

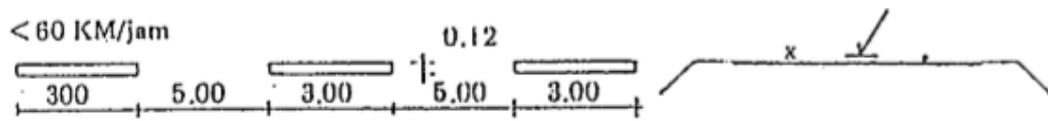
BAB II

TINJAUAN PUSTAKA

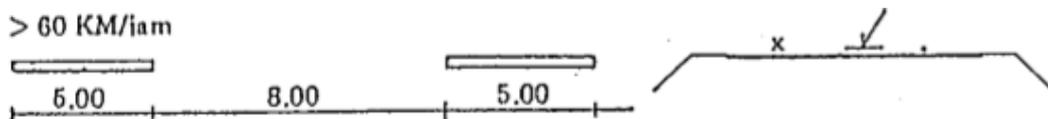
2.1 Marka Jalan

Marka Jalan adalah suatu tanda yang berada di permukaan jalan atau di atas permukaan jalan yang meliputi peralatan atau tanda yang membentuk garis membujur, garis melintang, garis serong serta lambang lainnya yang berfungsi untuk mengarahkan arus lalu lintas dan membatasi daerah kepentingan lalu lintas [3].

Marka jalan memiliki jenis-jenis garis yang berbeda dengan fungsi yang berbeda-beda pula. Sebagai contoh adalah garis terputus yang berfungsi sebagai garis sumbu dan pemisah pada jalan. Berikut adalah jalan dua lajur dan dua lajur dapat dilihat pada Gambar II.1 dan Gambar II.2 [3]:



Gambar II.1 Jalan 2 jalur, 2 arah dengan lebar > 5,50m [3]



Gambar II.2 Jalan lebih dari 2 jalur [3]

2.2 Sistem Lane Detection

Lane Detection merupakan suatu metode untuk mengetahui lokasi dari marka jalan tanpa diketahui terlebih dahulu noise yang terdapat pada lingkungan sekitarnya. Lane detection ini telah menjadi penelitian yang sering dilakukan oleh banyak orang agar dapat menjadi satu pendukung Driver Assistant maupun untuk Autonomous Navigation yang termasuk bagian dari Intelligent Transportation System [4].

2.2.1 Pengolahan Citra

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Sebagai contoh, citra burung

nuri pada Gambar II.3 tampak agak gelap, lalu dengan operasi pengolahan citra kontrasnya diperbaiki sehingga menjadi lebih terang dan tajam [5].



Gambar II.3 Citra Nuri Tampak Gelap



Gambar II.4 Citra Nuri Lebih Terang dan Tajam

Pengolahan Citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan. Termasuk ke dalam bidang ini juga adalah pemampatan citra (image compression) [5].

2.2.1.1 Operasi Pengolahan Citra

Operasi-operasi yang dilakukan di dalam pengolahan citra banyak ragamnya [5]. Namun, secara umum, operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis sebagai berikut:

1. Perbaikan kualitas citra (*image enhancement*)

Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Dengan operasi ini, ciri-ciri khusus yang terdapat di dalam citra lebih ditonjolkan [5]. Contoh-contoh operasi perbaikan citra:

- a. Perbaikan kontras gelap/terang
- b. Perbaikan tepian objek (*edge enhancement*)
- c. Penajaman (*sharpening*)
- d. Pembrian warna semu (*pseudocoloring*)
- e. Penapisan derau (*noise filtering*)

Gambar II.5 dan II.6 adalah contoh operasi penajaman. Operasi ini menerima masukan sebuah citra yang gambarnya hendak dibuat tampak lebih tajam. Bagian citra yang ditajamkan adalah tepi-tepi objek.



Gambar II.5 Citra Lena Asli



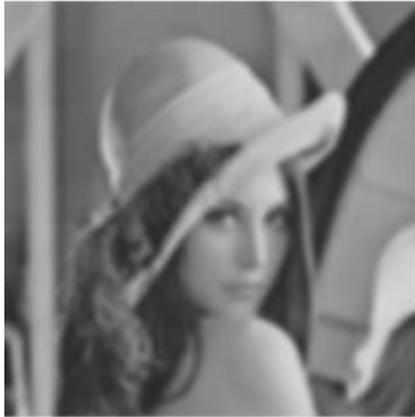
Gambar II.6 Citra Lena Setelah Ditajamkan

2. Pemugaran citra (*image restoration*)

Operasi ini bertujuan menghilangkan/meminimumkan cacat pada citra. Tujuan pemugaran citra hampir sama dengan operasi perbaikan citra [5]. Bedanya, pada pemugaran citra penyebab degradasi gambar diketahui. Contoh-contoh operasi pemugaran citra:

- a. Penghilangan kesamaran (*deblurring*)
- b. Penghilangan derau (*noise*)

Gambar II.7 dan II.8 adalah contoh operasi penghilangan kesamaran. Citra masukan adalah citra yang tampak kabur (*blur*). Kekaburan gambar mungkin disebabkan pengaturan fokus lensa yang tidak tepat atau kamera bergoyang pada pengambilan gambar. Melalui operasi deblurring, kualitas citra masukan dapat diperbaiki sehingga tampak lebih baik.



Gambar II.7 Citra Lena Yang Kabur (*blur*)



Gambar II.8 Citra Lena Setelah *Deblurring*

3. Pemampatan citra (*image compression*)

Jenis operasi ini dilakukan agar citra dapat direpresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Hal penting yang harus diperhatikan dalam pemampatan adalah citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus. Contoh metode pemampatan citra adalah metode JPEG. Perhatikan Gambar II.9 dan II.10. Gambar II.9 adalah citra kapal yang berukuran 258 KB. Hasil pemampatan citra dengan metode JPEG dapat mereduksi ukuran citra semula sehingga menjadi 49 KB saja [5].



Gambar II.9 Citra Boat.bmp (258KB) Sebelum Dimampatkan



Gambar II.10 Citra Boat.jpg (49KB) Sesudah Dimampatkan

4. Segmentasi citra (*image segmentation*)

Jenis operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Jenis operasi ini berkaitan erat dengan pengenalan pola [5].

5. Pengorakan citra (*image analysis*)

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik pengorakan citra mengekstraksi ciri-ciri tertentu yang membantu dalam identifikasi objek. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya [5]. Contoh-contoh operasi pengorakan citra:

- a. Pendeteksian tepi objek (*edge detection*)
- b. Ekstraksi batas (*boundary*)

c. Representasi daerah (*region*)

Gambar II.11 dan II.12 adalah contoh pendeteksian tepi pada citra Camera. Operasi ini menghasilkan semua tepi (*edge*) di dalam citra.



Gambar II.11 Citra Camera Asli



Gambar II.12 Citra Hasil Pendeteksian Seluruh Tepi

6. Rekontruksi citra (*image reconstruction*)

Jenis operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis. Misalnya beberapa foto rontgen dengan sinar X digunakan untuk membentuk ulang gambar organ tubuh [5].

2.2.2 Transformasi Hough

Transformasi Hough menspesifikasikan kurva dalam bentuk parametrik. Kurva dinyatakan sebagai bentuk parametrik $(x(u), y(u))$ dari parameter u . Bentuk parametrik tersebut menspesifikasikan titik-titik sepanjang kurva dari titik awal kurva $p_1 = (x(u_1), y(u_1))$ ke titik akhir $p_2 = (x(u_2), y(u_2))$. Panjang kurva adalah:

$$L = \int_{u_1}^{u_2} \sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2} du \quad (2.1)$$

Transformasi Hough menggunakan mekanisme voting untuk mengestimasi nilai parameter. Setiap titik di kurva menyumbang suara untuk beberapa kombinasi parameter. Parameter yang memperoleh suara terbanyak terpilih sebagai pemenang [5].

Pada awalnya, Transformasi Hough digunakan untuk mendeteksi garis lurus. Namun, ia juga dapat digunakan untuk mendeteksi kurva sederhana lainnya seperti lingkaran dan elips. Pembahasan dimulai dengan transformasi Hough untuk mendeteksi keberadaan garis lurus di dalam citra tepi [5].

2.2.2.1 Mendeteksi Garis

Misalkan citra tepi berukuran $n = N \times M$ pixel. Cara yang paling sederhana mendeteksi garis lurus adalah menemukan semua garis yang ditentukan oleh dua buah pixel dan memeriksa apakah sebagian dari pixel tepi termasuk ke dalam garis tersebut (cara exhaustive search) [5].

Jumlah maksimum garis yang dideteksi adalah $n(n-1)/2$. Karena setiap pixel harus diperiksa apakah ia termasuk ke dalam suatu garis, maka kompleksitas algoritma pendeteksian garis lurus untuk kasus terburuk adalah $O(n^3)$. Untuk aplikasi praktis, jelas metode pendeteksian dengan cara ini tidak mangkus [5].

Transformasi Hough mengurangi kompleksitas komputasi dengan menggunakan bentuk parametrik dan menggunakan mekanisme pemungutan suara terbanyak (voting) untuk menentukan nilai parameter yang tepat [5].

Tinjau persamaan garis lurus:

$$y = mx + c \quad (2.2)$$

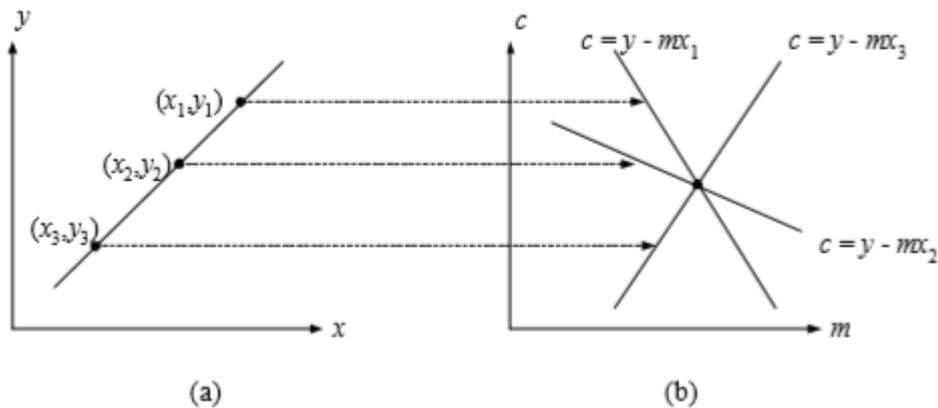
Dalam bentuk parametrik, setiap garis dinyatakan sebagai (m', c') di dalam ruang parameter m - c . Persamaan 2.2 dapat ditulis menjadi

$$c = y - mx \quad (2.3)$$

Sembarang titik (x,y) pada bidang planar X - Y berkoresponden dengan sebuah garis lurus pada ruang parameter m - c .

Tinjau 3 buah titik pada sebuah garis lurus pada Gambar II.13. Sembarang garis yang melalui titik (x_1, y_1) berkoresponden dengan garis $c = y_1 - mx_1$ pada ruang parameter m - c . Begitu juga, sembarang garis lurus yang melalui (x_2, y_2) berkoresponden dengan garis $c = y_2 - mx_2$ dan sembarang garis lurus yang melalui (x_3, y_3) berkoresponden dengan garis $c = y_3 - mx_3$ pada ruang m - c . Perpotongan (m', c') dari ketiga garis pada ruang m - c tersebut menentukan garis unik yang melalui (x_i, y_i) , $i = 1, 2, 3$, di bidang X - Y [5].

Dengan cara ini, maka setiap pixel pada garis lurus di bidang citra berkoresponden dengan sejumlah garis lurus yang melalui satu titik tertentu di ruang parameter m - c . Sifat ini dimanfaatkan untuk mendeteksi garis lurus. Jika setiap pixel tepi melakukan “pemungutan suara” pada ruang parameter, maka keberadaan garis lurus pada citra ditandai dengan penumpukan suara pada tempat-tempat tertentu di ruang parameter [5].



Gambar II.13 (a) Garis Lurus Pada Ruang X-Y; (b) Representasinya dalam ruang parameter m - c

Karena itu, prosedur mendeteksi garis lurus adalah sebagai berikut:

1. Ruang parameter didiskritkan sebagai matriks $P(m, c)$, yang dalam hal ini $m_l \leq m \leq m_K$ dan $c_l \leq c \leq c_L$.
2. Tiap elemen pada ruang parameter diasumsikan sebagai akumulator. Inisialisasi setiap elemen $P(m, c)$ dengan 0.
3. Untuk setiap pixel tepi (x_i, y_i) –pixel tepi dicirikan mempunyai nilai intensitas putih (1) dalam skala 0 - 1)– hitung nilai $c = y_i - mx_i$. Untuk setiap nilai parameter m , $m_l \leq m \leq m_K$, yang berkoresponden dengan nilai c , maka elemen matriks $P(m, c)$ yang bersesuaian dinaikkan satu:

$$P(m, c) = P(m, c) + 1 \quad (2.4)$$

Dengan kata lain, tambahkan satu suara pada ruang parameter m - c .

4. Ulangi langkah 3 sampai seluruh pixel di dalam citra tepi ditelusuri.
5. Pada akhir prosedur, tiap elemen matriks $P(m, c)$ menyatakan jumlah pixel tepi yang memenuhi persamaan (1). Tentukan elemen matriks yang memiliki penumpukan suara cukup besar (yang nilainya di atas nilai ambang tertentu).

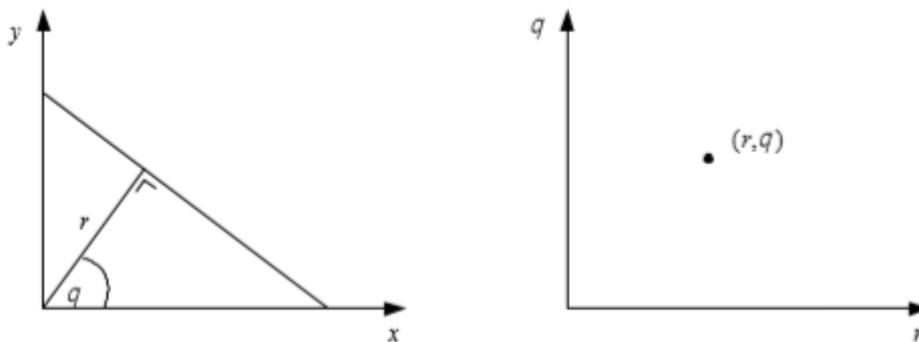
Tingkat ketelitian dari Transformasi Hough bergantung pada ukuran matriks $P(m, c)$, yaitu $K \times L$. Kompleksitas komputasi Transformasi Hough pada kasus terburuk

adalah $O(Kn)$, yang dalam hal ini K adalah jumlah pembagian parameter m , dan n adalah jumlah *pixel* di dalam citra tepi. Karena $O(Kn) < O(n^3)$, maka pendeteksian garis lurus dengan Transformasi Hough lebih cepat daripada metode exhaustive search [5].

Model parametrik pada persamaan 2.2 tidak dapat digunakan untuk mendeteksi garis vertikal atau hampir vertikal karena gradiennya (m) menuju nilai takberhingga. Karena itu, garis dinyatakan dalam representasi polar:

$$r = x \cos q + y \sin q \quad (2.5)$$

yang dalam hal ini r adalah jarak garis ke titik asal (Gambar II.14).



Gambar II.14 Representasi Polar Dari Garis Lurus

Sembarang garis yang melalui (x_1, y_1) pada ruang x - y berkoresponden dengan kurva sinusoida $r = x_1 \cos q + y_1 \sin q$ pada ruang r - q . *Pixel-pixel* yang terletak segaris pada citra tepi berkoresponden dengan titik potong seluruh kurva sinusoidanya pada ruang parameter r - q [5].

2.3 Karakteristik Video Digital

Video digital pada dasarnya tersusun atas serangkaian frame. Rangkaian frame tersebut ditampilkan pada layar dengan kecepatan tertentu, bergantung pada laju frame yang diberikan (dalam frame/detik). Jika laju frame cukup tinggi, mata manusia tidak dapat menangkap gambar per frame, melainkan menangkapnya sebagai rangkaian yang kontinyu. Masing-masing frame merupakan gambar/citra

(image) digital. Suatu image digital direpresentasikan dengan sebuah matriks yang masing-masing elemennya merepresentasikan nilai intensitas. Jika I adalah matriks dua dimensi, $I(x,y)$ adalah nilai intensitas yang sesuai pada posisi baris x dan kolom y pada matriks tersebut. Titik-titik dimana image disampling disebut sebagai picture elements atau sering dikenal sebagai pixel. Karakteristik video digital ditentukan oleh resolusi (resolution) atau dimensi frame (frame dimension), kedalaman pixel (pixel depth), dan laju frame (frame rate). Karakteristik-karakteristik ini yang akan menentukan kualitas video dan jumlah bit yang dibutuhkan untuk menyimpan atau mentransmisikannya [6].

2.3.1 Resolusi

Resolusi (resolution) atau dimensi frame (frame dimension) adalah ukuran sebuah frame pada video digital. Resolusi dinyatakan dalam pixel x pixel. Semakin tinggi resolusi, semakin baik kualitas video tersebut, dalam arti bahwa dalam ukuran fisik yang sama, video dengan resolusi tinggi akan lebih detail. Namun, resolusi yang tinggi akan mengakibatkan jumlah bit yang diperlukan untuk menyimpan atau mentransmisikannya meningkat [6].

2.3.2 Kedalaman Bit

Kedalaman bit (bit depth) menentukan jumlah bit yang digunakan untuk merepresentasikan tiap pixel pada sebuah frame. Kedalaman bit dinyatakan dalam bit per pixel. Semakin banyak jumlah bit yang digunakan untuk merepresentasikan sebuah pixel, berarti semakin tinggi kedalaman pixelnya, maka semakin tinggi pula kualitasnya, mengakibatkan jumlah bit yang diperlukan menjadi lebih tinggi. Satu byte (8 bit) untuk tiap pixel, diperoleh 28 atau 256 level intensitas. Dengan level intensitas sebanyak itu, umumnya mata manusia sudah dapat dipuaskan. Ke dalam pixel paling rendah terdapat pada binary-value image yang hanya menggunakan 1 bit untuk tiap pixel, sehingga hanya ada dua kemungkinan bagi tiap pixel, yaitu 0 (hitam) atau 1 (putih) [6].

2.3.3 Laju Frame

Laju frame (Frame rate) menunjukkan jumlah frame yang digambar tiap detik, dan dinyatakan dengan frame per detik. Sehubungan dengan laju frame ini, ada dua hal yang perlu diperhatikan, yaitu kehalusan gerakan (smooth motion) dan kilatan (flash). Kehalusan gerakan ditentukan oleh jumlah frame yang berbeda per detik. Untuk mendapatkan gerakan yang halus, video digital setidaknya harus menampilkan sedikitnya 25 frame per detik. Kilatan ditentukan oleh jumlah berapa kali layar digambar per detik. Dengan 20 frame per detik, kilatan sudah dapat dilenyapkan. Video yang berkualitas baik akan memiliki laju frame yang tinggi, minimal 30 frame/detik, setidaknya sesuai dengan mata manusia, yang berarti membutuhkan jumlah bit yang lebih tinggi [6].

2.4 Perangkat Lunak

2.4.1 Python 3.7

Python adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Hal ini membuat Python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai bahasa pemrograman lain [7].

Bahasa ini muncul pertama kali pada tahun 1991, dirancang oleh seorang bernama Guido van Rossum. Sampai saat ini Python masih dikembangkan oleh Python Software Foundation. Bahasa Python mendukung hampir semua sistem operasi, bahkan untuk sistem operasi Linux, hampir semua distronya sudah menyertakan Python di dalamnya [7].

Dengan kode yang simpel dan mudah diimplementasikan, seorang programmer dapat lebih mengutamakan pengembangan aplikasi yang dibuat, bukan malah sibuk mencari syntax error [7].

2.4.2 PyCharm

PyCharm adalah Lingkungan pengembangan terintegrasi (IDE) memfasilitasi pemrogram komputer dengan mengintegrasikan alat-alat dasar (mis., Editor kode,

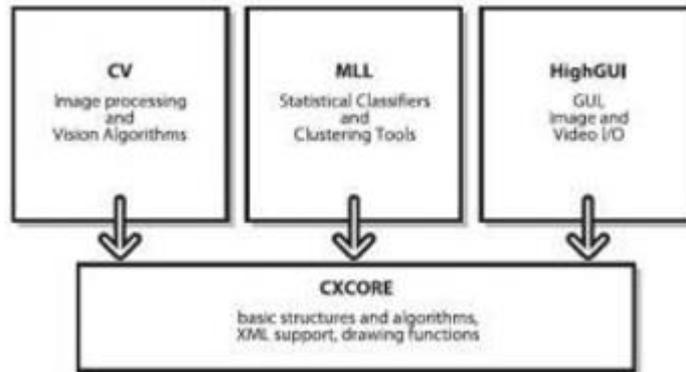
kompiler, dan debugger) ke dalam satu paket perangkat lunak. Pengguna tidak perlu menginstal kompiler/juru bahasa pada mesin mereka; sebuah IDE menyediakan lingkungan itu sendiri [8].

2.4.3 OpenCV

OpenCV (Open Source Computer Vision Library), adalah sebuah library open source yang dikembangkan oleh intel yang fokus untuk menyederhanakan programing terkait citra digital. Di dalam OpenCV sudah mempunyai banyak fitur, antara lain: pengenalan wajah, pelacakan wajah, deteksi wajah, Kalman filtering, dan berbagai jenis metode AI (Artificial Intellegence). Dan menyediakan berbagai algoritma sederhana terkait Computer Vision untuk low level API [9].

OpenCV merupakan open source computer vision library untuk bahasa pemrograman C/C++, dan telah dikembangkan ke phyton, java, matlab. Library dapat diunduh melalui link <http://sourceforge.net/projects/opencvlibrary> atau melalui <http://opencv.org/releases.html>. Saat ini versi yang terbaru adalah 3.3.0 dengan tanggal rilis 3 Agustus 2017 [9].

Intel meluncurkan versi pertama dari OpenCV pada 1999, dan awalnya memerlukan library dari Intel Image Processing Library. Kemudian dependecy tersebut akhirnya dihilangkan sehingga terciptalah OpenCV seperti yang sekarang sebagai standalone library. OpenCV mendukung multiplatform, dapat mendukung baik windows ataupun linux, dan sekarang telah mendukung MacOSX dan android [9]. Berikut adalah struktur dan Konten OpenCV, dapat dilihat pada Gambar II.15.

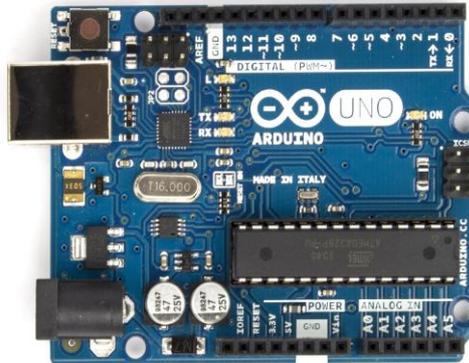


Gambar II.15 Struktur dan Konten OpenCV

2.5 Perangkat Keras

2.5.1 Arduino Uno

Arduino Uno adalah board mikrokontroler berbasis ATmega328 (datasheet). Memiliki 14 pin input dari output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM dan 6 pin input analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang-ke adaptor-DC atau baterai untuk menjalankannya. Uno berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter USB-to-serial berbeda dengan board sebelumnya yang menggunakan chip FTDI driver USB-to-serial. Nama “Uno” berarti satu dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0. Uno dan versi 1.0 akan menjadi versi referensi dari Arduino. Uno adalah yang terbaru dalam serangkaian board USB Arduino, dan sebagai model referensi untuk platform Arduino, untuk perbandingan dengan versi sebelumnya, lihat indeks board Arduino. Berikut adalah perangkat Arduino uno, dapat dilihat pada Gambar II.6 [10].



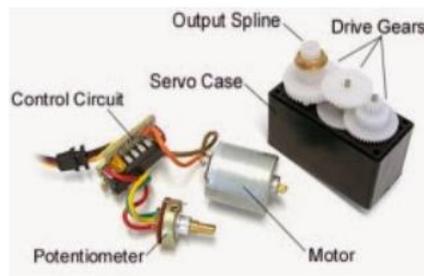
Gambar II.16 Arduino Uno

2.5.2 Motor Servo

Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik loop tertutup (servo), sehingga dapat di set-up atau di atur untuk menentukan dan memastikan posisi sudut dari poros output motor. motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian gear, rangkaian kontrol dan potensiometer. Serangkaian gear yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo [11].

Penggunaan sistem kontrol loop tertutup pada motor servo berguna untuk mengontrol gerakan dan posisi akhir dari poros motor servo. Penjelasan sederhananya begini, posisi poros output akan di sensor untuk mengetahui posisi poros sudah tepat seperti yang di inginkan atau belum, dan jika belum, maka kontrol input akan mengirim sinyal kendali untuk membuat posisi poros tersebut tepat pada posisi yang diinginkan. Untuk lebih jelasnya mengenai sistem kontrol loop tertutup, perhatikan contoh sederhana beberapa aplikasi lain dari sistem kontrol loop tertutup, seperti penyetelan suhu pada AC, kulkas, setrika dan lain sebagainya [11].

Motor servo biasa digunakan dalam aplikasi-aplikasi di industri, selain itu juga digunakan dalam berbagai aplikasi lain seperti pada mobil mainan radio kontrol, robot, pesawat, dan lain sebagainya. Berikut adalah perangkat motor servo, dapat dilihat pada Gambar II.7 [11].

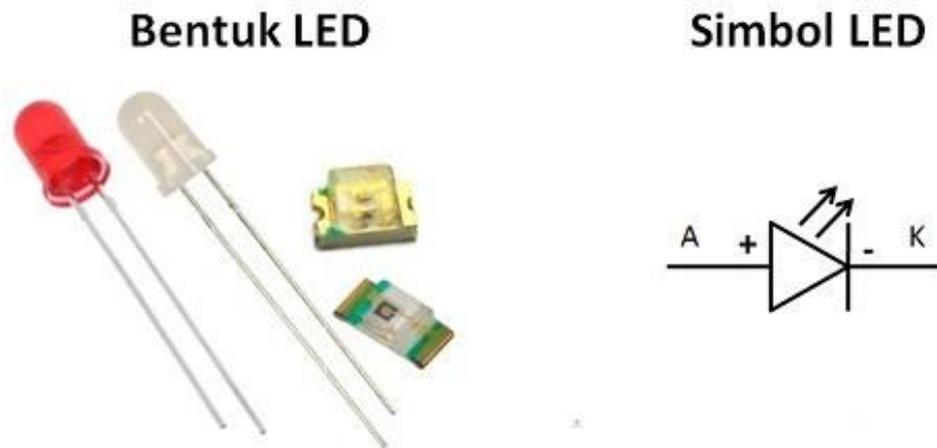


Gambar II.17 Motor Servo

2.5.3 LED

Light Emitting Diode atau sering disingkat dengan LED adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna-warna Cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya. LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering kita jumpai pada Remote Control TV ataupun Remote Control perangkat elektronik lainnya [12].

Bentuk LED mirip dengan sebuah bohlam (bola lampu) yang kecil dan dapat dipasangkan dengan mudah ke dalam berbagai perangkat elektronika. Berbeda dengan Lampu Pijar, LED tidak memerlukan pembakaran filamen sehingga tidak menimbulkan panas dalam menghasilkan cahaya. Berikut adalah bentuk dan simbol LED, dapat dilihat pada Gambar II.18 [12].



Gambar II.18 Bentuk dan Simbol LED

2.6 Persamaan Rumus Pengujian Alat

Pada pengujian alat, tentunya pengujian tersebut memiliki persamaan rumus yang dapat menghasilkan kesimpulan. Berikut beberapa persamaan yang digunakan pada pengujian penelitian ini.

1. Untuk mencari persentase keberhasilan alat pada pengujian estimasi tikungan dapat menggunakan persamaan:

$$Kesesuaian = \frac{\text{Data sesuai}}{\text{Jumlah data yang tampil}} \times 100\% \quad (2.6)$$

2. Untuk mencari persentase kesesuaian nilai sudut yang dikirim dan yang direspon alat dapat menggunakan persamaan:

$$Kesesuaian = \frac{\text{Jumlah data nilai masukan} - \text{Jumlah data yang error}}{\text{Jumlah data nilai masukan}} \times 100\% \quad (2.7)$$

3. Untuk mencari ketinggian rata-rata error sudut pada alat dapat menggunakan persamaan:

$$\text{Rata - rata error alat} = \frac{\text{Jumlah nilai error}}{\text{jumlah data error}} \quad (2.8)$$