganalisis yaitu citra yang telah melewati proses analisis hasilnya dapat digunakan untuk proses lain. Dalam penelitian ini, penulis menggunakan *computer vision* dengan tujuan untuk mendeteksi objek berupa garis dan kemudian menggunakan informasi tersebut untuk menghasilkan arah navigasi.

2.3 OpenCV 4.5.3

Open Source Computer Vision atau yang lebih dikenal dengan OpenCV merupakan API sumber terbuka yang berisi fungsi-fungsi pemrograman terkait tentang pengolahan citra digital. API ini berlisensi BSD sehingga bebas digunakan untuk keperluan akademik, penelitian, maupun komersial. OpenCV ditulis dalam bahasa C++, Python, dan Java. OpenCV mendukung sistem operasi Windows, Linux, Android, iOS dan Mac OS[8].

Sejak peluncuran OpenCV versi alfa pada januari 1999, OpenCV sudah digunakan pada berbagai aplikasi, produk dan riset untuk memecahkan masalah terkait pengolahan citra digital. Aplikasi-aplikasi tersebut antara lain pengolahan citra satelit, pengurangan derau pada citra medis dan sistem pemantau keamanan. OpenCV memiliki lebih dari 2500 algoritma yang telah dioptimalkan. Dalam penelitian ini, penulis menggunakan OpenCV 4.5.3 dengan bahasa pemrograman Python.

2.4 *Canny Edge Detection*

Canny Edge Detection merupakan proses pendeteksian tepi menggunakan metode canny. Metode canny adalah metode yang dikemukakan oleh John Canny pada tahun 1986, terkenal sebagai operator deteksi tepi yang sederhana dan baik sehingga optimal dalam tingkat akurasi[9]. Deteksi tepi canny adalah operasi yang

dijalankan untuk mendeteksi garis tepi (*edges*) yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda. Algoritma canny termasuk algoritma multi-tahap dengan tugas yaitu *noise reduction*, *gradient calculation*, *non-maximum suppression*, *double threshold* dan *edge tracking by hysteresis*. Beberapa kelebihan dari *canny edge detection* ini adalah sebagai berikut[9] :

1. Mendeteksi dengan baik

   Kemampuan untuk menandai semua edge yang ada dengan baik dan juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan *edges* sesuai yang diinginkan.
2. Melokalisasi dengan baik

   Dengan canny memungkinkan menghasilkan jarak yang minimum antara *edges* yang dideteksi dengan *edges* yang asli.
3. Respon yang jelas

   Hanya ada satu respon untuk setiap edges, sehingga mudah terdeteksi dan tidak menimbulkan kerancuan pada pengolahan citra selanjutnya.

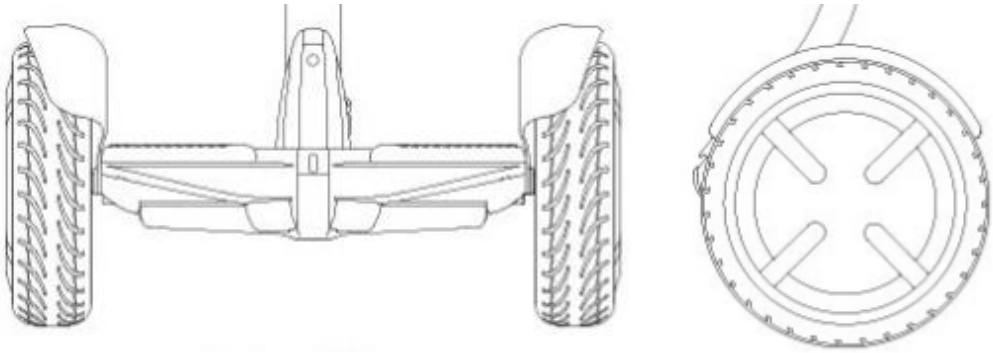
2.5 Perangkat Keras

Robot terdiri dari perangkat keras yang merupakan bagian mekanik berupa fisik robot, bagian aktuator berupa penggerak yang akan digunakan robot dan bagian elektronik berupa sensor serta komponen penunjang lainnya.

2.5.1 Sistem Kemudi Segway Ninebot

Sistem kemudi segway ninebot yang digunakan ini sangat umum dalam dunia nyata maupun dalam dunia robotika. Kemampuan manuver sistem kemudi segway ninebot masih terbatas namun mempunyai keunggulan pada arah dan geometri kemudi sangat baik dalam stabilitas lateral. Sistem kemudi segway ninebot menjadi salah satu sistem pergerakan yang sederhana dimana perputaran belok tergantung pada motor penggerak. Dengan satu kondisi dimana mekanisme belok harus dikontrol dengan tepat, kesalahan posisi kecil dalam mekanisme putaran belok akan menyebabkan kesalahan sistem kemudi dalam

bermanuver[10]. Berikut adalah sistem kemudi segway ninebot dapat dilihat pada Gambar 2.7.



(a) (b)

Gambar 2.7 Sistem Kemudi Segway Ninebot, Tampak Depan (a), Tampak Samping (b)

Sistem kemudi segway ninebot dikarakterisasi dengan kemampuan penyeimbangan oleh roda dan berayun dalam titik sekitar posisi keseimbangannya (*auto balancing*). *Auto balancing* tersebut merupakan mekanisme penyeimbangan posisi robot ikon secara tegak lurus terhadap permukaan bumi tanpa memerlukan gaya dari luar. Proses penyeimbangan dilakukan melalui pemberian sinyal kendali kepada dua buah roda.

2.5.2 Kamera

Kamera merupakan alat untuk mengakuisisi citra digital[11]. Data citra digital yang diperoleh dipergunakan robot ikon untuk diproses oleh sistem, parameter yang diambil dari sebuah citra pada penelitian ini adalah objek berupa garis tepi (*edges*). Kamera digunakan sebagai alat untuk menangkap gelombang cahaya yang dipantulkan oleh objek, sehingga objek tersebut dapat dipetakan dalam bentuk citra. Pada kamera digital, citra yang dihasilkan merupakan citra dalam bentuk digital. Secara umum kamera digital terdiri dari lensa, sensor cahaya, dan digitizer. Dalam penelitian ini, penulis menggunakan kamera Logitech HD Pro Webcam C920. Berikut adalah kamera dapat dilihat pada Gambar 2.8.

Gambar 2.8 Kamera

2.5.3 Mini PC

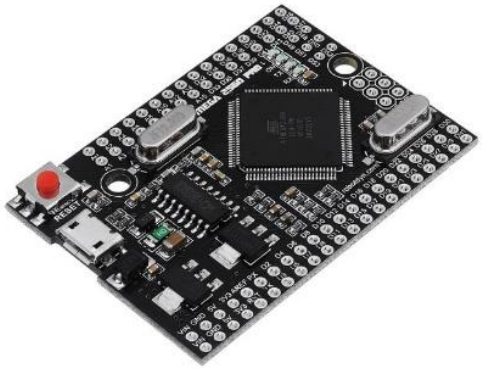
Mini PC merupakan sebuah *minicomputer*[12]. Pertimbangan dipilihnya mini pc dikarenakan ukurannya yang memungkinkan bisa di masukan ke dalam robot ikon. Performa dari mini pc yang hampir sama dengan komputer seperti pada umumnya, dengan sumber daya listrik DC (*direct current*) secara langsung, maka lebih memudahkan untuk menggunakan catu daya baterai yang bisa diambil dari baterai yang ada pada robot ikon. Dalam penelitian ini, penulis menggunakan Lenovo IdeaCentre Q190 Mini PC, pengiriman data dari mini pc ke mikrokontroler dan sebaliknya menggunakan protokol komunikasi serial (*serial communication*). Berikut adalah mini pc dapat dilihat pada Gambar 2.9.

Gambar 2.9 Mini PC

2.5.4 Mikrokontroler

Mikrokontroler merupakan sebuah *microchip* yang secara fisik berupa sebuah IC (*integrated circuit*). Mikrokontroler berisikan bagian-bagian utama yaitu CPU (*central processing unit*), RAM (*random-access memory*), ROM

(*read-only memory*) dan I/O (*input/output*). Selain bagian utama terdapat beberapa perangkat keras yang dapat digunakan untuk beberapa keperluan lainnya seperti melakukan pencacahan, komunikasi serial maupun interupsi[13]. Dalam penelitian ini, penulis menggunakan mikrokontroler Arduino Mega 2560 Pro. Berikut adalah mikrokontroler dapat dilihat pada Gambar 2.10.



Gambar 2.10 Mikrokontroler

2.5.5 Modul Bluetooth HC05

Modul Bluetooth HC05 merupakan modul komunikasi nirkabel via bluetooth yang dimana beroperasi pada frekuensi 2.4GHz dengan pilihan dua mode konektivitas[14]. Mode pertama berperan sebagai *slave* atau *receiver* data saja, sedangkan mode kedua berperan sebagai *master* atau dapat bertindak sebagai *transceiver*. Modul Bluetooth HC05 dipilih karena spesifikasi modul sangat cocok dengan komunikasi nirkabel atau *wireless*. Dalam penelitian ini, penulis menggunakan Modul Bluetooth HC05. Berikut adalah modul bluetooth HC05 dapat dilihat pada Gambar 2.11.

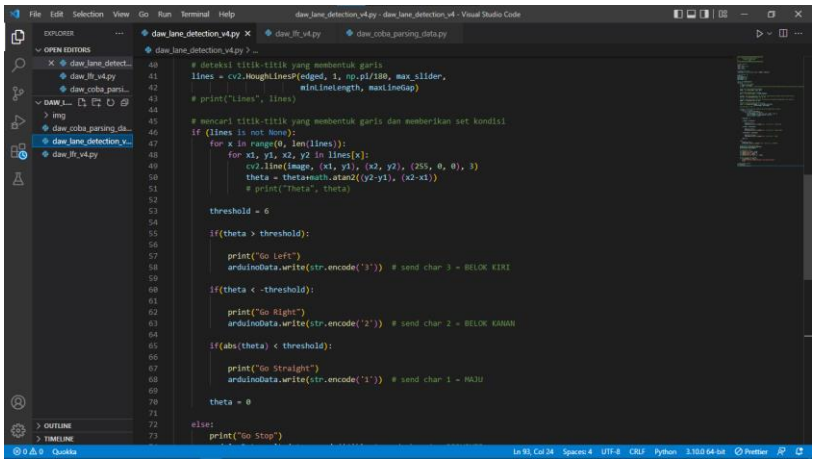
Gambar 2.11 Modul Bluetooth HC05

2.6 Perangkat Lunak

Dalam proses pembangunan sistem navigasi jarak dekat ini penulis menggunakan *software* diantaranya sebagai berikut.

2.6.1 Visual Studio Code

Visual Studio Code merupakan *Integrated Development Environment* (IDE) yang digunakan untuk mengembangkan program komputer, website dan aplikasi *mobile*[15]. Visual studio sudah mendukung *Editor Code IntelliSense* (komponen yang menyelesaikan secara otomatis) serta dengan *Code Refactoring* (restrukturisasi kode program komputer yang ada tanpa mengubah perilaku eksternalnya). Dengan dukungan banyak bahasa pemrograman yang berbeda dan memungkinkan melakukan *debug* ke berbagai bahasa pemrograman dengan syarat mempunyai layanan *language-specific*. Bahasa yang sudah *built-in* dalam visual studio yaitu C, C++, C++/CLI, Python, Visual Basic .NET, C#, F#, Javascript, TypeScript XML, XSLT, HTML, CSS dan sebagainya. Berikut adalah visual studio code dapat dilihat pada Gambar 2.12.



Gambar 2.12 Visual Studio Code

2.6.2 Arduino IDE

Arduino IDE pertama kali dikembangkan di *Ivrea Interaction Design Institute* sebagai alat yang mudah untuk membuat *prototype* secara cepat, yang di targetkan untuk bidang elektronik dan pemrograman. Arduino IDE merupakan program *open-source* dikhususkan untuk *platform* elektronik mikrokontroler. *Board* yang di buat arduino sepenuhnya *open-source* sehingga

pengguna bisa menyesuaikan dengan kebutuhan khusus tersendiri. Bahasa pemrograman yang dipakai arduino menggunakan bahasa pemrograman tingkat tinggi dengan bahasa pemrograman C[16]. Berikut adalah arduino IDE dapat dilihat pada Gambar 2.13.

Gambar 2.13 Arduino IDE