BAB II

DASAR TEORI

Pada bab ini akan dijelaskan mengenai studi pustaka tentang robot dan vacuum, teori-teori penunjang sistem baik perangkat keras (*hardware*), maupun perangkat lunak (s*oftware*), serta beberapa teori penunjang lainnya.

2.1 Sejarah Robot

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu (kecerdasan buatan). Istilah robot berasal dari Czech, *robota*, yang berarti bekerja. Istilah itu diperkenalkan ke publik oleh Karel Capek saat mementaskan *Rossum's Universal Robots* (RUR) pada tahun 1921. Meskipun begitu, awal kemunculan robot sebetulnya dapat diurutkan dari bangsa Yunani kuno yang membuat patung yang dapat dipindah-pindahkan. Sekitar 270 BC, Ctesibus, seorang insinyur Yunani, membuat organ dan jam air dengan komponen yang dapat dipindahkan. Al-Jajari (1136-1206), seorang ilmuwan pada dinasti Artuqid, dianggap sebagai tokoh yang pertama kali menciptakan robot humanoid yang berfungsi sebagai 4 musisi. Pada Tahun 1770, Pierre Jacquet Droz, seorang pembuat jam berkebangsaan Swiss, membuat 3 boneka mekanis. Boneka-boneka itu dapat melakukan fungsi spesifik seperti menulis, menggambar, memainkan musik dan organ. Pada tahun 1898 Nikola Tesla membuat sebuah boat yang dikontrol melalui radio *remote control*, dan didemokan di Madison Square Garden. Namun usaha untuk membuat *autonomous boat* tersebut gagal karena masalah dana.

Jepang, pada tahun 1967, yang pada saat itu merupakan Negara yang baru bangkit, mengimpor robot *Versatran* dari AMF. Kejayaan robot dimulai pada tahun 1970 ketika Profesor Victor Scheinman dari Universitas Stanford mendesain lengan standar. Terakhir, pada tahun 2000, Honda memamerkan robot yang dibangun bertahun-tahun lamanya, yang diberi nama ASIMO, yang kemudian disusul oleh Sony dengan robot anjing AIBO-nya.

2.2 Robot *Mobile*

Robot *Mobile* adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain dengan bantuan navigasi dari sebuah sensor. Ada banyak navigasi yang digunakan robot *mobile*, seperti yang akan dijelaskan dibawah ini.

2.2.1 Manual jauh atau *tele-op*

Sebuah robot manual *tele-op* benar-benar di bawah kendali seorang sopir dengan joystick atau perangkat kontrol lainnya. Perangkat mungkin dihubungkan langsung ke robot, mungkin joystick nirkabel, atau mungkin menjadi aksesori ke komputer nirkabel atau pengendali lainnya. Sebuah robot tele-op biasanya digunakan untuk menjaga operator dari bahaya. Contoh, robot remote manual termasuk Robotika Design ANATROLLER ISPA-100, ISPA-50, Foster-Miller Talon, iRobot's PackBot, dan KumoTek's MK-705 Roosterbot.

2.2.2 Guarded tele-op

Sebuah robot *guarded tele-op* memiliki kemampuan untuk merasakan dan menghindari rintangan tetapi sebaliknya akan menavigasi sebagai penggerak,

seperti robot di bawah manual *tele-op*. Jika ada beberapa robot *mobile* hanya menawarkan *guarded tele-op Sliding*.

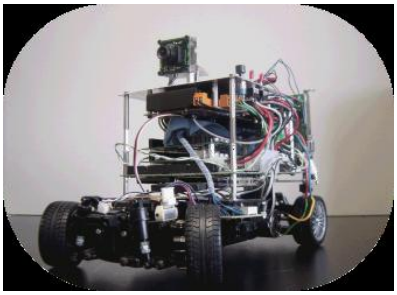
2.2.3 Next-line

Beberapa *Automated* awal Dipandu Kendaraan (*AGVs*) adalah baris berikut *mobile robot*. Mereka mungkin mengikuti garis visual dicat atau tertanam di lantai atau langit-langit atau sebuah kabel listrik di lantai. Sebagian besar robot dioperasikan sederhana "menjaga garis di pusat sensor" algoritma. Mereka tidak bisa mengelilingi hambatan, mereka hanya berhenti dan menunggu ketika sesuatu menghalangi jalan mereka. Banyak contoh dari kendaraan tersebut masih dijual, oleh Transbotics , FMC, Egemin, HK Systems dan perusahaan lainnya.

2.2.4 Robot Otonom acak

Otonomi robot dengan gerakan acak pada dasarnya terpental ke dinding, baik dinding-dinding yang merasakan dengan bumper fisik seperti pembersih Roomba atau dengan sensor elektronik seperti mesin pemotong rumput robotika friendly. Algoritma sederhana bump dan putar 30° akhirnya mengarah ke jangkauan sebagian besar atau seluruh permukaan lantai atau halaman.

Robot *mobile* ini sangat disukai bagi orang yang mulai mempelajari robot. Hal ini karena membuat robot *mobile* tidak memerlukan kerja fisik yang berat. Untuk dapat membuat sebuah robot *mobile* minimal diperlukan pengetahuan tentang mikrokontroler dan sensor-sensor elektronik.



Gambar 2. 1 Robot *Mobile*

Base robot *mobil*e dapat dengan mudah dibuat dengan menggunakan *plywood* /triplek, akrilik sampai menggunakan logam (aluminium). Robot mobil dapat dibuat sebagai pengikut garis (*Line Follower*) atau pengikut dinding (*Wall Follower*) ataupun pengikut cahaya.

2.3 Sejarah Robot Vacuum Cleaner

Robot pembersih pertama kali yang diproduksi adalah *Electrolux Trilobite* oleh Swedia rumah tangga dan peralatan profesional produsen, *Electrolux*. Pada tahun 1997, salah satu versi pertama *Electrolux* dari *Trilobite vacuum* ditampilkan pada program sains BBC, *Tomorrow's World*.



Gambar 2. 2*Electrolux* dari *Trilobite vacuum*

Pada tahun 2001, perusahaan teknologi Inggris, Dyson (perusahaan) membangun dan menunjukkan robot *vacuum* yang dikenal sebagai DC06. Namun, karena harga tinggi, itu tidak pernah dirilis ke pasar.



Gambar 2. 3Robot *Vacuum DC06*

Pada tahun 2002, perusahaan teknologi canggih Amerika, *iRobot*, meluncurkan *Roomba* robot pembersih lantai. Awalnya, *iRobot* memutuskan untuk memproduksi 15.000 unit dan 10.000 unit lebih tergantung pada keberhasilan peluncuran. *Roomba* segera menjadi sensasi konsumen yang besar. Sejak tahun 2002, variasi baru dari robot *vacuum* telah muncul di pasar. Sebagai contoh, *Canadian bObsweep robotic vacuum* yang mempunyai pel dan *vacuum*,atau *Neato Robotika XV-11 robotic vacuum*, yang menggunakan *laser-vision* daripada model sensor *ultrasound* tradisional yang didasarkan. Pada tahun 2014, Dyson mengumumkan comeback-nya ke industri vakum robot dengan *Dyson 360 Eye*. 360 mengacu pada kamera 360^0 yang dipasang pada robot dan seharusnya memberikan orientasi yang lebih baik saat membersihkan.



Gambar 2. 4*Dyson 360 Eye*

Ini adalah pendekatan yang berbeda untuk navigasi dan jalur temuan dibandingkan dengan produk dari *Neato Robotika* atau *iRobot*. Robot *vacuum* dijadwalkan hanya rilis di Jepang pada musim semi 2015, tetapi *International Launched* diharapkan untuk mengikuti akhir tahun ini.

2.4 Mikrokontroler

Mikrokontroler adalah sebuah sistem *microprocessor* dimana di dalamnya sudah terdapat CPU, ROM, RAM dan I/O, *clock* dan peralatan internal lainnya yang sudah saling terhubung dan terorganisasi dengan baik oleh pabrik pembuatnya dan dikemas dalam satu *chip* yang siap pakai. Sehingga kita tinggal memprogram isi ROM sesuai aturan penggunaan oleh pabrik yang membuatnya.

Terdapat dua jenis mikrokontroler yang berkembang saat ini dan masing-masing mempunyai keturunan/keluarga sendiri Hal ini didasarkan pada arsitektur prosesor dan set intruksinya.

1. *Reduced Instruction Set Computer (*RISC) : instruksi terbatas tapi memiliki fasilitas yang lebih banyak.
2. *Complex Instruction Set Computer (CISC*) : instruksi bisa dikatakan lebih lengkap tapi dengan fasilitas secukupnya.

Arduino mega 2560 merupakan sebuah board mikrokontroler berbasis Atmega2560. Modul ini memiliki 54 *digital input/output* dimana 14 digunakan untuk *PWM output* dan 16 digunakan sebagai *analog input*, 4 port serial, 16 MHz osilator Kristal, koneksi USB, *power jack*, *ICISP Header,* dan tombol *reset.* Memiliki *flash memory* sebesar 256Kb sangat cukup untuk menampung program yang banyak.

Arduino mega 2560 tidak memerlukan *flash* program *external* karena di dalam chip mikrokontroler Arduino telah diisi dengan *bootloader* yang membuat proses *upload* program yang kita buat menjadi lebih sederhana dan cepat. Untuk koneksi dengan computer sudah tersedia RS232 to TTL *converter* atau menggunakan chip USB ke *serial converter* seperti FTDI FTD232.

2.5 Aktuator

Aktuator adalah elemen yang berfungsi mengkonversi energy, dari energi listrik ke energi mekanik atau energi yang terlihat. Bentuk konkrit aktuator ini misalnya: motor listrik, solenoid, relay, display, dll.

2.5.1. Motor DC

Motor DC merupakan peralatan elektromekanik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik. Secara umum, kecepatan putaran poros motor DC akan meningkat seiring dengan meningkatnya tegangan yang diberikan. Dengan demikian, putaran motor DC akan berbalik arah jika polaritas tegangan yang diberikan juga dirubah.

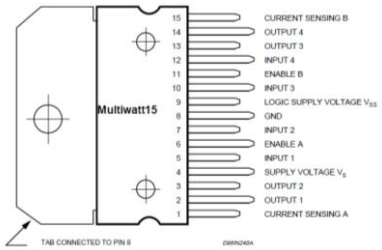


Gambar 2. 5Motor DC

Motor gear DC tidak dapat dikendalikan langsung oleh mikrokontroler, karena kebutuhan arus yang besar sedangkan keluaran arus dari mikrokontroler sangat kecil. Motor *driver* merupakan alternatif yang dapat digunakan untuk menggerakkan motor DC.

2.5.2. IC Driver Motor

L298N adalah IC yang digunakan sebagai *driver* motor DC pada penelitian ini. IC ini menggunakan prinsip kerja *H-Bridge*. Tiap *H-Bridge* dikontrol menggunakan level tegangan TTL yang berasal dari *output* mikrokontroler. Tegangan yang dapat digunakan untuk mengendalikan robot bisa mencapai tegangan 46 VDC dan arus 2A untuk setiap kanalnya. L298 dapat mengontrol 2 buah motor DC, karena di dalam satu komponen L 298 N terdapat dua rangkaian H-Bridge. Berikut ini bentuk IC L298 yang digunakan sebagai motor *driver*.

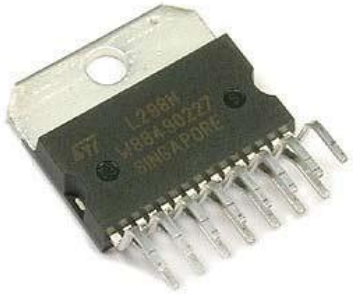


Gambar 2. 6Konfigurasi Pin IC L298N

Pengaturan kecepatan kedua motor dilakukan dengan cara pengontrolan lama pulsa aktif (mode PWM – *Pulse width Modulation*) yang dikirimkan ke rangkaian *driver* motor oleh pengendali (mikrokontroler). *Duty cycle* PWM yang dikirimkan menentukan kecepatan putar motor DC.

Driver motor L298N mempunyai karakteristik sebagai berikut:

1. tegangan operasional sampai 46 Volt,
2. total arus dc sampai 4 Ampere,
3. tegangan saturasinya rendah,
4. pengamanan terhadap suhu yang tinggi,
5. tegangan logic low (0) sampai 1.5 volt.



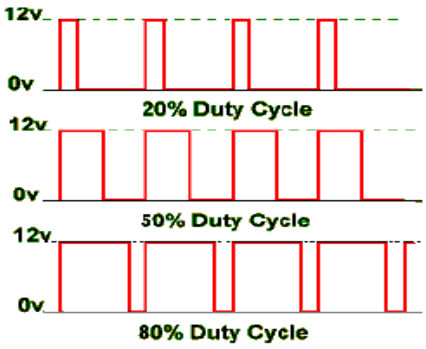
Gambar 2. 7 Bentuk Fisik IC Driver Motor L298N

2.5.3. Pulse Widht Modulation (PWM)

Pulse width Modulation (PWM), adalah sebuah metode untuk pengaturan kecepatan perputaran, dalam hal ini adalah motor DC untuk gerak robot. PWM dapat dihasilkan oleh empat metode, sebagai berikut.

1. Metode analog
2. Metode digital
3. IC diskrit
4. Mikrokontroler

Pada robot ini, metode PWM dikerjakan oleh mikrokontroler. Metode PWM ini akan mengatur lebar atau sempitnya periode pulsa aktif yang dikirimkan oleh mikrokontroler ke driver motor. Pada pengaturan kecepatan robot, nilai PWM mulai dari 0-255. Secara *analog* besaran PWM dihitung dalam prosentase, nilai ini didapat dari perbandingan: T *high* / (T *high* + T *low*) * 100%. Dimana T adalah periode atau waktu tempuh untuk sebuah pulsa, yang terbagi menjadi bagian puncak positif (T *high*) dan puncak negatif (T *low*).



Gambar 2. 8 Ilustrasi persentasi PWM

Semakin rapat periode antar pulsa, maka frekuensi yang dihasilkan akan semakin tinggi, ini berarti kecepatan akan bertambah. Semakin lebar jarak antar pulsa, maka frekuensi semakin rendah ini berarti kecepatan berkurang atau menurun. Kondisi pemberian kecepatan harus disesuaikan dengan kondisi *track* yang akan dilewati oleh robot, misal pada saat jalan lurus, naik atau turun harus mendapatkan nilai PWM yang tepat.

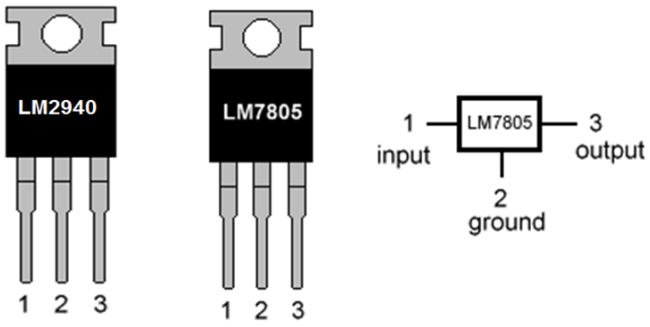
2.6 Regulator

Catu daya adalah faktor pendukung yang sangat penting, karena mikrokontroler, sensor-sensor dan aktuator dapat bekerja karena adanya tegangan. *Power supply* yang dibutuhkan sebanyak 3 buah, pertama untuk mencatu mikrokontroler yang membutuhkan level tegangan 5V. Kedua, mencatu modul *Wifly* RN-171-XV dengan tegangan 3.3Vdc. Ketiga, tegangan kerja aktuator berupa motor DC. Pemisahan tegangan ini dimaksudkan agar tegangan yang masuk ke mikrokontroler tidak terganggu oleh tegangan dari motor. Tegangan yang digunakan berasal dari baterai *rechargeable* Li-Po 7,4V 1000mAH, karena mikrokontroler dan modul *Bluetooth* **JY-MCU BT board v1.3 slave** hanya mampu bekerja pada level tegangan 5V, maka diperlukan regulator tegangan untuk

mendapatkan tegangan 5V, dengan demikian digunakan IC regulator LD1117-3.3, LM2940 dan atau LM7805CT.

Karakteristik IC regulator :

1. Ada tipe positif untuk meregulasi tegangan positif.
2. Ada tipe negatif untuk meregulasi tegangan negatif.
3. Penerapan IC mengharuskan tegangan *input* lebih besar dari tegangan *output* (Vin > Vout).
4. Transistor dalam konfigurasi tunggal-emiter, dimana tegangan *input* Vin > tegangan *output* Vout hanya kalau transistornya jenuh.



Gambar 2. 9 Regulator LM2940CT dan LM7805CT

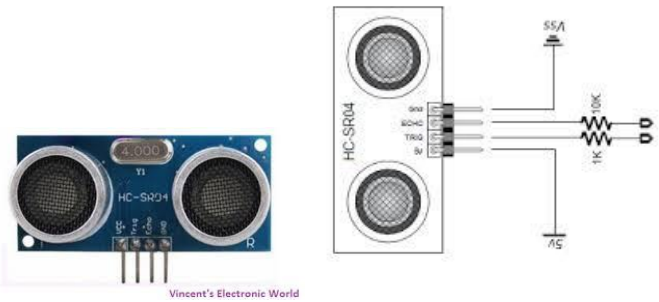
2.7 Sensor

Sensor adalah sesuatu yang digunakan untuk mendeteksi adanya perubahan lingkungan fisik atau kimia. Variabel keluaran dari sensor yang diubah menjadi besaran listrik disebut Transduser. Pada saat ini, sensor tersebut telah dibuat dengan ukuran sangat kecil dengan orde nanometer. Ukuran yang sangat kecil ini sangat memudahkan pemakaian dan menghemat energi.

2.7.1 Sensor Ultrasonik

Sensor ultrasonik bekerja berdasarkan prinsip pantulan gelombang suara, dimana sensor ini menghasilkan gelombang suara yang kemudian menangkapnya kembali dengan perbedaan waktu sebagai dasar penginderaannya. Perbedaan waktu antara gelombang suara dipancarkan dengan ditangkapnya kembali gelombang suara tersebut adalah berbanding lurus dengan jarak atau tinggi objek yang memantulkannya. Jenis objek yang dapat diindera diantaranya adalah: objek padat, cair, butiran maupun tekstil.

HC-SR04 memiliki 4 pin, VCC, TRIG, ECHO dan GND. VCC dihubungkan dengan 5V dari Arduino dan GND dengan GND pada Arduino. TRIG dan ECHO dihubungkan dengan pin mikrokontroler. Untuk lebih jelasnya dapat dilihat pada gambar berikut ini.



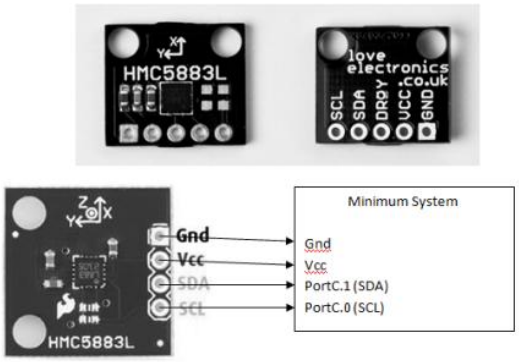
Gambar 2. 10 HC-SR04

2.7.2 Sensor HMC5883L

Sensor HMC5883L ini adalah sensor yang sangat sensitif sekali terhadap rotasi dan arah hadap sensor, dikarenakan sensor ini menggunakan medan magnet sebagai acuan dari pendeteksiannya. Sensor ini memiliki port koneksi ke mikrokontroller yaitu SDA, SCL, VCC dan GND, sehingga port I^2C ini adalah port I/O dari sensor ini. Data yang telah didapat untuk 4 buah mata angin yaitu.

Tabel 2. 1Data arah mata angin pada HMC5883L

Arah Mata Angin	Data
Utara	255 < X < 257
Selatan	79 < X < 81
Timur	16 < X < 18
Barat	148 < X < 150

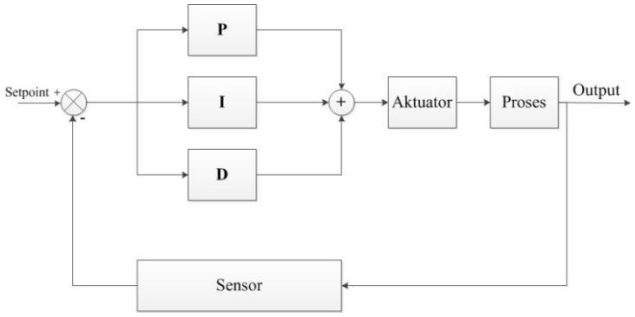


Gambar 2. 11 Sensor HMC5883L

2.8 Pengontrolan Proporsional Integral Derivative (PID)

Sistem Kontrol *Proportional–Integral–Derivative controller* (PID) merupakan pengendalian *error* dengan cara memasukkan *error* tersebut ke dalam *input* yang akan dibandingkan dengan sistem pengendalian. Tujuan dari PID untuk menghasilkan *output* atau keluaran yang sesuai dengan *set point* yang diberikan dengan cara mengurangi *error tersebut.* Pengontrolan PID menggunakan *closed loop* atau umpan balik, yaitu program diolah pada mikrokontroler, lalu menjalankan actuator setelah itu mengeluarkan *output.* Keluaran akan dibandingkan dengan sensor agar dapat mencapai *setpoint* yang diinginkan.

Dibawah ini diperlihatkan gambar blok diagram pengontrol PID dengan *closed loop.*



Gambar 2. 12 Pengontrolan PID dengan Closed loop

PID dapat juga digambarkan dengan persamaan :

$$u(t) = Kp.e(t) + Ki\int_0^1 e(t)\,dt + Kd\frac{de(t)}{dt} \quad \text{..... (1)}$$

dengan,

$$Ki = Kp * \frac{1}{Ti}\ dan\ Kd = Kp * Td \quad \text{..... (2)}$$

Keterangan :

u(t) : *Output* dari pengontrol PID

Kp : *Gain proporsional*

Ti : *Time integral*

Td : *Time derivative*

Ki : *Gain integral*

Kd : *Gain derivative*

Pengontrol PID merupakan paduan dari Pengontrol *Proporsional, Integral, Derivative.* Cara kerja dari masing-masing pengontrol adalah sebagai berikut.

1. Pengontrol *proporsional* memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan. Ciri-ciri pengontrol *proporsional :*
   - jika nilai Kp kecil, pengontrol *proporsional* hanya mampu koreksi kesalahan yang kecil, sehingga akan menghasilkan repon yang lambat,

- jika nilai Kp dinaikkan, respon system akan semakin cepat,
- jika nilai Kp diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan system bekerja tidak stabil atau respon system akan berosilasi,
- nilai Kp dapat diatur sehingga mengurangi *steady state error* tetapi tidak menghilangkannya

2. Pengontrol *Integral* berfungsi untuk menghilangkan *steady state error*. Ciri-ciri dari pengontrol *integral* adalah sebagai berikut :
   - keluaran pengontrol *integral* membutuhkan selang waktu tertentu, sehingga pengontrol *integral* cenderung memperlambat respon,
   - Keluaran pengontrol bertahan pada nilai sebelumnya, ketika sinyal kesalahan berharga nol,
   - keluaran menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnyasinyal kesalahan dan nilai Ki, jika sinyal kesalahan tidak berharga nol,
   - Ki yang berharga besar mempercepat hilangnya nilai *offset.* Tetapi semakin besar nilai Ki mengakibatkan peningkatan osilasi dari sinyal keluaran pengontrol.
3. Pengontrol *Derivative* berfungsi untuk memperbaiki sekaligus mempercepat respon *transient*. Ciri-ciri pengontrol *derivative* adalah sebagai berikut:
   - pengontrol tidak dapat menghasilkan keluaran jika tidak ada perubahan pada *input* (berupa sinyal perubahan kesalahan),

- jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan pengontrol tergantung pada nilai Kd dan laju perubahan sinyal kesalahan,
- pengontrol differensial mempunyai suatu karakter untuk mendahului, sehingga pembangkit kesalahan menjadi sangat besar. Jadi pengontrol differensial dapat mengantisipasi pembangkit kesalahan, memberikan aksi yang sangat bersifat *korektif* dan cenderung meningkatkan stabilitas sistem,
- dengan meningkatkan nilai Kd, dapat meningkatkan stabilitas sistem dan mengurangi *overshoot.*

2.9 IDE Arduino

Software arduino yang akan digunakan adalah *driver* dan IDE. IDE diciptakan untuk para pemula bahkan yang tidak memiliki *basic* bahasa pemrograman sama sekali karena menggunakan bahasa C++ yang telah dipermudah melalui *library*. IDE Arduino adalah *software* canggih yang ditulis dengan menggunakan bahasa Java.

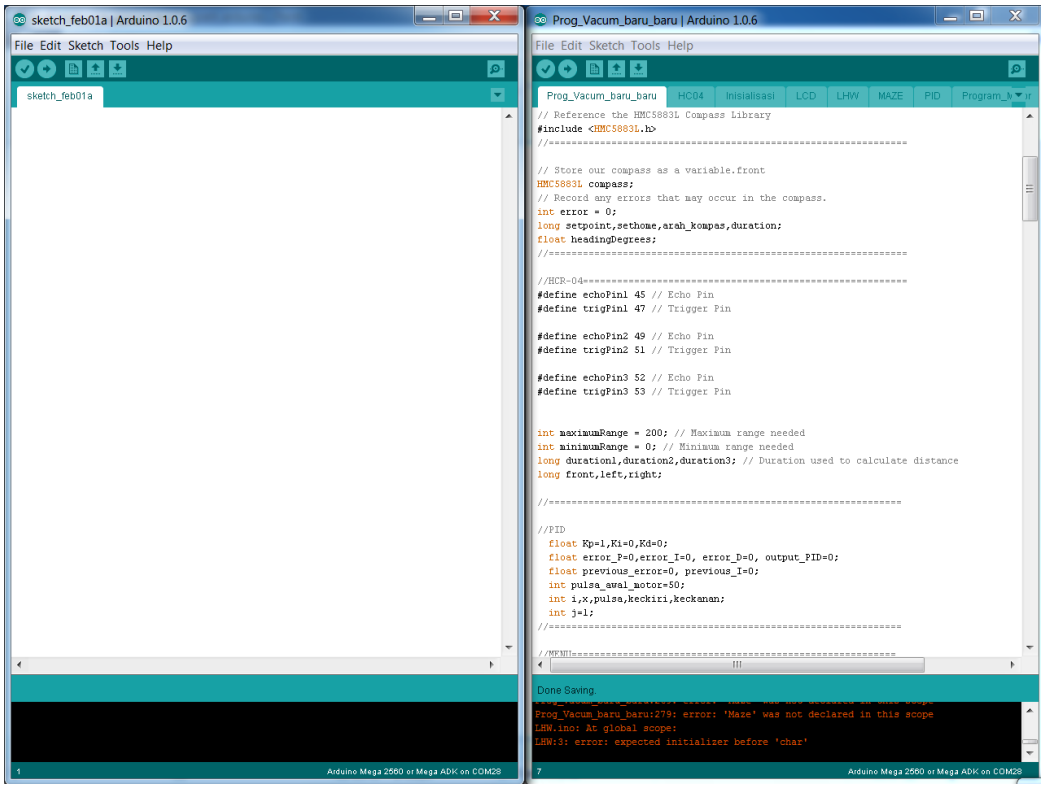
Software IDE arduino terdiri dari 3 bagian :

1. Editor program, untuk menulis dan mengedit program dalam bahasa *processing*. Listing program pada arduino disebut *sketch.*

2. *Compiler,* sebuah modul yang berfungsi mengubah bahasa *processing* (kode program) menjadi kode biner karena kode biner merupakan bahasa program yang dipahami oleh mikrokontroler.

3. *Uploder,* sebuah modul yang berfungsi memasukkan kode biner ke dalam memori mikrokontroler.

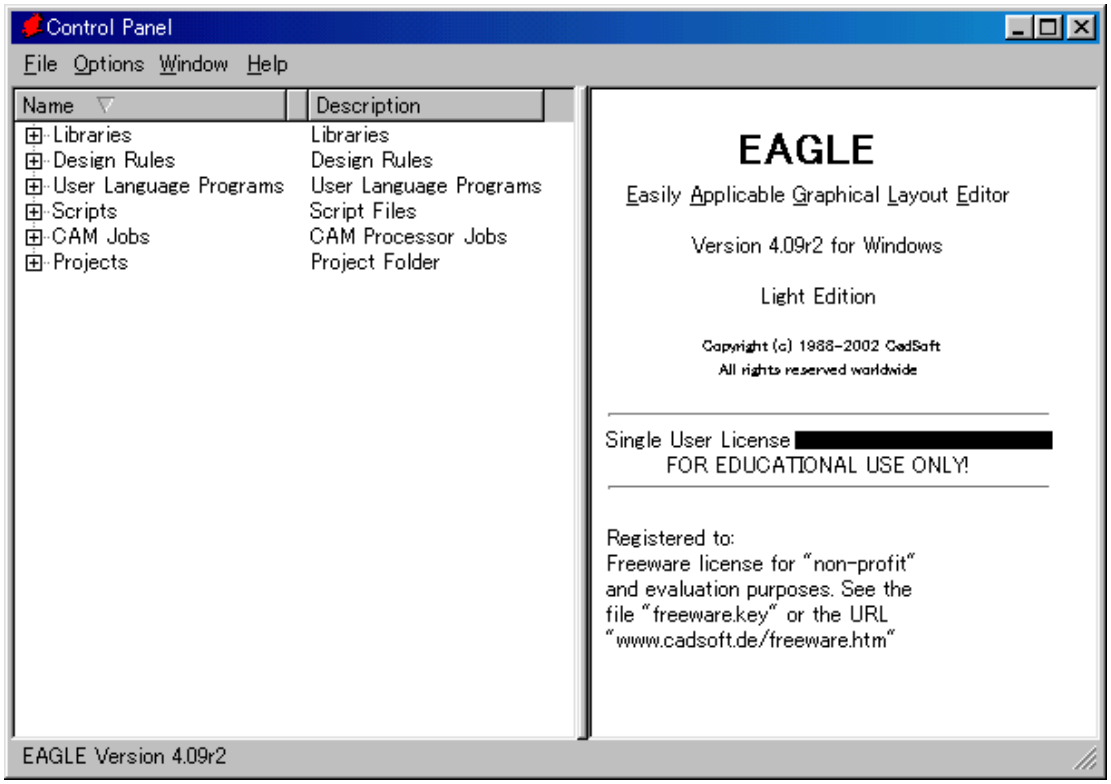
Struktur perintah pada arduino secara garis besar terdiri dari dua bagian yaitu *void setup* dan *void loop. Void setup* berisi perintah yang akan dieksukusi hanya satu kali sejak arduino dihidupkan sedangkan *void loop* berisi perintah yang akan dieksekusikan berulang-ulang selama arduino dinyalakan.



Gambar 2. 13 Tampilan Jendela Program CodeWizard AVR

2.10 Eagle

Software eagle merupakan *software* yang dapat membantu untuk mendisain suatu *printed circuit board* (PCB). Selain mudah digunakan, di dalam *software* Eagle juga terdapat berbagai macam jenis atau merek komponen elektronika yang mendukung dalam pembuatan PCB. Salah satu fasilitas penting yang dimiliki oleh Eagle adalah "*print*" yang memungkinkan kita untuk mencetak hasil PCB yang kita buat secara langsung.



Gambar 2. 14Software Eagle