

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Beberapa penelitian yang berhubungan dengan Interaksi Manusia-Robot (*Human – Robot Interaction*) dan *Graphical User Interface* (GUI) diantaranya:

Interaksi manusia-robot memungkinkan pertukaran informasi antara pengguna manusia dengan berbagai jenis mesin, komputer dan perangkat elektronik lainnya. Interaksi dengan perangkat modern ini paling sering dilakukan melalui visual, auditori, dan taktil antar muka pengguna (Jaka Sodnik dan Saso Tomazic, 2015)

Pengujian *software Graphical User Interface* (GUI) merupakan salah satu tantangan utama dalam siklus hidup dari setiap *software*. Pengujian GUI secara inheren lebih sulit daripada pengujian antarmuka tradisional (*remote control*) dan *command line*. Beberapa faktor yang membuat pengujian GUI berbeda dengan pengujian *software* tradisional yaitu: jumlah besar objek, tampilan dan nuansa objek yang berbeda, disetiap objek banyak parameter terkait, pengungkapan progresif, input yang kompleks dari berbagai sumber, dan *output* grafis. Teknik pengujian yang sudah ada perlu adanya penyesuaian dan peningkatan untuk GUI, dan Teknik pengujian yang baru diharapkan dapat menciptakan sesuatu yang lebih mudah dan efektif (Zafar, Singhera, Ellis Horowitz, dan Abad Shah, 2009).

*Graphical User Interface* (GUI) adalah bagian penting dari setiap aplikasi perangkat lunak bagi pengguna pada saat ini dan GUI juga harus memenuhi kriteria berikut: *significant design, development, dan testing activities* (Zafar Singhera, Ellis Horowitz dan Abad Shah, 2009).

*Graphical User Interface* (GUI) menyajikan sesuatu hal yang lebih mudah dalam penggunaan fungsi aplikasi, yaitu dengan cara mengorganisir kedalam bentuk pilihan dan hanya menyajikan sesuatu hal yang mudah diterima akal. Dalam implementasinya, GUI memiliki beberapa sifat yaitu: menggunakan modus *asynchronous* dalam pengoperasiannya, menggunakan *input* dan *output* tradisional, mempunyai struktur yang mudah untuk berinteraksi dengan pengguna sehingga menghasilkan pengujian yang berbeda secara signifikan dan lebih sulit dari

pengujian *software* tradisional (Zafar Singhera, Ellis Horowitz, dan Abad Shah, 2009).

Mustar, Muhammad, Dkk (2014) yang berjudul “Perancangan Model Interaksi Manusia dan Robot Dalam Bentuk Tampilan Visual Pada Komputer”, dalam penelitian tersebut dijelaskan model interaksi manusia dengan robot dengan metode tampilan visual pada komputer, robot yang dikendalikan yaitu robot lengan (*manipulator*) dengan 6 derajat kebebasan (*degree of freedom*). Setiap gerakan oleh robot menggunakan motor servo sebagai actuator. Tampilan visual pada perangkat komputer menggunakan *Graphical User Interface* (GUI) yang dibuat dengan menggunakan *software* LabVIEW sebagai pemrosesan data sekaligus sebagai pengontrolan gerak robot.

## **2.2 Landasan Teori**

### **2.2.1 *Graphical User Interface* (GUI)**

Metode antar muka antara manusia dengan computer (*user interface*) merupakan metode yang memungkinkan pengguna dapat berinteraksi dengan sebuah perangkat. Antar muka (*interface*) merupakan tempat dimana pengguna dapat berinteraksi dengan komputer.

Metode antar muka dapat digolongkan menjadi dua metode, yaitu:

1. *Command Line Interface* (CLI), yaitu metode antar muka yang memungkinkan pengguna berinteraksi dengan sistem melalui text-terminal. Pengguna memberikan perintah dengan cara mengetikan baris-baris tertentu.
2. *Graphical User Interface* (GUI), yaitu sebuah sistem yang dirancang agar pengguna (*user*) dapat berinteraksi dengan sebuah perangkat komputer yang digunakan oleh pengguna tersebut. *Graphical User Interface* (GUI) dapat dikontrol dengan berbagai macam inputan seperti *mouse (pointer)*, *touchscreen*, *keyboard*, dan lainnya. Pada sistem operasi *Graphical User Interface* (GUI) terdapat menu tombol, jendela, dan berbagai icon yang dirancang agar pengguna dapat mengoperasikan sistem tersebut dengan mudah. Pengguna hanya cukup memilih ikon, tombol, gambar, atau grafik yang sudah ada pada sistem tersebut

dikarenakan *Graphical User Interface* (GUI) telah dirancang untuk merubah perintah menjadi ikon dalam layer computer yang dapat dipilih untuk menjalankan perintah.

Pada awalnya, saat lahirnya komputer, komputer dapat dioperasikan dengan memberikan perintah-perintah yang diinputkan menggunakan *keyboard* ke layar monitor. Metode dengan cara mengetikkan perintah-perintah teks tertentu pada komputer ini disebut *Command Line Interface* (CLI). Metode *Command Line Interface* ini memiliki banyak kekurangan dikarenakan metode ini dirasa kurang efisien dan kurang canggih. Seiring perkembangan teknologi dan pengetahuan *interface*, maka terciptalah media interaksi yang mudah, praktis, efektif, dan dapat digunakan setiap orang, yaitu *Graphical User Interface* (GUI). *Graphical User Interface* (GUI) memudahkan penggunaanya dalam mengoperasikan perangkat tanpa harus mengingat setiap perintah, pengguna hanya perlu memilih icon, gambar, grafik, atau tombol yang sudah disediakan pada sistem. *Graphical User Interface* (GUI) menggunakan sifat alamiah manusia yang lebih mudah mengingat gambar daripada tulisan.

**1) Kelebihan dan Kekurangan *Graphical User Interface* (GUI):**

Pada tabel 2.1 disebutkan beberapa kelebihan dan kekurangan yang terdapat pada *Graphical User Interface* (GUI)

**Tabel 2.1** Kelebihan dan Kekurangan *Graphical User Interface* (GUI)

No	Kelebihan	Kekurangan
1	<i>User friendly</i> dan tampilan lebih menarik	Membutuhkan spesifikasi perangkat keras yang lebih tinggi terlebih pada VGA Card
2	Tidak perlu mengingat setiap teks perintah untuk mengoperasikan sistem	Performa lebih lambat
3	Lebih mudah digunakan	Mudah terserang virus

### 2.2.2 Kombinasi Warna Terbaik pada desain *Interface*

Pada saat merancang sebuah desain *Interface* perlu adanya pengetahuan mengenai pemilihan warna tagar dapat menghasilkan kombinasi warna terbaik pada desain *Interface*. Warna sangat berperan serta dalam kenyamanan berinteraksi pengguna dengan antar muka, oleh karenanya perlu diperhatikan aspek psikologi pada pemilihan warna dalam perancangan desain GUI. Tabel dibawah ini menunjukkan kombinasi warna terbaik dan kombinasi warna terburuk dalam aspek psikologi pengguna.

#### 1) Komposisi Kombinasi Warna

Pada tabel 2.2 merupakan komposisi pemilihan kombinasi warna terbaik dalam perancangan desain

**Tabel 2.2** Komposisi Kombinasi Warna Terbaik

No	Background	Garis dan Teks (Normal)	Garis dan Teks (Tebal)
1	Putih	Biru (94%), Hitam (63%), Merah (25%)	Hitam (69%), Biru (63%), Merah (31%)
2	Hitam	Putih (75%), Kuning (63%)	Kuning (69%), Putih (59%), Hijau (25%)
3	Merah	Kuning (75%), Putih (56%), Hitam (44%)	Hitam (50%), Kuning (44%), Putih (44%), Cyan (31%)
4	Hijau	Hitam (100%), Biru (56%), Merah (25%)	Hitam (69%), Merah (63%), Biru (31%)
5	Biru	Putih (81%), Kuning (50%), Cyan (25%)	Kuning (38%), Magenta (31%), Hitam (31%), Cyan (31%), Putih (25%)
6	Cyan	Biru (69%), Hitam (56%), Merah (37%)	Merah (56%), Biru (50%), Hitam (44%), Magenta (25%)
7	Magenta	Hitam (63%), Putih (56%), Biru (44%)	Biru (50%), Hitam (44%), Kuning (25%)

8	Kuning	Merah (63%), Biru (63%), Hitam (56%)	Merah (75%), Biru (63%), Hitam (50%)
---	--------	--------------------------------------	--------------------------------------

Pada tabel 2.3 merupakan komposisi pemilihan kombinasi warna terburuk dalam perancangan desain.

**Tabel 2.3** Komposisi Kombinasi Warna Terburuk

No	Background	Garis dan Teks (Normal)	Garis dan Teks (Tebal)
1	Putih	Kuning (100%), Cyan (94%)	Kuning (94%), Cyan (75%)
2	Hitam	Biru (87%), Merah (44%), Magenta (25%)	Biru (81%), Magenta (31%)
3	Merah	Magenta (81%), Biru (44%), Hijau & Cyan (25%)	Magenta (69%), Biru (50%), Hijau (37%), Cyan (25%)
4	Hijau	Cyan (81%), Magenta (50%), Kuning (37%)	Cyan (81%), Magenta & Kuning (44%)
5	Biru	Hijau (62%), Merah & Hitam (37%)	Hijau (44%), Merah & Hitam (31%)
6	Cyan	Hitam (81%), Kuning (75%), Putih (31%)	Kuning (69%), Hijau (62%), Putih (56%)
7	Magenta	Hijau (75%), Merah (56%), Cyan (44%)	Cyan (81%), Hijau (69%), Merah (44%)
8	Kuning	Putih & Cyan (81%)	Putih (81%), Cyan (56%), Hijau (25%)

Hal-hal yang harus diperhatikan dalam pemilihan kombinasi warna:

1. Menghindari penggunaan kombinasi warna yang dapat melelahkan mata seperti warna cyan, kuning, dan magenta secara bersamaan.
2. Bila pengguna pada usia tua lebih baik menggunakan warna yang kontras.

3. Pemilihan warna hijau dan merah dengan jumlah yang besar dan pada tempat yang berseberangan. Perpaduan terbaik dalam hal ini adalah kuning dan biru.
4. Pada kombinasi warna biru dan hijau secara psikologi dapat mengakibatkan hasil atau kesan yang buruk, lebih baik jika menggunakan warna yang berlawanan yaitu biru dan kuning atau hijau dan merah.
5. Bagi pengguna (*user*) yang memiliki keterbatasan dalam melihat warna, sebaiknya menghindari penggunaan warna tunggal.

### 2.2.3 Sistem Kendali Robot

Saat ini dengan adanya kemajuan teknologi dan ilmu pengetahuan dimana kegiatan dalam kehidupan sehari-hari banyak terbantu dengan adanya sistem kontrol. Sistem kendali dapat mengontrol berbagai peralatan dimana dapat meringankan pekerjaan manusia, dari hal yang sangat sederhana hingga yang sangat kompleks.

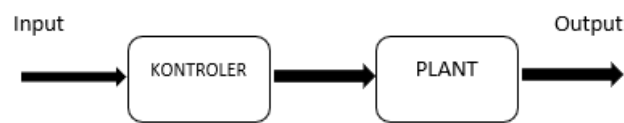
Sistem kendali dapat dikatakan sebagai hubungan antara komponen yang membentuk sebuah konfigurasi sistem, yang akan menghasilkan tanggapan sistem yang diharapkan. Sehingga harus ada yang dikendalikan, yang merupakan suatu sistem fisis, yang biasa disebut dengan kendalian (*plant*) (Rahmat Fauzi Siregar, 2017).

Pada dasarnya sistem kendali terdiri atas beberapa komponen dasar, yaitu berupa masukan, proses pengendalian, dan keluaran. Masukan dan keluaran merupakan *variable* atau besaran *fisis*. Keluaran merupakan hal yang dihasilkan oleh kendalian, artinya yang dikendalikan. Masukan adalah yang mempengaruhi kendalian, yang mengatur keluaran. Kedua dimensi masukan dan keluaran tidak harus sama (Rinda Hedwig, 2004).

Rinda Hedwig (2004) mengatakan bahwa secara umum, tujuan sistem kendali adalah mengendalikan keluaran dari berbagai masukan tertentu melalui unsur-unsur sistem kendali. Secara garis besar, sistem kendali dapat dibagi menjadi dua kategori yaitu sistem kendali terbuka (*open loop system*) dan sistem kendali tertutup (*close loop system*).

**Sistem kendali dibagi menjadi dua, yaitu:**

**1. Sistem kendali terbuka (*Open Loop System*)**, sistem kendali terbuka yaitu sistem kendali yang tidak memperhatikan keluarannya (*output*). Sistem kendali terbuka disebut juga sebagai sistem bukan umpan balik, dikarenakan output tidak mempengaruhi atau efek tindakan *control* sinyal *input*. Sehingga sistem kendali terbuka dapat mengikuti perintah *input* atau *set point* tanpa memperhatikan hasil akhirnya.

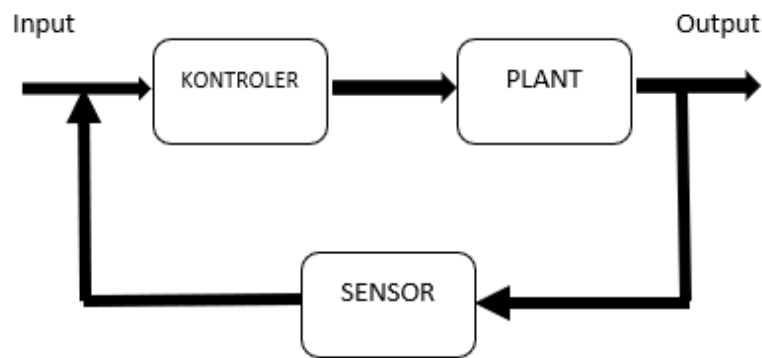


**Gambar 2.1** Diagram sistem kendali terbuka (*Open Loop System*)

**a. Karakteristik sistem kendali terbuka ( *Open Loop System* )**

- 1) Tidak terdapat perbandingan nilai actual dengan nilai yang diinginkan.
- 2) Sistem kendali terbuka tidak memiliki pengaturan tersendiri atau aksi kendali terhadap nilai output.
- 3) Pada pengaturan input menentukan posisi operasi agar tetap sebagai pengendali. Perubahan atau gangguan pada kondisi eksternal tidak akan mengakibatkan perubahan output secara langsung.

**2. Sistem kendali tertutup (*Close Loop System*)**, sistem kendali tertutup yaitu sistem kendali yang sinyal keluarannya mempunyai tidak diperhatikan sehingga tidak ada pengaruh langsung pada aksi pengendalian, sistem kendali tertutup juga merupakan sistem kendali umpan balik. Sinyal kesalahan penggerak, merupakan selisih antara sinyal *input* dan sinyal umpan balik yang berupa sinyal *output* atau suatu fungsi sinyal *output*, diumpankan ke kontroler untuk memperkecil kesalahan dan menghasilkan *output* sistem mendekati nilai yang diinginkan. *Close Loop System* dapat diartikan menggunakan aksi umpan balik untuk memperkecil kesalahan sistem.



**Gambar 2.2** Diagram Sistem kendali tertutup (*Close Loop System*)

**a. Karakteristik sistem kendali tertutup ( *Close Loop System* ) :**

- 1) Menyesuaikan input sistem secara otomatis untuk mengurangi kesalahan.
- 2) Meningkatkan stabilitas sistem yang tidak stabil.
- 3) Meningkatkan dan mengurangi kepekaan sistem.
- 4) Meningkatkan ketahanan terhadap gangguan dari luar saat proses.
- 5) Menghasilkan performa yang handal dan berulang.

**2.2.4 Processing**

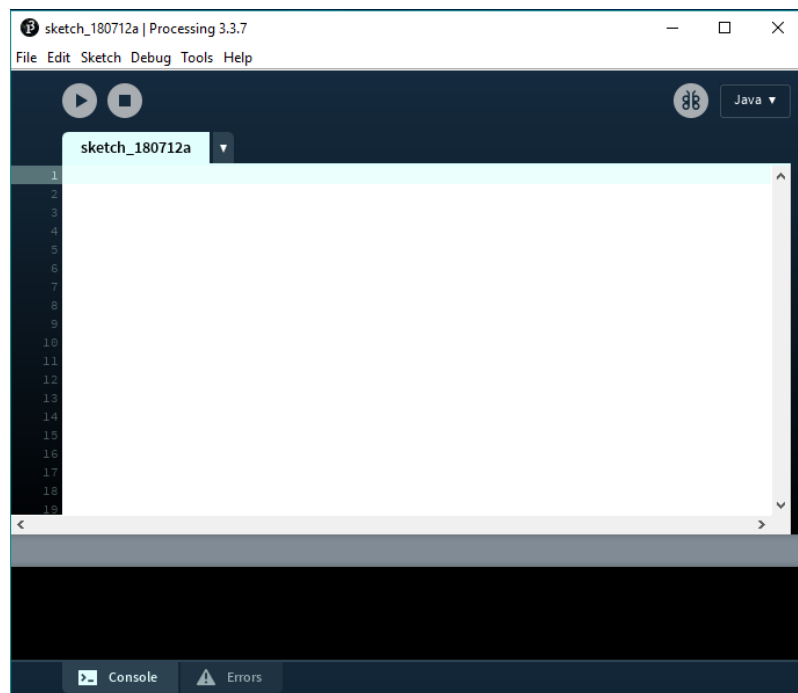
*Processing* adalah nama Bahasa pemrograman yang ditujukan untuk komputer. Bahasa pemrograman ini dirancang oleh Case Reas dan Ben Frey di Massachusetts Institute of Technology (MIT) pada tahun 2001. Tujuannya adalah untuk memudahkan siapa saja yang tidak berlatar belakang ilmu computer dapat menangani hal-hal seperti:

- a. Gambar;
- b. Animasi;
- c. Suara;
- d. Video;
- e. Perangkat keras.

Dengan hanya menggunakan sedikit kode, hasil secara visual dapat diperoleh. Hal ini berbeda dengan bahasa-bahasa lain yang lebih cenderung mengutamakan pemrosesan teks saja. Pada *Processing* dilengkapi juga dengan IDE sehingga dapat dibuat program (yang secara khusus dinamakan sketsa) melalui



*editor* teks yang tersedia pada IDE (Abdul Kadir, 2017). *Processing* merupakan *software* yang dapat digunakan untuk merepresentasikan gambar, animasi, ataupun *interface* peralatan elektronika (Arduino) dengan komputer. Gambar 2.3 memperlihatkan tampilan dari *Processing* IDE. Tipe *Processing* yang digunakan pada penelitian ini adalah *Processing* 3.3.7 merupakan tipe terbaru dari *Processing*.



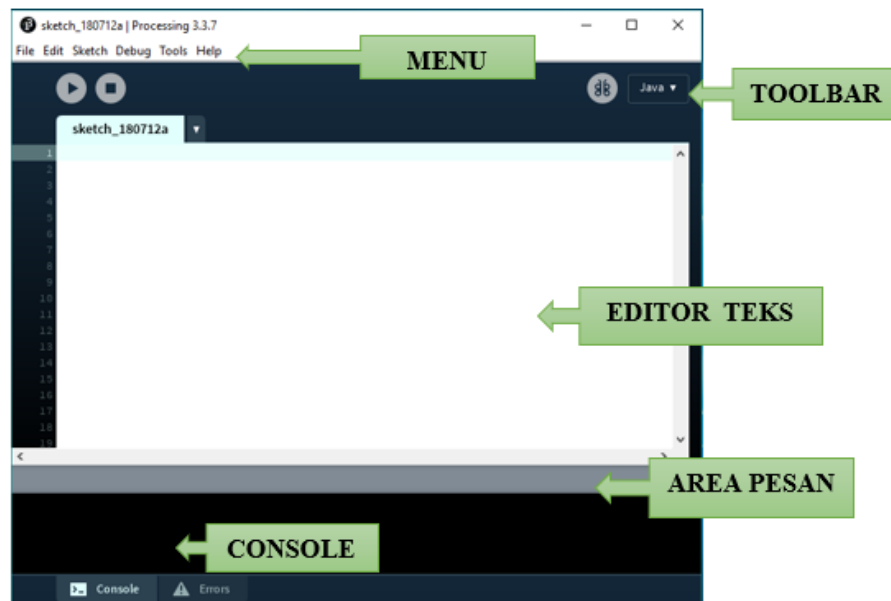
**Gambar 2.3** Processing IDE

Pada mulanya, Bahasa pemrograman *processing* dibuat menjadi *software* sketsa untuk mengajarkan dasar-dasar pemrograman pada komputer dalam kategori visual, *processing* menghadirkan tampilan yang berbeda dan sangat mudah untuk digunakan sehingga banyak mahasiswa, seniman, desainer hingga peneliti menggunakan *Processing* untuk belajar membuat visual pada komputer. *Processing* dibuat juga dengan tujuan mengajarkan dasar-dasar pemrograman komputer untuk memudahkan semua orang belajar desain komputer.

Bahasa *processing* pada implementasinya tidak hanya dibuat dengan bahasa Java, tetapi juga menggunakan sintaks-sintaks yang lebih sederhana sehingga banyak yang lebih memilih menggunakan *software processing* untuk membuat proyek-proyek yang berhubungan dengan sesuatu hal yang sifatnya visual, gerak,

dan interaksi. *Processing* dibuat khusus agar mudah digunakan untuk menghasilkan dan memodifikasi gambar yang sedemikian rupa sehingga untuk para pemula tidak mengalami kesulitan untuk menulis program sendiri. Pada *software* *processing* terdapat banyak pilihan fitur yang mendukung untuk belajar komputer grafik dan interaksi, termasuk juga vector atau raster gambar, model warna, pengolahan citra, jaringan komunikasi, *keyboard*, serta pemrograman berorientasi obyek. *Library* dengan fungsi tambahan dapat di-*download* dan dipelajari dengan mudah. *Library* dengan mudah memperpanjang *Processing* untuk menghasilkan suara, mengirim/menerima data dalam format yang beragam, dan untuk *import/export* dari format file 2D dan 3D (Sulistyo, 2013).

### 1) Bagian-bagian pada *Processing*



**Gambar 2.4** Bagian-bagian pada *Processing*

- a. Menu
- b. Toolbar
- c. Editor Teks
- d. Area Pesan
- e. Console
- f. Run dan Pause

Menu terdiri dari *File, Edit, Sketch, Debug, Tools, Help*. Pada bagian Toolbar terdiri dari *Run, Pause, Variables*, dan Bahasa yang digunakan. Pada bagian Editor teks digunakan untuk menuliskan sketsa. Area pesan digunakan untuk memberikan peringatan apabila terdapat *error*. Pada bagian Console merupakan tempat hasil dari sketsa dan pesan apabila terjadi kesalahan.

Pemrograman *Processing* terdiri dari teks editor terintegrasi dan jendela tampilan untuk menampilkan program. Apabila tombol “*Run*” ditekan, program akan mengkompilasi dan berjalan di jendela grafis (jendela tampilan). Pada bagian jendela utama, dapat menerapkan, mengakhiri, menyimpan, membuka, dan mengekspor file. *Processing* menggunakan sistem koordinat kartesian dengan titik asal terletak pada sudut kiri bagian atas. Contohnya, jika program tersebut berukuran lebar 320 piksel dan 240 piksel maka koordinat [0,0] terletak pada kiri bagian atas dan koordinat [320, 240] terletak di bagian bawah.

## 2) Perintah dasar pada *Processing*

Pada tabel 2.4 merupakan perintah-perintah dasar yang terdapat pada *software Processing*

**Tabel 2.4** Perintah dasar pada *Processing*

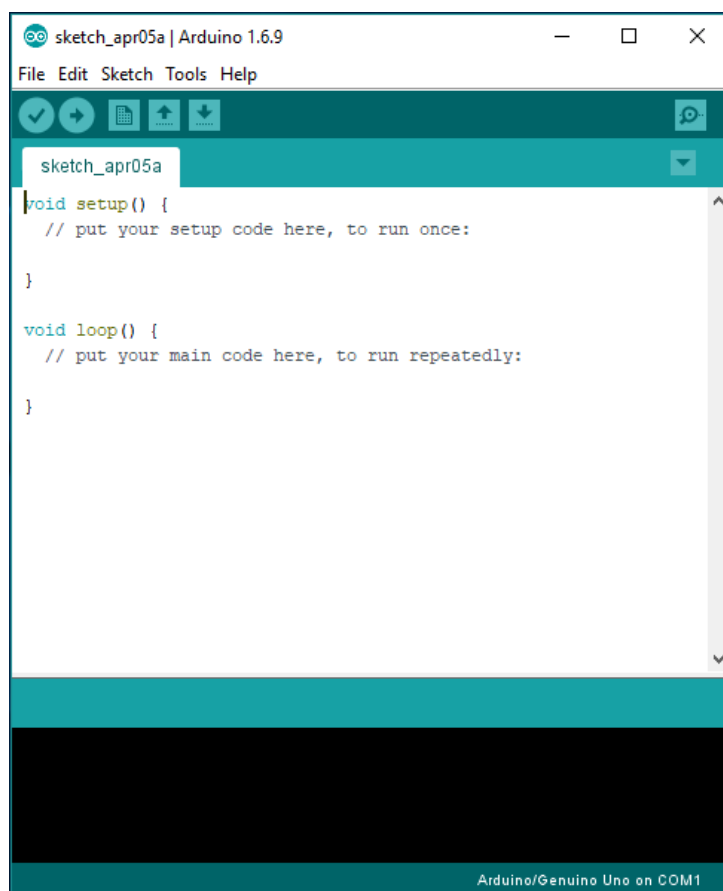
No	Fungsi	Keterangan
1	size()	Digunakan untuk membentuk ukuran jendela
2	setup()	Digunakan untuk melakukan tindakan awal, tindakan awal hanya dilakukan sekali.
No.	Fungsi	Keterangan
3	draw()	Berisi kode yang akan diulang secara terus-menerus. Dalam satu detik, pengulangan dilakukan sebanyak 24 kali.

4	//	Keterangan di sketsa yang khusus ditujukan untuk pembaca memahami kode yang menyusun sketsa. Komentar dapat dimanfaatkan juga untuk mematikan pernyataan tertentu selama pengujian.
5	point()	Digunakan untuk membuat sebuah piksel.
6	line()	Digunakan untuk membuat garis.
7	triangle()	Digunakan untuk membuat bentuk segitiga.
8	quad()	Digunakan untuk membuat bentuk trapesium.
9	ellipse()	Digunakan untuk membuat bentuk elips atau lingkaran.
10	rect()	Digunakan untuk membuat bentuk persegi.
11	arc()	Digunakan untuk membuat busur lingkaran.
12	vertex() dan endShape()	Digunakan untuk membuat bentuk buatan sendiri.
13	Stroke()	Digunakan untuk menentukan warna tepi bentuk.
14	background()	Digunakan untuk mengatur warna latar belakang.
15	fill()	Digunakan untuk mengatur warna latar belakang bentuk.
16	Rotate()	Digunakan untuk memutar gambar.

### 2.2.5 Arduino Uno ATmega 328

Arduino merupakan platform *computing* fisik atau *embedded*, yang berarti sistem interaktif yang dapat berinteraksi dengan lingkungannya melalui penggunaan perangkat keras dan perangkat lunak. Arduino dapat merasakan lingkungan dengan menerima masukan (*input*) dari berbagai sensor, serta dapat mempengaruhi sekitarnya dan mampu mengendalikan sebuah lampu, actuator,

motor, dan lainnya. Mikrokontroler pada papan (*board*) rangkaiannya dapat deprogram menggunakan Bahasa pemrograman Arduino (berdasarkan *Wiring*) dan lingkungan pengembangan Arduino (berdasarkan *Processing*). Arduino dapat berdiri sendiri atau dapat berkomunikasi dengan perangkat lunak yang berjalan pada komputer. Ketika akan meng-*compile*, menulis, dan men-*download* program Arduino dibutuhkan *software* yang bernama Arduino IDE atau lebih dikenal sketch. Gambar 2.5 Memperlihatkan tampilan dari Arduino IDE.



**Gambar 2.5** Arduino IDE

Tipe Arduino yang digunakan pada penelitian ini adalah Arduino Uno ATmega 328, Arduino Uno ATmega 328 yaitu chip mikrokontroler 8-bit berbasis AVR-RISC buatan Atmel. Chip ini memiliki 32 kb memori ISP flash dengan kemampuan baca-tulis (*read write*) 1kb EEPROM, dan 2 kb SRAM. Kapasitas memori flash sebesar 32 kb pada chip, menjadi alasan chip ini diberi nama ATmega 328. Chip lain memiliki memori 8 kb dan diberi nama ATmega 8, begitu juga dengan ATmega 16 untuk chip yang memiliki memori 16 kb.

Chip tersebut memiliki 23 jalur general purpose I/O (*input/output*), 32 buah register, 3 buah *timer/counter* dengan mode perbandingan, *interrupt* internal dan eksternal, serial *programmable* USART, *2-wire interface* serial, serial port SPI, 6 buah channel 10-bit A/D converter, *programmable watchdog timer* dengan oscillator internal, dan lima *power saving mode*. Chip bekerja pada tegangan antara 1.8V ~ 5.5V. *Output* komputasi bisa mencapai 1 MIPS per Mhz. Maksimum *operating frequency* adalah 20 Mhz. ATmega328 menjadi cukup populer setelah chip ini dipergunakan dalam board Arduino. Dengan adanya Arduino yang didukung oleh *software* Arduino IDE, pemrograman chip ATmega328 menjadi jauh lebih sederhana dan mudah.

Adapun Fitur mengenai Arduino Uno ATmega 328 adalah sebagai berikut:

### **Performa Tinggi, Komputer Mikrokontroler 8-Bit Daya Atmel®AVR® Rendah**

#### **1. Arsitektur RISC Lanjutan**

- a. 131 Instruksi yang Kuat
- b. Kebanyakan Eksekusi Siklus Jam Tunggal
- c. 32 x 8 Register Kerja Serba Guna
- d. Operasi Sepenuhnya Statis
- e. Hingga 20 MIPS Throughput di 20MHz
- f. *Multiplier* 2-siklus On-chip

#### **2. Segmen Memori Ketahanan Tinggi Tidak Berubah**

- a. 32KBytes program *In-System Self-Programmable Flash Memory*
- b. 1KBytes EEPROM
- c. 2KBytes SRAM Internal
- d. Tulis/Hapus Siklus: 10.000 Flash / 100.000 EEPROM
- e. Penyimpanan Data: 20 tahun pada 85 ° C / 100 tahun pada 25 C<sup>(1)</sup>
- f. Bagian Kode Boot Opsional dengan Kunci Kunci Independen

#### **3. Pemrograman Dalam Sistem oleh Program Boot On-chip**

#### **4. Bagian Kode Boot Opsional dengan Kunci Kunci Independen**

- a. Kunci Program untuk Keamanan Perangkat Lunak

#### **5. Dukungan Perpustakaan Atmel® QTouch®**

- a. Tombol Sentuh Kapasitif, Penggeser, dan Roda
- b. QTouch dan QMatrix® Akuisisi
- c. Hingga 64 *channels*.

#### **6. Fitur Periferal**

- a. Dua Timer/Penghitung 8-bit dengan Prescaler Terpisah dan Mode bandingkan
- b. Satu Timer/Counter 16-bit dengan Prescaler Terpisah, Mode Bandingkan, dan Mode Tangkap
- c. Konter Real Time dengan Oscillator Terpisah
- d. Enam Saluran PWM
- e. 8-channel 10-bit ADC dalam paket TQFP dan QFN / MLF

#### **7. Pengukuran suhu**

- a. 6-channel ADC 10-bit dalam Paket PDIP

#### **8. Pengukuran suhu**

- a. Dua Antarmuka Serial Master / Slave SPI
- b. Satu Programmable USART Serial
- c. Satu Antarmuka Serial 2-kawat berorientasi-byte (kompatibel dengan Philips I2C)
- d. Pengatur Waktu Pengawas yang Dapat Diprogram dengan Osilator Terpisah On-chip
- e. Satu *On-chip Analog Comparator*
- f. Mengganggu dan Bangun di Pin Change

#### **9. Fitur Mikrokontroler Khusus**

- a. Penyetelan Ulang Daya dan Deteksi *Brown-out* yang Dapat Diprogram
- b. Osilator Internal Kalibrasi
- c. Sumber Interupsi Eksternal dan Internal
- d. Enam Mode Tidur: Menganggur, *ADC Noise Reduction*, *Power-save*, *Power-down*, *Standby*, dan Siaga diperpanjang

#### **10. I/O dan Paket**

- a. 23 Garis I / O Dapat Diprogram
- b. PDIP 28-pin, TQFP 32-lead, 28-pad QFN / MLF dan 32-pad QFN/ MLF

### 11. Tegangan Operasi:

- a. 1,8 - 5,5V

### 12. Kisaran Suhu

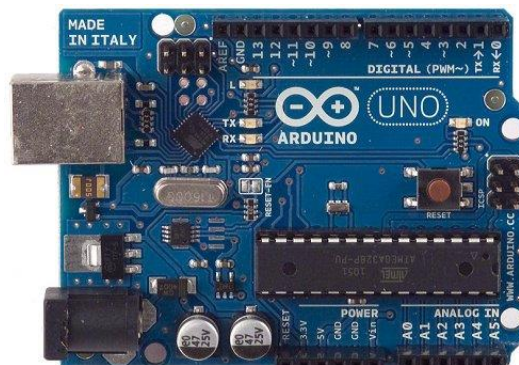
- a. -40 ° C hingga 105 ° C

### 13. Kecepatan Kelas

- a. 0 - 4MHz @ 1,8-5,5V
- b. 0 - 10MHz @ 2.7 - 5.5V
- c. 0 - 20MHz @ 4.5 - 5.5V

### 14. Konsumsi Daya pada 1MHz, 1.8V, 25 ° C

- a. Mode Aktif: 0.2mA
- b. Mode Power-down: 0.1 $\mu$ A
- c. Mode Hemat Daya: 0.75 $\mu$ A (Termasuk 32kHz RTC)



**Gambar 2.6** Arduino Uno ATmega 328

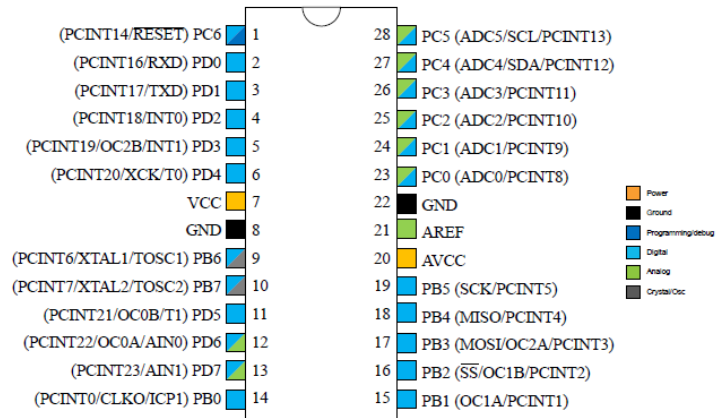
(Source: <http://www.hobbytronics.co.uk/arduino-uno>)

### 1) Pin Configuration

Setiap pin mikrokontroler ATmega 328 dipetakan sesuai dengan kebutuhan standar Arduino pada umumnya. Pemetaan pin (*pin mapping*) ATmega 328 dapat dilihat pada gambar 2.7



**a. Pin-out**



**Gambar 2.7** Figure 5-1. 28-pin PDIP

**b. Pin Descriptions**

**Tabel 2.5** Pin Descriptions

Pada tabel berikut ini merupakan deskripsi Pin yang terdapat pada Arduino Uno ATmega 328

No	Nama Pin	Fungsi
1	Pin 0 dan 1	0 sebagai RX dan 1 sebagai TX, kedua pin ini dapat berfungsi sebagai pengirim dan penerima data serial. Pin ini terhubung dengan pin ATmega USB to serial.
2	Pin 2 dan 3	Pin interupsi external dapat diaktifkan dengan perintah attachInterrupt()
No	Nama Pin	Fungsi
3	Pin 3, 5, 6, 9, 10	Pin digital dan juga sebagai pin yang menyediakan fungsi PWM

4	SPI : pin 10 (SS), 11 (MOSI), 12 (MISO), 13(SCK)	Digunakan sebagai komunikasi SPI (Serial Peripheral Interface)
5	Pin 13	Sebagai pin I/O digital sekaligus sebagai LED <i>built-in</i> sehingga LED akan menyala ketika pin 13 diaktifkan.
6	Pin A0-A5	Sebagai PIN analog yang semuanya mensupport resolusi 10 bit.
7	Pin A4 dan A5	Digunakan sebagai komunikasi TWI (Two Wire Interface) A4 sebagai SDA (Serial Data) dan pin A5 sebagai SCL (Serial Clock).

### c. Memori Program

Pada Arduino Uno terdapat memori program dimana memori program ini digunakan untuk penyimpanan data dan program.

Terdapat tiga jenis memori yang terdapat pada Arduino, yaitu:

1. Flash Memory
2. SRAM (*Static Random Access Memory*)
3. EEPROM (*Electrically Erasable Programmable Read Only Memory*)

*Flash Memory* dan EEPROM termasuk dalam kategori memori *non-volatile*, sehingga apabila Arduino kehilangan catu daya, data dan program yang ada pada kedua memori tersebut tidak akan hilang. Sedangkan SRAM termasuk dalam kategori memori *volatile*, sehingga apabila Arduino kehilangan catu daya, data yang tersimpan akan hilang.

Arduino uno memiliki Flash Memory 32 KB dimana 0,5 KB digunakan sebagai *bootloader*. *Bootloader* adalah suatu program kecil yang tertanam pada suatu sistem operasi. *Bootloader* digunakan untuk memanggil sistem operasi saat sistem pertama kali dinyalakan. *Bootloader* akan menerima file hasil kompilasi yang telah di

*upload* ke Arduino dan akan menyimpannya ke memori program kemudian Arduino akan langsung mengeksekusi program tersebut.

SRAM (*Static Random Access Memory*) yaitu memori untuk menyimpan data-data variabel sementara. Ukuran SRAM di Arduino Uno hanya 2 KB. Oleh karena itu, pada saat deklarasi variabel, menggunakan jenis tipe data sesuai dengan kapasitasnya.

EEPROM (*Electrically Erasable Programmable Read Only Memory*) memiliki ukuran penyimpanan sebesar 1 KB.

### **2.2.6 Catu Daya**

Arduino dapat diberikan catu daya melalui koneksi USB atau catu daya dari luar non-USB seperti Adaptor to DC dan baterai. Jangkauan tegangan yang dapat disuplai ke Arduino sebesar 4-20 Volt. Namun tegangan yang direkomendasikan yaitu dari 7-12 Volt.

### **2.2.7 Komunikasi Serial**

Komunikasi serial merupakan metode pengiriman data berurut bit demi bit melalui sebuah saluran transmisi. Komunikasi serial memiliki kelebihan seperti, transmisi data yang dilakukan hanya dengan melalui sebuah saluran sehingga mengurangi *cost* yang dikeluarkan untuk kebutuhan saluran transmisi.

### **2.2.8 Motor DC ( *Direct Current* )**

Motor listrik adalah suatu peralatan listrik yang dapat merubah energi listrik menjadi energi mekanik (gerak). Prinsip kerjanya berdasarkan hukum *Lorentz* yang menyatakan bahwa apabila suatu penghantar didalam medan magnet yang dialiri arus listrik maka akan timbul suatu gaya. Medan magnet dalam pengertiannya adalah daerah disekitar magnet yang masih memperoleh pengaruh dari magnet. Medan magnet yang berubah-ubah terhadap waktu akan menimbulkan medan listrik pada suatu kumparan yang akhirnya akan menimbulkan medan listrik pada suatu beda tegangan. Medan magnet sangat berperan dalam proses konversi energi *elektromagnetik*.

Motor listrik digunakan untuk menggerakkan peralatan-peralatan lain. Energi mekanis yang diperoleh motor listrik pada umumnya dalam bentuk putaran. Energi mekanis putar dapat dengan mudah diubah menjadi gerakan linier. Hal ini dikarenakan motor listrik mempunyai prinsip kerja yang sederhana, sehingga perlengkapannya juga sederhana, sehingga perlengkapannya juga sederhana dan memiliki torsi yang cukup besar. Motor listrik mempunyai dua jenis yaitu motor listrik AC (arus bolak-balik) dan motor listrik DC (arus searah).

### **1. Motor DC ( Arus searah )**

(C. T. Leondes, 1972) Motor DC adalah pranti elektronik yang mengubah energi listrik menjadi energi mekanik berupa gerak rotasi. Pada Motor DC terdapat jangkar dengan satu atau lebih kumparan terpisah. Tiap kumparan berujung pada cincin belah (*komutator*). Dengan adanya *insulator* antara *komutator*, cincin belah dapat berperan sebagai saklar kutub ganda (*double pole, double throw switch*). Motor DC bekerja berdasarkan prinsip gaya *Lorentz*, yang menyatakan ketika sebuah konduktor beraliran arus diletakkan dalam medan magnet, maka sebuah gaya (dikenal dengan gaya *Lorentz*) akan tercipta secara *orthogonal* diantara arah medan magnet dan arah aliran arus.

#### **a. Prinsip kerja Motor DC**

Pada Motor DC terdapat dua bagian yaitu *Stator* dan *Rotor*. *Stator* merupakan bagian motor yang diam (*statis*) atau tidak berputar. Pada bagian *statis* ini terdiri dari rangka dan kumparan medan. *Rotor* merupakan bagian yang berputar. Pada bagian *rotor* terdiri dari kumparan jangkar. Kedua bagian utama terbagi lagi menjadi beberapa komponen penting, yaitu *Yoke* (kerangka magnet), *Poles* (kutub motor), *Field winding* (kumparan medan magnet), *Armature Winding* (kumparan jangkar), *Commutator* (komutator), dan *Brushes* (kuas/sikat)

Motor DC pada prinsipnya menggunakan reaksi *elektromagnet* untuk bergerak, ketika kumparan diberi arus listrik, maka permukaan kumparan yang bersifat utara akan bergerak menghadap ke magnet yang berkutub selatan dan kumparan yang bersifat selatan akan bergerak menghadap ke arah utara. Ketika

kumparan kutub utara bertemu dengan kutub selatan atau juga sebaliknya, maka terjadi tarik menarik yang menyebabkan pergerakan kumparan berhenti. Saat kumparan berhenti dan ingin digerakkan kembali, saat kutub kumparan berhadapan dengan kutub magnet, arah arus pada kumparan akan dibalik. Maka, kutub utara kumparan akan berubah menjadi kutub selatan, dan kutub selatan akan merubah menjadi kutub utara. Ketika terjadi perubahan kutub, kutub selatan kumparan akan berhadapan dengan kutub selatan magnet, begitu juga kutub utara kumparan akan berhadapan dengan kutub selatan magnet. Saat kutub yang sama berhadapan-hadapan maka terjadi reaksi tolak menolak sehingga kumparan bergerak memutar hingga kutub utara kumparan berhadapan dengan kutub selatan magnet, dan kutub selatan kumparan berhadapan dengan kutub utara magnet. Pada saat ini, arus yang mengalir ke kumparan dibalik kembali dan kumparan akan berputar kembali karena terjadi perubahan kutub. Hal ini akan terjadi berulang-ulang hingga arus listrik pada kumparan diputus atau dihilangkan.

#### **b. Kecepatan putaran Motor DC**

Kecepatan putar motor dc dapat dirumuskan dengan persamaan :

$$N = \frac{V_{TM} - I_A R_A}{K_{\phi}}$$

Keterangan:

N = kecepatan putar motor dc

V<sub>tm</sub> = Tegangan Terminal

I = Arus jangkar motor

R = Hambatan jangkar motor

K = Konstanta motor

Ø = Fluks magnet

Pengendalian kecepatan putar motor dc dapat dilakukan dengan mengatur besar tegangan terminal motor V<sub>tm</sub>. Metode lain yang bisa digunakan untuk mengendalikan kecepatan motor dc yaitu dengan Teknik modulasi lebar pulsa atau *Pulse Width Modulation* (PWM).

### **c. Pulse Width Modulation (PWM)**

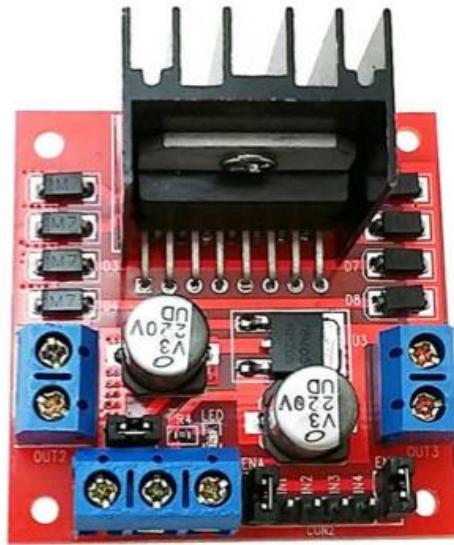
*Pulse Width Modulation* (PWM) merupakan salah satu cara untuk memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu periode, untuk mendapatkan tegangan rata-rata yang berbeda. Contoh implementasi PWM yaitu pemodulasi data untuk telekomunikasi, pengontrolan daya atau tegangan yang masuk ke beban, regulator tegangan, audio *effect*, penguatan, dan lainnya. Implementasi PWM berbasis mikrokontroler biasanya digunakan untuk pengendalian kecepatan Motor DC, pengendalian Motor Servo, pengaturan nyala LED, dan lainnya.

Sinyal PWM biasanya memiliki amplitudo frekuensi dasar yang stabil, tetapi lebar pulsa dapat bervariasi. Lebar pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. Maka disimpulkan bahwa sinyal PWM memiliki frekuensi gelombang yang tetap namun *duty cycle* bervariasi (antara 0% hingga 100%).

#### **2.2.9 Motor Driver L298**

L298 merupakan jenis IC driver motor yang dapat mengendalikan arah putaran dan kecepatan Motor DC ataupun Motor Stepper. Mampu mengeluarkan output tegangan untuk Motor DC dan Motor Stepper sebesar 50 volt.

L298 yaitu sebuah IC tipe H-Bridge yang dapat mengendalikan beban-beban induktif seperti *relay*, *solenoid*, Motor DC, dan Motor Stepper. IC L298 terdiri dari *transistor-transistor logic* (TTL) dengan gerbang nand yang berfungsi untuk memudahkan dalam menentukan arah putaran sebuah motor dc maupun motor stepper. Modul driver Motor L298 memiliki kelebihan yaitu dalam hal ketelitian dalam mengontrol motor sehingga motor mudah untuk dikontrol.



**Gambar 2.8** Driver motor L298

<https://www.nyebarilmu.com/tutorial-arduino-mengakses-driver-motor-l298n/>