

## BAB 2

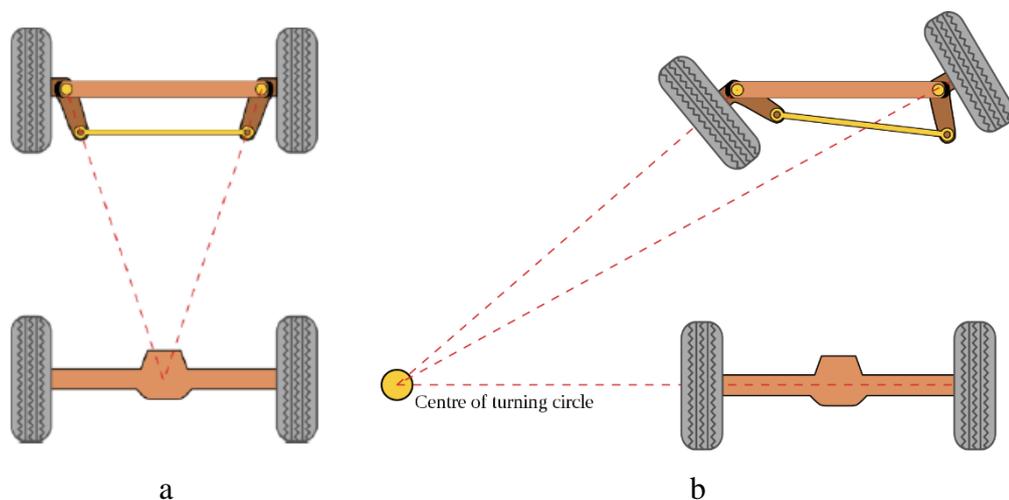
### TINJAUAN PUSTAKA

#### 2.1 Perangkat Keras

Sebuah robot tentunya memerlukan perangkat keras yang terdiri dari bagian mekanik berupa fisik robot, dan dibutuhkan bagian elektronik berupa sensor serta komponen penunjang lainnya[4].

##### 2.1.1 Sistem Kemudi Ackerman

Pertimbangan sistem kemudi ackerman yang digunakan dalam mobil ini sangat umum dalam dunia nyata tetapi tidak biasa dalam dunia robot. Kemampuan manuver kemudi ackerman yang terbatas memiliki keunggulan arah dan geometri kemudi sangat baik dalam stabilitas lateral dalam kecepatan tinggi. Sistem kemudi seperti mobil ini adalah salah satu sistem pergerakan yang paling sederhana dimana motor penggerak dan perputaran belok terpisah. Dengan satu kondisi dimana mekanisme belok harus dikontrol dengan tepat kesalahan posisi kecil dalam mekanisme putaran belok akan menyebabkan kesalahan odometri yang besar. Kelemahan dari sistem ackerman adalah tidak bisa melakukan rotasi terhadap sumbu vertikal dan radius belok yang dibatasi oleh sudut belok roda depan dan jarak antara roda belakang dengan roda depan[5]. Berikut adalah sistem kemudi ackerman dapat dilihat pada *Gambar 2.1*.



**Gambar 2.1 Sistem Kemudi Ackerman, Sistem Kemudi (a), Alur Kemudi (b)**

### **2.1.2 Mikrokontroler**

Mikrokontroler merupakan sebuah chip mikrokomputer 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 pecacahan, komunikasi serial atau interupsi. Mikrokontroler tertentu bahkan menyertakan ADC (*Analog-To-Digital Converter*), *USB Controller*, CAN (*Controller Area Network*)[6].

Mikrokontroler dapat bekerja berdasarkan program yang sudah di tanamkan didalamnya dengan program yang sesuai dengan aplikasi yang diinginkan. Program yang ditanamkan pada mikrokontroler merupakan intruksi-intruksi dalam bentuk kode dibuat dengan menggunakan bahasa pemrograman tertentu. Program biasanya dibuat di komputer yang selanjutnya dituliskan pada mikrokontroler menggunakan alat bantuan perangkat keras pemrograman sesuai dengan jenis mikrokontroler yang digunakan. Pemrograman mikrokontroler dapat dilakukan dengan menggunakan bahasa tingkat rendah (*Assembly*) atau bahasa tingkat tinggi (C, Basic dan lainnya).

### **2.1.3 Mini PC**

Mini PC merupakan komputer yang berukuran kecil, dipilihnya mini PC dikarenakan ukurannya yang memungkinkan bisa di masukan ke dalam sebuah robot. Performa dari mini PC hampir sama dengan komputer seperti pada umumnya, dengan sumber daya listrik DC secara langsung, maka lebih memudahkan untuk menggunakan catu daya baterai yang bisa diambil dari baterai yang ada pada robot.

### **2.1.4 Kamera**

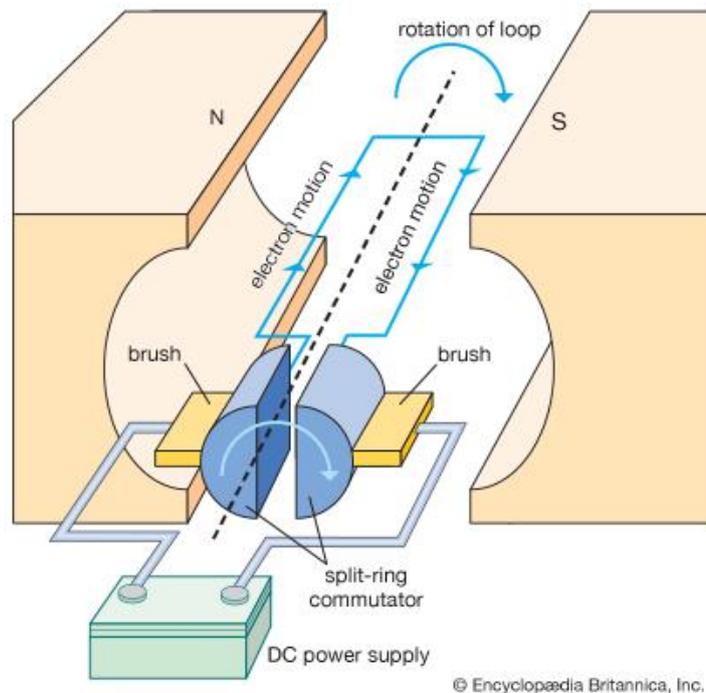
Kamera merupakan alat pengambilan citra digital. Citra digital yang diperoleh dipergunakan robot untuk diproses oleh sistem, parameter yang diambil dari sebuah citra pada penelitian ini adalah posisi objek, bentuk, resolusi, dan kecepatan.

## 2.1.5 Aktuator

Aktuator merupakan bagian penggerak yang akan digunakan robot. Aktuator robot yang dipakai pada penelitian ini terdiri dari dua jenis yaitu motor DC dan motor servo.

### 2.1.5.1 Motor DC

Motor DC merubah energi listrik DC menjadi energi mekanik putaran. Pada motor DC kumparan medan disebut stator (bagian yang tidak berputar) yang terdiri dari rangka dan kumparan medan dan kumparan jangkar disebut rotor (bagian yang berputar). Dari dua bagian ini dapat dibagi lagi menjadi beberapa komponen penting yaitu diantaranya adalah kerangka magnet (*yoke*), kutub motor (*poles*), kumparan medan magnet (*field winding*), kumparan jangkar (*armature winding*), komutator (*Commutator*) dan kuas/sikat (*brushes*). Prinsip kerjanya motor DC menggunakan fenomena elektromagnet untuk bergerak, ketika arus diberikan ke kumparan maka permukaan kumparan yang bersifat utara akan bergerak ke magnet yang berkutub selatan dan sebaliknya maka akan terjadi saling tarik menarik yang menyebabkan pergerakan kumparan berhenti. Berikut adalah komponen motor DC dapat dilihat pada *Gambar 2.2*.

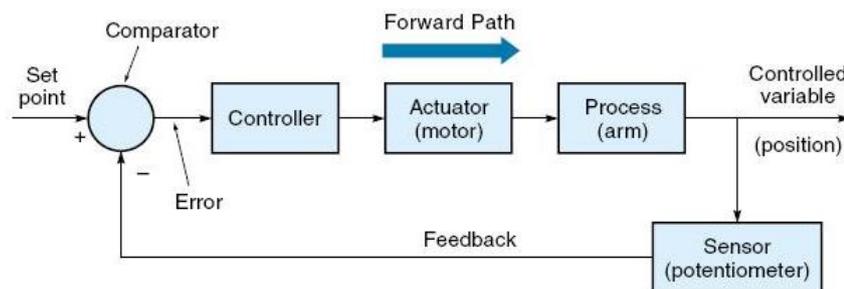


**Gambar 2.2** Komponen Motor DC

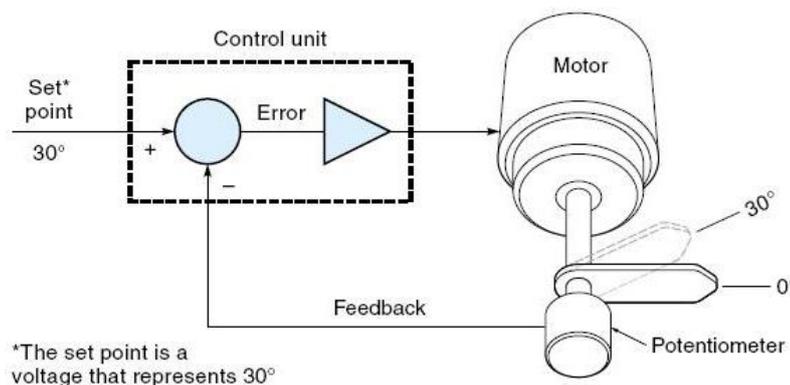
### 2.1.5.2 Motor Servo

Motor servo dijalankan dengan menggunakan *control loop* dan memerlukan sejumlah umpan balik. *Control loop* menggunakan umpan balik dari motor untuk membantu motor memperoleh keadaan (*state*) yang diinginkan. Ada beberapa jenis *control loop* salah satunya *control loop* PID (*proportional, integral, Derivative*)[1].

Motor servo terdiri dari beberapa bagian utama yaitu motor, *gearbox*, sensor posisi, *error amplifier* dan motor driver serta IC (*Integrated Circuit*) yang mengkode posisi yang diminta. *Radio control receiver system* membangkitkan suatu pulsa dan lebarnya berubah sekitar 20ms. Pulsa ini lebarnya biasanya antara 1 dan 2 ms. Lebar pulsa digunakan oleh servo untuk menentukan posisi rotasi yang dikehendaki[7]. Berikut adalah diagram blok motor servo dapat dilihat pada *Gambar 2.3*.

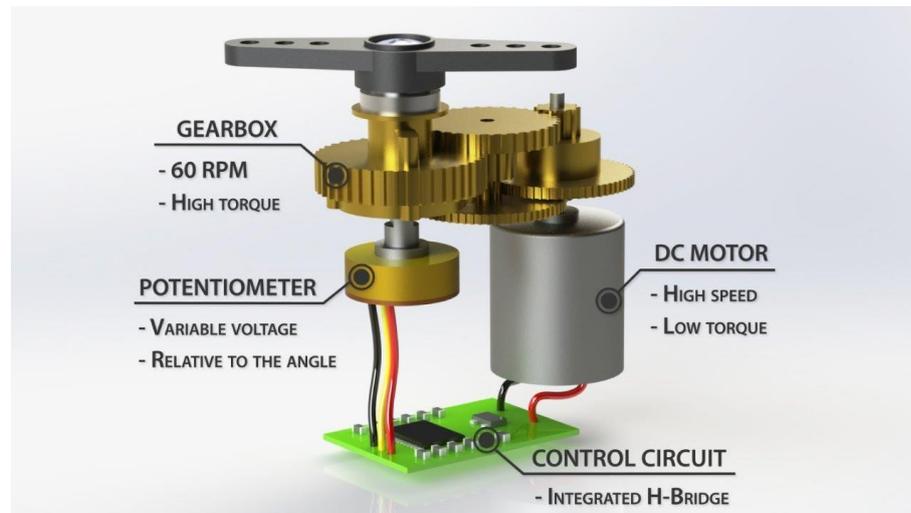


(a) Block diagram



**Gambar 2.3 Diagram Blok Motor Servo**

Berikut adalah kontruksi yang akan menggambarkan motor servo dapat dilihat pada *Gambar 2.4*.



**Gambar 2.4 Kontruksi Motor Servo**

### 2.1.6 Sensor Ultrasonik

Sensor ultrasonik terdiri dari dua rangkaian pemancar, yaitu rangkaian pengirim (*transmitter*) dan rangkaian penerima (*receiver*). Gelombang ultrasonik merupakan gelombang akustik yang memiliki frekuensi dari 20 kHz sampai 40 kHz. Prinsip kerja dari sensor ultrasonik yaitu ketika *transmitter* memancarkan gelombang ultrasonik disaat yang sama sensor akan menghasilkan *output* transisi naik yang menandakan sensor mulai menghitung waktu, setelah *receiver* menerima pantulan yang dihasilkan oleh suatu objek maka waktu akan dihentikan dengan menghasilkan *output* transisi turun. Jika waktu adalah  $t$  dan kecepatan suara adalah 340 m/s, maka jarak antara sensor dengan objek dihitung.

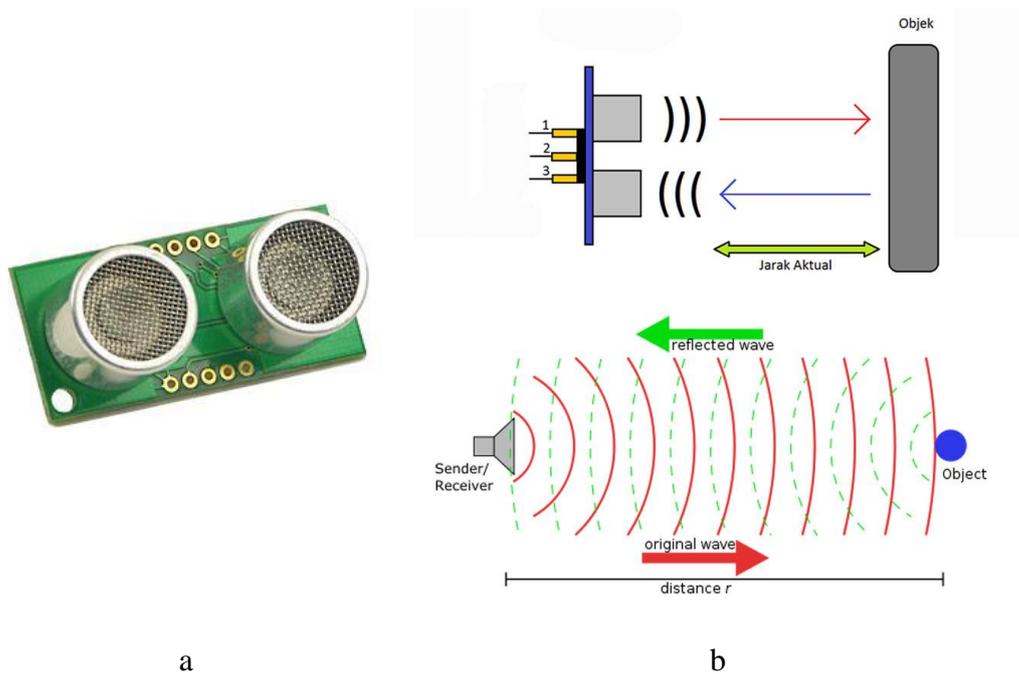
$$s = \frac{t * 340 \text{ m/s}}{2}$$

Dimana :

$s$  = jarak antara sensor dengan objek (meter)

$t$  = waktu gelombang ultrasonik dari *transmitter* ke *receiver* (s)

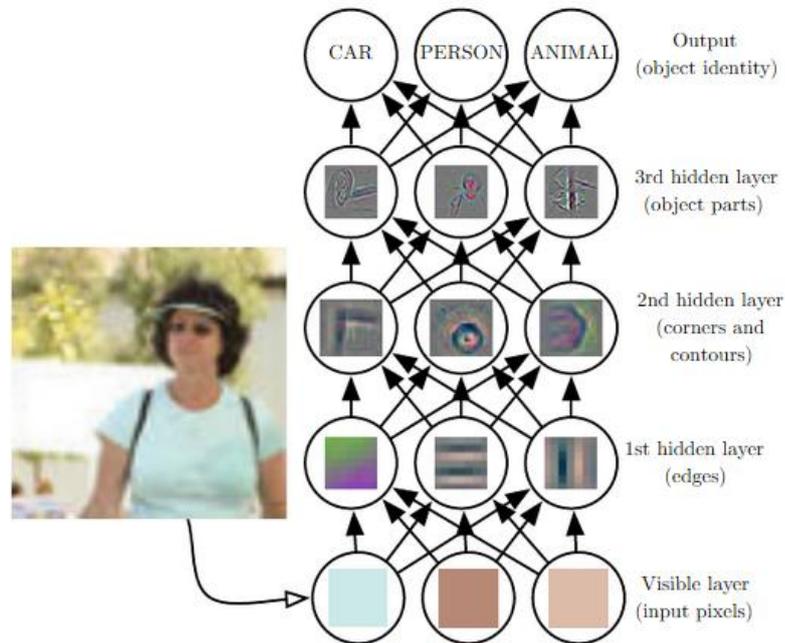
Berikut adalah contoh sensor ultrasonik dan prinsip kerja dari sensor ultrasonik dapat dilihat pada *Gambar 2.5*.



**Gambar 2.5 Sensor Ultrasonik, Contoh Sensor Ultrasonik (a), Prinsip Kerja Sensor Ultrasonik (b)**

## 2.2 *Deep Learning*

*Deep learning* merupakan salah satu dari algoritma *machine learning* dengan menggunakan beberapa lapisan secara progresif mengekstrak fitur tingkat tinggi dari input mentah. *Deep learning* mengatasi permasalahan dari representasi dengan memperkenalkan representasi yang diekspresikan dalam bentuk representasi lain yang lebih sederhana. Contoh model *deep learning* adalah *feedforward deep network* atau *multilayer perceptron* (MLP). Sebuah MLP hanyalah sebuah fungsi matematika memetakan beberapa set nilai input ke nilai output[8]. Berikut adalah bagaimana sistem *deep learning* dapat mewakili konsep gambar seseorang dengan menggabungkan konsep yang lebih sederhana dapat dilihat pada *Gambar 2.6*.



**Gambar 2.6 Ilustrasi Model *Deep Learning***

Dari *Gambar 2.6* komputer sulit untuk memahami arti dari data input mentah tersebut. Gambar direpresentasikan sebagai himpunan dari nilai *pixel*. Pemetaan fungsi dari satu set *pixel* ke identitas objek sangat rumit. Evaluasi pemetaan ini tampak tidak dapat diatasi jika ditangani secara langsung. *Deep learning* menyelesaikan kesulitan ini dengan memecah pemetaan yang rumit menjadi serangkaian pemetaan bersarang yang lebih sederhana, masing-masing dijelaskan oleh lapisan model yang berbeda. Input di masukan dalam *visible layer*, karena dapat di diamati. Kemudian *hidden layer* mengekstrak fitur abstrak dari gambar. Dinamakan *hidden layer* karena nilai tidak memeberikan data sehingga model harus menentukan pilihan konsep yang berguna untuk menjelaskan relasi dari data yang di amati. Dari *hidden layer* pertama bisa dengan mudah mengidentifikasi garis tepi dengan membandingkan *brightness* dari *pixel* per *pixel*. *Hidden layer* kedua dengan mudah mencari sudut dan contours dimana didapatkan dari garis tepi yang sudah diidentifikasi. *Hidden layer* ketiga dapat mendeteksi beberapa bagian objek, dengan menemukan objek yang spesifik dari sudut dan contours. Sehingga akhirnya deskripsi dari objek yang dikandungnya bisa digunakan untuk mengenali objek yang ada pada gambar.

### 2.3 YOLOv3

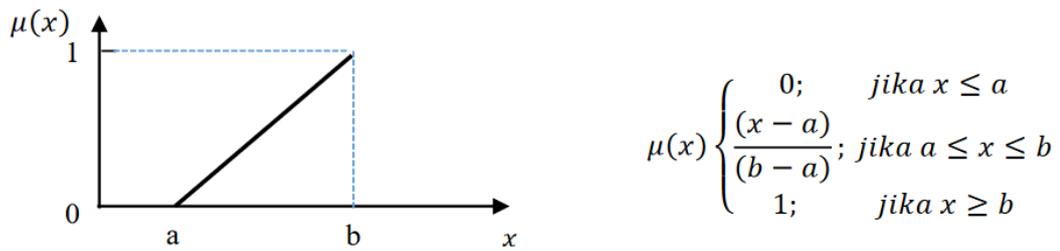
*YOLO (You Only Look Once)* merupakan sebuah algoritma *deep learning* deteksi objek yang ditargetkan untuk pemrosesan secara *real-time*[9]. Sistem kerja dari *YOLOv3* dengan cara mengulang sistem *classifiers* atau *localizers* untuk melakukan pendeteksian. Dengan menerapkan model ke sebuah gambar di berbagai lokasi dan berbagai skala. Daerah dengan nilai tertinggi dari sebuah gambar akan dianggap sebagai objek yang terdeteksi.

Dengan menggunakan pendekatan yang sama sekali berbeda. *YOLO* menerapkan *single neural network* ke dalam sebuah gambar. Jaringan tersebut dibagi menjadi beberapa daerah (*regions*) dan memprediksi *bounding box* serta mencari probabilitas untuk setiap daerah (*regions*). Dengan *bounding box* yang diberi bobot dari probabilitas yang diprediksi.

### 2.4 Logika Fuzzy

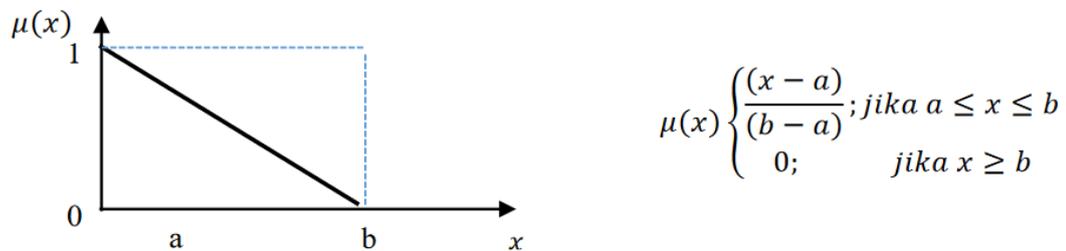
Logika *fuzzy* merupakan suatu cara tepat untuk memetakan suatu ruang *input*, ke dalam suatu ruang *output* dengan memakai *if-then rules*[10]. Himpunan *fuzzy (fuzzy set)* nilai keanggotaan terletak pada tentang 0 sampai 1, himpunan tegas (*crisp*) nilai keanggotaan hanya terdapat dua yaitu 0 dan 1. Jika  $x$  merupakan nilai keanggotaan *fuzzy*  $\mu_A[x] = 0$  yang berarti  $x$  tidak menjadi himpunan A, apabila  $x$  merupakan nilai keanggotaan *fuzzy*  $\mu_A[x] = 1$  yang berarti  $x$  menjadi anggota penuh himpunan A, himpunan *fuzzy* memiliki dua atribut yaitu linguistik dan numeris[10].

Fungsi keanggotaan (*membership function*) merupakan kurva yang menunjukkan pemetaan *input* ke dalam nilai keanggotaannya (derajat keanggotaan) yang memiliki rentang interval 0 sampai 1. Ada beberapa fungsi keanggotaan yang dapat digunakan yaitu fungsi keanggotaan linear, fungsi keanggotaan segitiga, dan fungsi keanggotaan trapesium[10]. Berikut adalah grafik fungsi keanggotaan himpunan representasi linear naik dapat dilihat pada *Gambar 2.7*.



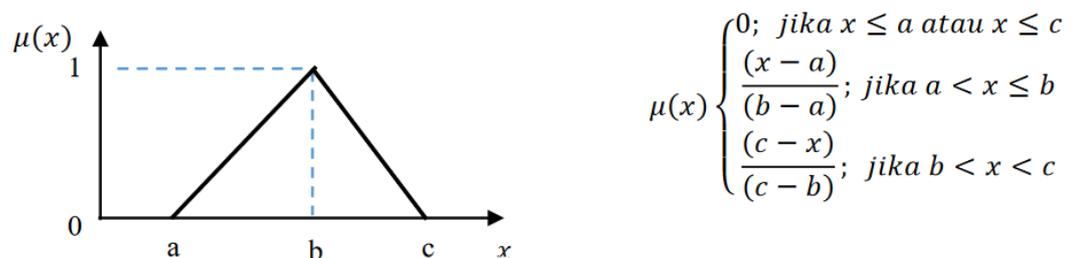
**Gambar 2.7 Grafik Fungsi Keanggotaan Himpunan Representasi Linear Naik**

Berikut adalah grafik fungsi keanggotaan himpunan representasi linear turun dapat dilihat pada *Gambar 2.8*.



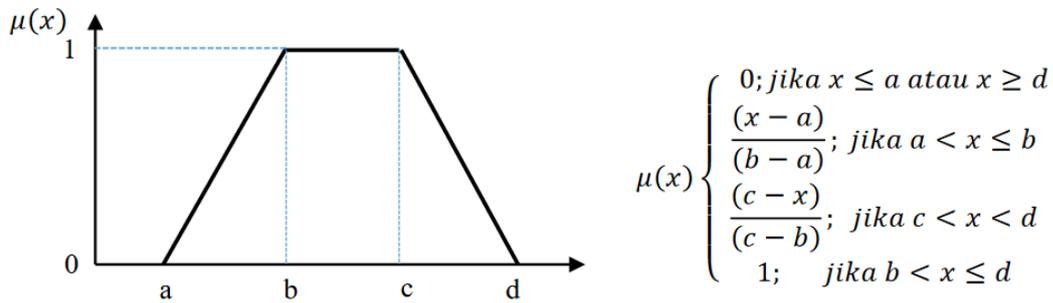
**Gambar 2.8 Grafik Fungsi Keanggotaan Himpunan Representasi Linear Turun**

Berikut adalah grafik fungsi keanggotaan himpunan representasi segitiga dapat dilihat pada *Gambar 2.9*.



**Gambar 2.9 Grafik Fungsi Keanggotaan Himpunan Representasi Segitiga**

Berikut adalah grafik fungsi keanggotaan himpunan representasi trapesium dapat dilihat pada *Gambar 2.10*.



**Gambar 2.10 Grafik Fungsi Keanggotaan Respresentasi Trapesium**

## 2.5 Metode Fuzzy Tsukamoto

Pada metode Tsukamoto pembentukan aturan (*rules*) secara umum dalam bentuk *IF-THEN* bentuk model *fuzzy* Tsukamoto adalah *IF(X IS A) and (Y IS B) and (Z IS C)*, dimana A, B, dan C merupakan himpunan *fuzzy*. Dengan mesin inferensi menggunakan fungsi implikasi *MIN* untuk mendapatkan  $\alpha$ -predikat dari setiap aturan (*rules*). Masing-masing dari nilai  $\alpha$ -predikat digunakan untuk menghitung *output* hasil dari inferensi secara tegas(*crisp*). Defuzzyfikasi dengan menggunakan metode rata-rata terpusat (*Center Average*)[11]:

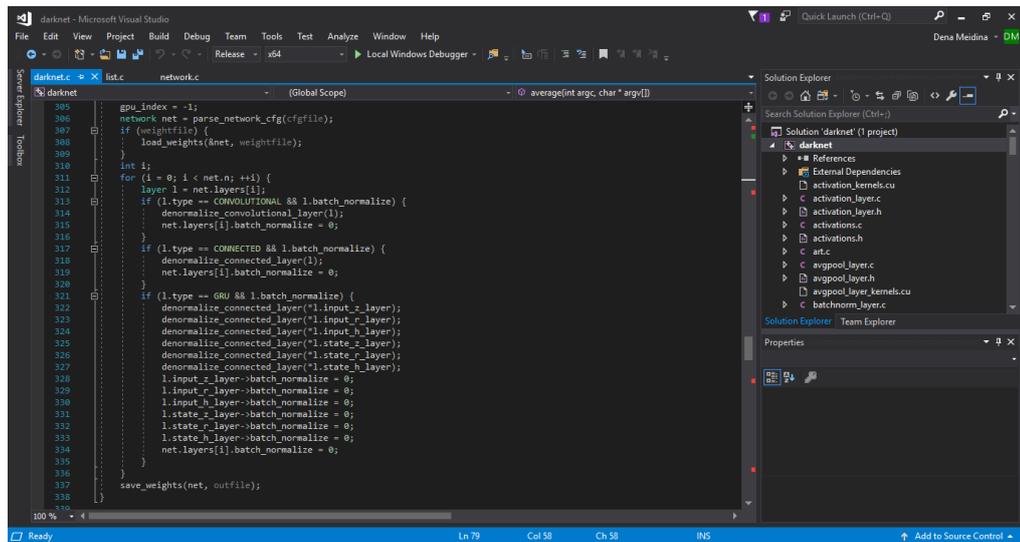
$$Z = \frac{\sum \alpha_i \cdot z_i}{\sum \alpha_i}$$

## 2.6 Perangkat Lunak

Dalam proses pembangunan sistem navigasi jarak dekat ini penulis menggunakan software bantuan, yaitu diantaranya sebagai berikut.

### 2.6.1 Visual Studio 2017

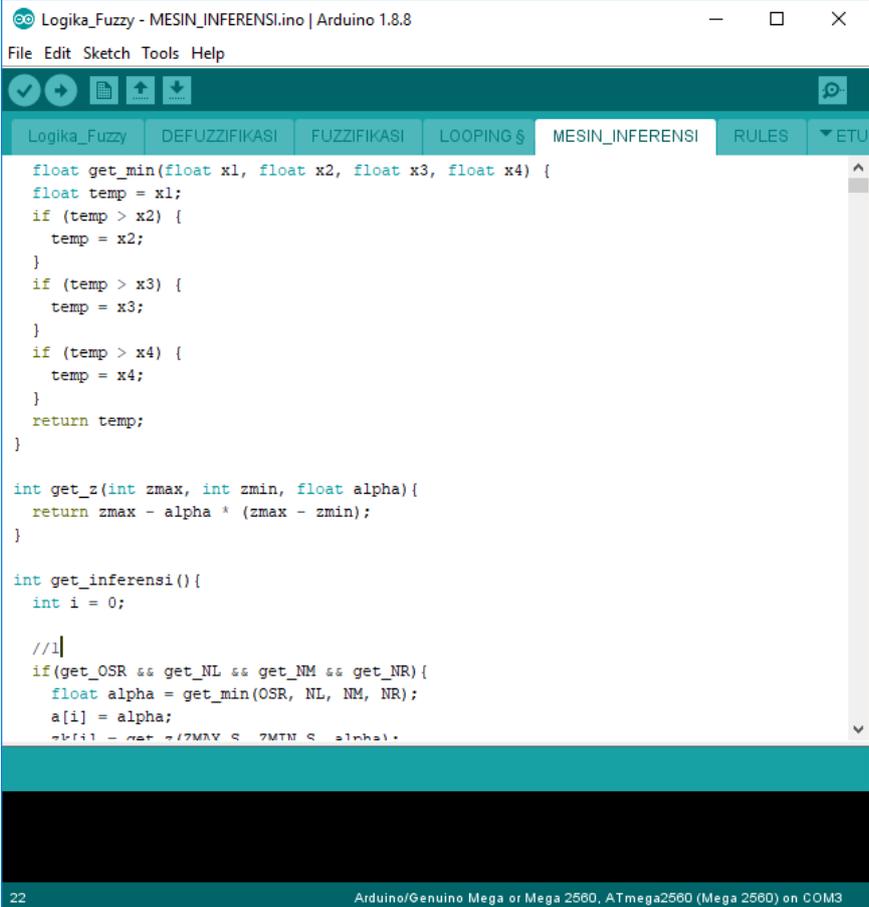
*Visual studio* adalah *Integrated Development Environment* (IDE) salah satu produk keluaran dari *Microsoft*. Digunakan untuk mengembangkan program komputer, website, serta aplikasi mobile. *Visual studio* sudah mendukung *editor code IntelliSense* (komponen yang menyelesaikan secara otomatis) serta dengan *code refactoring*. Dengan dukungan 36 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, Visual Basic .NET, C#, F#, Javascript, TypeScript XML, XSLT, HTML dan CSS. Berikut adalah tampilan *visual studio* 2017 dapat dilihat pada *Gambar 2.11*.



**Gambar 2.11 Tampilan Visual Studio 2017**

## 2.6.2 Arduino IDE

Arduino merupakan program *open-source* dikembangkan untuk platform elektronik mikrokontroler. Arduino pertama kali dikembangkan di Ivrea Interaction Design Institute sebagai alat yang mudah untuk membuat prototipe dan cepat, yang di targetkan untuk siswa dengan tanpa latar belakang di bidang elektronik dan pemrograman. Setelah mencapai komunitas yang lebih luas, Arduino memulai untuk beradaptasi dengan kebutuhan dan tantangan baru. *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[12]. Berikut adalah tampilan arduino dapat dilihat pada *Gambar 2.12*.



The image shows a screenshot of the Arduino IDE interface. The title bar reads "Logika\_Fuzzy - MESIN\_INFERENSI.ino | Arduino 1.8.8". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for saving, running, and uploading. The main editor area displays the following C++ code:

```
float get_min(float x1, float x2, float x3, float x4) {  
  float temp = x1;  
  if (temp > x2) {  
    temp = x2;  
  }  
  if (temp > x3) {  
    temp = x3;  
  }  
  if (temp > x4) {  
    temp = x4;  
  }  
  return temp;  
}  
  
int get_z(int zmax, int zmin, float alpha){  
  return zmax - alpha * (zmax - zmin);  
}  
  
int get_inferensi(){  
  int i = 0;  
  
  //1|  
  if(get_OSR && get_NL && get_NM && get_NR){  
    float alpha = get_min(OSR, NL, NM, NR);  
    a[i] = alpha;  
    z[i] = get_z(ZMAX[i], ZMIN[i], alpha);  
  }  
}
```

The status bar at the bottom indicates "22" and "Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3".

**Gambar 2.12** Tampilan Arduino