

BAB II

TINJAUAN PUSTAKA

2.1 Proporsional Integral Derivative (PID)

Sistem kontrol PID (*Proportional Integral Derivative controller*) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut Feed back . Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (*Proportional*), D (*Derivative*) dan I (*Integral*), dengan masing-masing memiliki kelebihan dan kekurangan. Dalam implementasinya masing-masing cara dapat bekerja sendiri maupun gabungan diantaranya. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran system terhadap masukan tertentu sebagaimana yang diinginkan, sinyal keluaran di definisikan sebagai berikut :

2.1.1 Kontrol Proporsional

Proporsional menangani kesalahan (error) dengan segera, yang mana kesalahan tersebut dikalikan dengan suatu nilai konstanta K_p . Jika nilai kesalahan sama dengan nol, maka keluaran dari kontroler proportional juga sama dengan nol. Akan tetapi, kontroler proportional tidak akan mencapai nilai setpoint-nya jika nilai keluarannya yang dibutuhkan bukan nol untuk menjaga setpoint-nya.

2.1.2 Kontrol Integratif

Integratif mempelajari nilai yang telah lalu (lampau), kesalahan diintegrasikan dan dikalikan dengan konstanta K_i . Bentuk integrasi memungkinkan kontroler untuk menghilangkan kondisi kesalahan tetap (steady-state error) jika proses membutuhkan nilai masukan bukan nol untuk mendapatkan setpoint yang diinginkan. Suatu integral kontroler akan bereaksi terhadap kesalahan (error) oleh peningkatan nilai yang ditambahkan ke nilai keluarannya. Sehingga hal ini akan memaksa kontroler untuk mencapai nilai setpoint-nya lebih cepat dibandingkan dengan hanya kontroler proportional dan menekan atau menghilangkan kondisi

kesalahan tetap, hal ini juga menjamin bahwa proses overshoot di setpoint diawali dari nilai integral yang secara berkelanjutan ditambahkan ke nilai keluarannya[1].

2.1.3 Kontrol Derivative

Derivative untuk mengantisipasi nilai yang akan datang, derivative yang pertama dari kesalahan yang terjadi (error) dikalikan dengan suatu konstanta K_d . Hal ini dapat dipakai untuk mengurangi besarnya overshoot yang dihasilkan oleh komponen proportional dan integral, tetapi kontroler akan sedikit lebih lambat untuk mencapai setpoint.

2.2 Robot

Pada dasarnya robot jika ditinjau dari jenisnya terdiri dari dua jenis, yaitu non mobile dan mobile robot. Robot non mobile adalah robot yang melaksanakan aksinya tidak berpindah tempat, sedangkan robot mobile adalah robot yang melaksanakan aktivitasnya dengan bergerak dan berpindah dari suatu tempat ke tempat lain. Saat ini robot selain untuk membantu pekerjaan manusia juga digunakan sebagai hiburan untuk tujuan yang bermacam-macam seperti kompetisi robot yang ditunjukkan untuk pengembangan inovasi teknologi robot.

2.3 Mikrokontroler

Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol atau pengendali rangkaian elektronik dan umumnya dapat menyimpan program didalamnya (Widodo, 2000). Sistem yang digunakan mikrokontroler sering disebut sebagai embedded system atau dedicated system. Embedded system adalah sistem pengendali yang tertanam pada suatu produk, sedangkan dedicated system adalah sistem pengendali yang dimaksudkan hanya untuk suatu fungsi tertentu. Penggunaan mikrokontroler lebih menguntungkan dibandingkan penggunaan mikroprosesor. Hal ini dikarenakan dengan mikrokontroler tidak perlu lagi penambahan memori dan I/O eksternal selama memori dan I/O internal masih bisa mencukupi. Selain itu proses produksinya secara massal, sehingga harganya menjadi lebih murah dibandingkan mikroprosesor. Pada sebuah chip mikrokontroler umumnya memiliki fitur-fitur sebagai berikut:

1. Central processing unit mulai dari processor 4-bit yang sederhana hingga processor kinerja tinggi 64-bit.
2. Input/output antarmuka jaringan seperti serial port (UAR)
3. Antarmuka komunikasi serial lain seperti IC, serial peripheral interface and controller area network untuk sambungan sistem.
4. Periferal seperti timer dan watchdog.
5. RAM untuk menyimpan data.
6. ROM, EPROM, EEPROM atau flash memory untuk menyimpan program dikomputer.
7. Pembangkit clock biasanya berupa resonator rangkaian RC.

Pengubah analog ke digital. Berikut ini jenis-jenis mikrokonktroler yang telah umum digunakan:

2.3.1 Programmable Interface Controller (PIC)

Pada awalnya, PIC merupakan kependekan dari Programmable Interface Controller. Tetapi pada perkembangannya berubah menjadi Programmable Intelligent Computer. PIC termasuk keluarga mikrokonktroler berarsitektur Harvard yang dibuat oleh Microchip Technology. Awalnya dikembangkan oleh Divisi Mikroelektronik General Instruments dengan nama PIC1640. Sekarang Microhip telah mengumumkan pembuatan PIC-nya yang keenam. PIC cukup populer digunakan oleh para developer dan para penghobi ngoprek karena biayanya yang rendah, ketersediaan dan penggunaan yang luas, database aplikasi yang besar, serta pemrograman (dan pemrograman ulang) melalui hubungan serial pada komputer.

2.3.2 Alv and Vegard's Risc processor (AVR)

Mikrokonktroler Alv and Vegard's Risc processor atau sering disingkat AVR merupakan mikrokonktroler RISC 8 bit. Karena RISC inilah sebagian besar kode instruksinya dikemas dalam satu siklus clock. AVR adalah jenis mikrokonktroler yang paling sering dipakai dalam bidang elektronika dan instrumentasi.

2.4. Bahasa C++

C++ adalah bahasa pemrograman komputer yang di buat oleh (Bjarne Stroustrup) merupakan perkembangan dari bahasa C dikembangkan di Bell Labs (Dennis Ritchie) pada awal tahun 1970-an, Bahasa itu diturunkan dari bahasa sebelumnya, yaitu BCL, Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem Unix, Pada perkembangannya, versi ANSI (American National Standart Institute) Bahasa pemrograman C menjadi versi dominan, Meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk sistem embedded, Bjarne Stroustrup pada Bel labs pertama kali mengembangkan C++ pada awal 1980-an, Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (low level coding).

Pada C++ ditambahkan konsep-konsep baru seperti class dengan sifat-sifatnya seperti inheritance dan overloading. Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (Object Oriented Programming). Perbedaan Antara Bahasa pemrograman C dan C++ meskipun bahasa-bahasa tersebut menggunakan sintaks yang sama tetapi mereka memiliki perbedaan, C merupakan bahasa pemrograman prosedural, dimana penyelesaian suatu masalah dilakukan dengan membagi-bagi masalah tersebut kedalam su-submasalah yang lebih kecil, Selain itu, C++ merupakan bahasa pemrograman yang memiliki sifat Pemrograman berorientasi objek, Untuk menyelesaikan masalah, C++ melakukan langkah pertama dengan menjelaskan class-class yang merupakan anak class yang dibuat sebelumnya sebagai abstraksi dari object-object fisik, Class tersebut berisi keadaan object, anggota-anggotanya dan kemampuan dari objectnya, Setelah beberapa Class dibuat kemudian masalah dipecahkan dengan Class.

Bahasa C adalah sebuah bahasa dasar tingkat tinggi yang sifatnya kompleks dan membangun logika atau algoritma.C++ merupakan bahasa pemrograman yang memiliki sifat Pemrograman berorientasi objek, Untuk menyelesaikan masalah, C++ melakukan langkah pertama dengan menjelaskan class-class yang merupakan anak class yang dibuat sebelumnya sebagai abstraksi

dari object-object fisik, Class tersebut berisi keadaan object, anggota-anggotanya dan kemampuan dari objectnya, Setelah beberapa Class dibuat kemudian masalah dipecahkan dengan Class. Bahasa C adalah bahasa pemrograman prosedural yang memungkinkan kita untuk membuat prosedur dalam menyelesaikan suatu masalah.

Bahasa C++ adalah bahasa pemrograman yang berorientasi pada objek. Bahasa tingkat tinggi merupakan bahasa yang mudah dipahami oleh manusia, C dan C++ merupakan contoh bahasa dari bahasa tingkat tinggi. Contoh lain dari bahasa tingkat tinggi adalah Pascal, Perl, Java, dan lain lain. Sedangkan bahasa tingkat rendah merupakan bahasa mesin atau bahasa assembly.

Secara sederhana sebuah komputer hanya dapat mengeksekusi program yang ditulis dalam bahasa mesin. Oleh karena itu, jika suatu program ditulis dalam bahasa tingkat tinggi, maka program tersebut harus diproses dahulu sebelum bisa dijalankan dengan komputer. Proses untuk mengubah dari bahasa tingkat tinggi ke bahasa tingkat rendah dalam bahasa pemrograman ada 2 tipe yaitu interpreter dan compiler. Bahasa pemrograman seperti C dan C++ merupakan contoh dari tipe compiler. Namun ada bahasa yang menggabungkan 2 tipe ini salah satunya adalah bahasa Java. Kelebihan dan kekurangan bahasa pemrograman C++ itu pada dasarnya bahasa C juga, nah kemampuan C yang utama itu adalah seperti :

1. Kode bahasa C++ dengan portabilitas dan fleksibilitas yang tinggi untuk semua jenis komputer.
2. bahasa Standard-nya ANSI bisa dipakai diberbagai platform.
3. kecepatan program jika dibanding dengan program yang sama buatan bahasa lain, relatif lebih cepat.
4. Bahasa C++ tersedia hampir di semua jenis komputer.
5. Bahasa C++ hanya menyediakan sedikit kata-kata kunci (hanya terdapat 48 kata kunci).
6. Dukungan pustaka fungsi dan kelas yang banyak sehingga memungkinkan pembuatan aplikasi makro.

7. C++ adalah bahasa yang terstruktur, dengan demikian akan lebih mendukung OOP.
8. Bahasa C++ termasuk bahasa tingkat menengah dan lebih dekat dengan bahasa mesin.
9. Kode program bersifat reuseable, sehingga dapat digunakan kembali pada project lain dengan hanya menggunakan library dan file header.
10. C++ dapat membuat aplikasi graphic processor berkualitas tinggi. Jika dibandingkan dengan VB atau Delphi C++ memang tidak se-RAD dua bahasa tersebut , dan cenderung lebih rumit. Namundari perbandingan tersebut, sudah jelas pasti ada trade-off, seperti kecepatan, efisiensi kode, pengstrukturran kode.

Tidak hanya kelebihan-kelebihan yang dimiliki oleh C++, tetapi juga memiliki kekurangan seperti:

1. C++ tidak murni OOP sehingga kurang cocok untuk mengajarkan Konsep OOP karena kaidah-kaidah OOP dapat dilanggar. Dandi C++ ada konsep pointer yang sangat membingungkan, ini salah satu alasan mengapa C++ menjadi momok yg paling tidakdisukai.
2. C++ walaupun tidak terpengaruh oleh Sistem Operasi tetapi tool untuk developmentnya harus spesifik pada salah satusistem operasi contoh Visual Studio hanya dapat berjalan di Windows. Alasan ini juga nantinya akan dapat menurunkan minat terhadap sistem operasi yang lain contohnya Linux. Apabila dalam praktiknya menggunakan Visual Studio makaotomatis akan menggunakannya juga dan pada akhirnya malas untuk mencoba sistem operasi yang lain.
3. Implementasi C++ dalam teknologi IT pada saat sekarang sudah sangat sedikit sekali.
4. “susah” untuk membuat sesuatu dengan C++ sehingga kepeminatannya dalam memperdalam Programming akhirnya harus kandaskecuali dengan inisitif sendiri mempelajari bahasa/teknologi lain.

2.5 Arduino Mega 2560

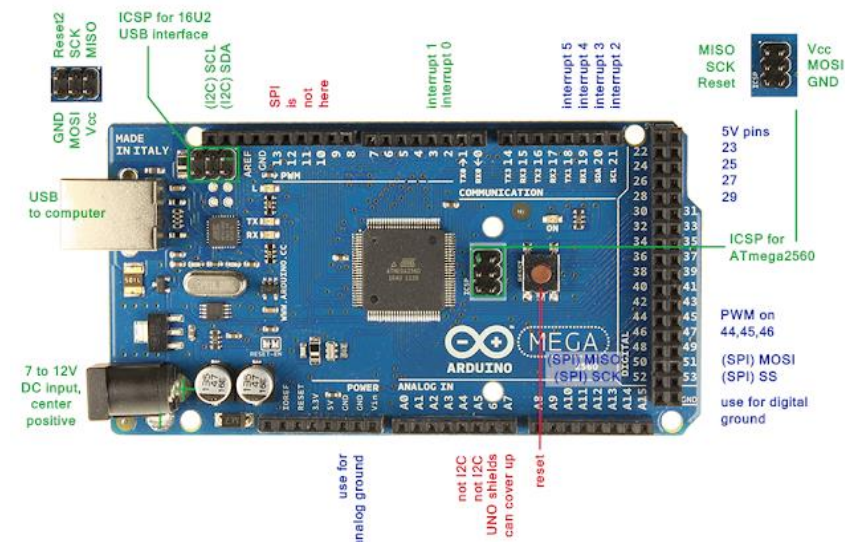
Board Arduino Mega 2560 adalah sebuah Board Arduino yang menggunakan ic Mikrokontroler ATmega 2560. Board ini memiliki Pin I/O yang relatif banyak, 54 digital Input / Output, 15 buah di antaranya dapat di gunakan sebagai output PWM, 16 buah analog Input, 4 UART. Arduino Mega 2560 di lengkapi kristal 16. Mhz Untuk penggunaan relatif sederhana tinggal menghubungkan power dari USB ke PC / Laptop atau melalui Jack DC pakai adaptor 7-12 V DC. Untuk lebih jelasnya dapat di lihat dari spesifikasi Arduino Mega 2560 di bawah ini.

2.5.1 Spesifikasi Arduino Mega

<i>Mikrokontroler</i>	<i>ATMega 2560</i>
<i>Tegangan Operasional</i>	<i>5V</i>
<i>Tegangan Input (rekomendasi)</i>	<i>7-12V</i>
<i>Tegangan Input (limit)</i>	<i>6-20V</i>
<i>Pin Digital I/O</i>	<i>54 (of which 15 provide PWM output)</i>
<i>Pin Analog Input</i>	<i>16</i>
<i>Arus DC per Pin I/O</i>	<i>20 mA</i>
<i>Arus DC untuk Pin 3.3 V</i>	<i>50 mA</i>
<i>Memori Flash</i>	<i>256 KB of which 8 KB used by bootloader</i>
<i>SRAM</i>	<i>8 KB</i>
<i>EEPROM</i>	<i>4 KB</i>
<i>Clock Speed</i>	<i>16 MHz</i>
<i>LED_BUILTIN</i>	<i>13</i>
<i>Panjang dan Lebar</i>	<i>101.52 mm*53.3mm</i>
<i>Berat</i>	<i>37 g</i>

2.5.2 Arduino Mega 2560 PIN OUT

Pada gambar 2.4 terdapat keterangan dari setiap pin yang ada pada Arduino Mega 2560.



Gambar 2.1 PIN OUT Arduino Mega
(Sumber : Lab Elektronika ,2017)

Pin digital Arduino Mega2560 ada 54 Pin yang dapat di gunakan sebagai Input atau Output dan 16 Pin Analog berlabel A0 sampai A15 sebagai ADC, setiap Pin Analog memiliki resolusi sebesar 10 bit. Arduino Mega 2560 di lengkapi dengan pin dengan fungsi khusus, sebagai berikut :

1. **Serial 4 buah** : Port Serial : Pin 0 (RX) dan Pin 1 (TX) ;Port Serial 1 : Pin 19 (RX) dan Pin 18 (TX); Port Serial 2 : Pin 17 (RX) dan Pin 16 (TX); Port Serial 3 : Pin 15 (RX) dan Pin 14 (TX). Pin Rx di gunakan untuk menerima data serial TTL dan Pin (Tx) untuk mengirim data serial TTL.
2. **External Interrupts 6 buah** : Pin 2 (Interrupt 0), Pin 3 (Interrupt 1), Pin 18 (Interrupt 5), Pin 19 (Interrupt 4), Pin 20 (Interrupt 3) dan Pin 21 (Interrupt 2).

3. **PWM 15 buah** : 2,3,4,5,6,7,8,9,10,11,12,13 dan 44,45,46 pin-pin tersebut dapat di gunakan sebagai Output PWM 8 bit.
4. **SPI** : Pin 50 (MISO), Pin 51 (MOSI), Pin 52 (SCK), Pin 53 (SS) ,Di gunakan untuk komunikasi SPI menggunakan SPI Library.
5. **I2C** : Pin 20 (SDA) dan Pin 21 (SCL) , Komunikasi I2C menggunakan wire library.
6. **LED** : 13. Buit-in LED terhubung dengan Pin Digital 13.[6]

2.6 Arduino (Software) IDE

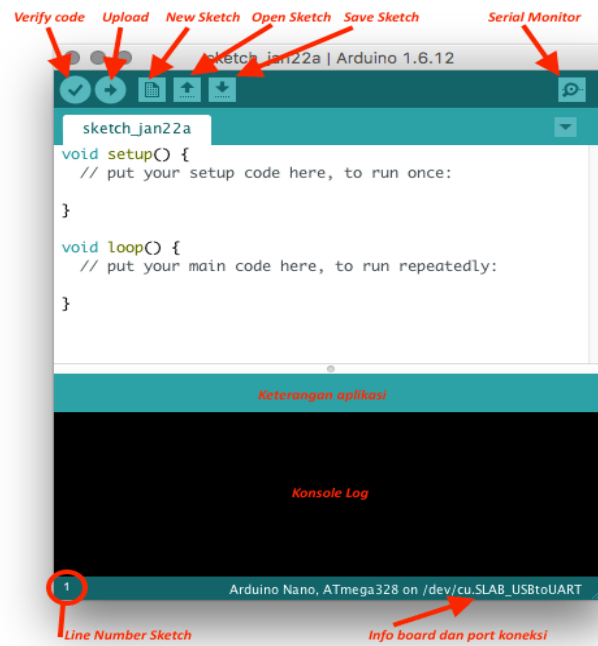
Arduino IDE adalah sebuah perangkat lunak pemrograman yang berfungsi untuk memasukkan program ke dalam sirkuit board modul Arduino salah satunya Arduino Nano. IDE merupakan kependekan dari *Integrated Development Enviroenment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan.

Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut Wiring yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

Editor Programming pada umumnya memiliki fitur untuk *cut / paste* dan untuk *find / replace* teks, demikian juga pada Arduino IDE. Pada bagian keterangan aplikasi memberikan pesan balik saat menyimpan dan mengeksport serta sebagai tempat menampilkan kesalahan. *Konsol log* menampilkan teks log

dari aktifitas Arduino IDE. Termasuk pesan kesalahan yang lengkap dan informasi lainnya. Pada gambar 2.4 merupakan tampilan dari *software* arduino IDE yang merupakan tampilan awal pengguna aplikasi IDE untuk menmbuat program kerja.



Gambar 2.2 Ilustrasi *Software* Arduino

(Sumber : [Sinauarduino](#), 2016)

1. *Verify* pada versi sebelumnya dikenal dengan istilah *Compile*. Sebelum aplikasi di- *upload* ke board Arduino, biasakan untuk memverifikasi terlebih dahulu sketch yang dibuat. Jika ada kesalahan pada sketch, nanti akan muncul error. Proses *Verify / Compile* mengubah sketch ke binary code untuk di-upload ke mikrokontroler.
2. *Upload* tombol ini berfungsi untuk mengupload sketch ke board Arduino. Walaupun kita tidak mengklik tombol *verify*, maka sketch akan di- *compile*, kemudian langsung diupload ke board. Berbeda dengan tombol *verify* yang hanya berfungsi untuk memverifikasi *source code* saja.

3. *New Sketch* Membuka window dan membuat sketch baru.
4. *Open Sketch* Membuka sketch yang sudah pernah dibuat. Sketch yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file .ino
5. *Save Sketch* menyimpan sketch, tapi tidak disertai dengan mengcompile.
6. *Serial Monitor* Membuka interface untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya.
7. Keterangan Aplikasi pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal Compiling dan Done Uploading ketika kita mengcompile dan mengupload sketch ke board Arduino
8. *Konsol log* Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang sketch akan muncul pada bagian ini. Misal, ketika aplikasi mengcompile atau ketika ada kesalahan pada sketch yang kita buat, maka informasi error dan baris akan diinformasikan di bagian ini.
9. *Baris Sketch* bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada sketch.
10. Informasi *Board dan Port* Bagian ini menginformasikan port yang dipakai oleh board Arduino.

2.7 ESP32-Camera

ESP32-Cameea adalah papan pengembangan WiFi / Bluetooth dengan mikrokontroler ESP32 dan kamera. Ada juga sejumlah GPIO yang tersedia dan ada koneksi untuk antena eksternal. Dengan itu, dewan terlihat sedikit seperti T-Journal TTGO dari Lilygo, tetapi ada juga beberapa perbedaan penting.

Papan ini tidak memiliki antarmuka USB ke serial. Pemrograman harus, oleh karena itu, dilakukan melalui antarmuka eksternal. Ini juga berarti tidak ada daya yang dapat disediakan melalui USB. Lebih lanjut, rangkaian manajemen baterai dan tampilan OLED tidak ada. Sebaliknya, board ini memang memiliki slot kartu SD dan senter LED, dan jauh lebih kompak. Modul ESP32-S memiliki antena pada PCB, tetapi juga koneksi U.FL untuk antena eksternal. Anda dapat

mengaktifkan antena yang diinginkan dengan menempatkan resistor nol-ohm (jembatan) yang benar.



Gambar 2.3 ESP32-Cam

(Sumber : <https://id.oneguyoneblog.com/2019/09/09/esp32-cam-esp32-dengan-kamera-dan-slot-sd/>)

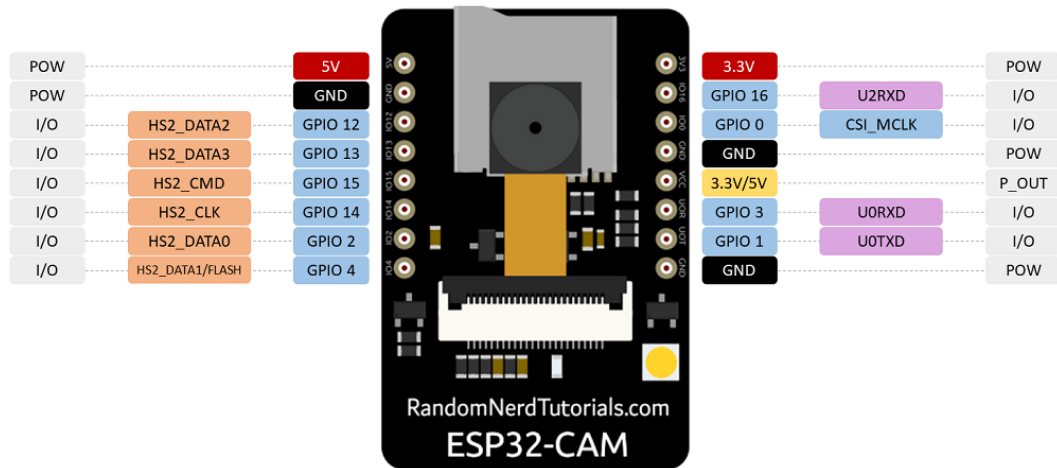
2.7.1 Fitur ESP32-Cam

Berikut adalah daftar dengan fitur *ESP32-Cam* :

1. Modul 802.11b / g / n Wi-Fi BT SoC terkecil
2. CPU 32-bit berdaya rendah, juga dapat melayani prosesor aplikasi
3. Kecepatan clock hingga 160MHz, daya komputasi ringkasan hingga 600 DMIPS
4. 520 KB SRAM internal, 4MPSRAM eksternal
5. Mendukung UART / SPI / I2C / PWM / ADC / DAC
6. Mendukung kamera OV2640 dan OV7670, lampu flash internal
7. Mendukung unggahan gambar WiFi
8. Mendukung kartu TF
9. Mendukung beberapa mode tidur
10. Lwip tertanam dan FreeRTOS
11. Mendukung mode operasi STA / AP / STA + AP
12. Dukungan untuk peningkatan versi firmware lokal dan jarak jauh port serial (FOTA)

2.7.2 Pinout ESP32-Cam

ESP32-CAM adalah papan pengembangan dengan chip ESP32-S, kamera OV2640, slot kartu microSD, dan beberapa GPIO untuk menghubungkan periferal.



Gambar 2.4 Pinout ESP32-Cam

(Sumber : <https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>)

1. Pin Daya

ESP32-CAM hadir dengan tiga GND pin (diwarnai dengan warna hitam) dan dua pin daya (diwarnai dengan warna merah): 3.3V dan 5V. Anda dapat menyalakan ESP32-CAM melalui 3.3V atau 5V pin. Namun, banyak orang melaporkan kesalahan saat menyalakan ESP32-CAM dengan 3.3V, jadi kami selalu menyarankan untuk memberi daya ESP32-CAM melalui pin 5V .

2. Pin keluaran daya

Ada juga pin yang diberi label di silkscreen sebagai VCC (diwarnai dengan persegi panjang kuning). Anda tidak boleh menggunakan pin itu untuk menyalakan ESP32-CAM. Itu adalah pin daya keluaran. Ini dapat menghasilkan 5V atau 3.3V. Dalam kasus kami, ESP32-CAM menghasilkan 3.3V apakah itu didukung dengan 5V atau 3.3V. Di samping pin VCC, ada dua bantalan. Satu diberi label sebagai 3.3V dan lainnya sebagai 5V.

3. Pin Serial

GPIO 1 dan GPIO 3 adalah pin serial (TX dan RX, masing-masing). Karena ESP32-CAM tidak memiliki pemrogram bawaan, Anda perlu menggunakan pin ini untuk berkomunikasi dengan papan dan mengunggah kode, cara terbaik untuk

mengunggah kode ke ESP32-CAM menggunakan programmer FTDI. Pelajari cara mengunggah kode ke ESP32-CAM AI-Thinker. Kamu bisa gunakan GPIO 1 dan GPIO 3 untuk menghubungkan periferal lain seperti output atau sensor setelah mengupload kode. Namun, Anda tidak akan dapat membuka Monitor Serial dan melihat apakah semuanya berjalan dengan baik dengan pengaturan Anda.

4. GPIO 0

GPIO 0 menentukan apakah ESP32 dalam mode berkedip atau tidak. GPIO ini secara internal terhubung ke resistor 10k Ohm pull-up. Ketika GPIO 0 terhubung ke GND, ESP32 masuk ke mode berkedip dan Anda dapat mengunggah kode ke papan. GPIO 0 terhubung dengan GND »ESP32-CAM dalam mode berkedip, untuk membuat ESP32 berjalan "normal", Anda hanya perlu memutuskan GPIO 0 dari GND.

definisi pin untuk ESP32-CAM AI-Thinker di Arduino IDE harus sebagai berikut :

```
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27
#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22
```

2.8 Driver L298

L298 adalah jenis IC driver motor yang dapat mengendalikan arah putaran dan kecepatan motor DC maupun Motor Stepper. Driver motor digunakan untuk mengontrol arah putaran dan kecepatan motor DC yang merupakan penggerak utama dari rangkaian proyek akhir ini. IC driver motor L293 yang didalamnya terdapat rangkaian H-Bridge akan mengontrol putaran motor sesuai data masukan digital yang berasal dari PLC Zelio SR2 B201 BD, dan pada IC L293 ini juga

terdapat pin untuk pengaturan aplikasi PWM (Pulse Width Modulator) yang akan mengatur kecepatan motor dc yang dikendalikannya. L293 memiliki rangkaian dual H-Bridge, sehingga mampu mengendalikan dua buah motor DC sekaligus.

Karakteristik dari driver motor L293 adalah:

1. Tegangan operasi supply sampai dengan 36 Volt.
2. Total arus DC sampai dengan 1A.
3. Tegangan logic "0" sampai dengan 1,5 Volt.
4. Memiliki dua Enable input.

Fungsi dari tiap-tiap pin driver motor L293 adalah sebagai berikut:

1. Output 1 dan Output 2 (pin 3 dan pin 6)
Pin ini merupakan output untuk bridge A.
2. Vs (pin 8)
Merupakan pin supply tegangan untuk output.
3. Input 1 dan Input 2 (pin 2 dan pin 7)
Pin ini digunakan untuk mengontrol bridge A.
4. Enable 1 dan Enable 2 (pin 1 dan pin 9)
Pin ini berfungsi untuk mengaktifkan dan menonaktifkan bridge A dan bridge B.
5. Ground (pin 4, 5, 12, dan 13)
Berfungsi sebagai grounding rangkaian driver.
6. Vss (pin 16)
Pin ini berfungsi sebagai supply logic untuk driver.
7. Input 3 dan Input 4 (pin 10 dan 15)

Berfungsi sebagai masukan pada bridge B.

8. Output 3 dan Output 4 (11 dan 14)

Merupakan pin output untuk bridge B.

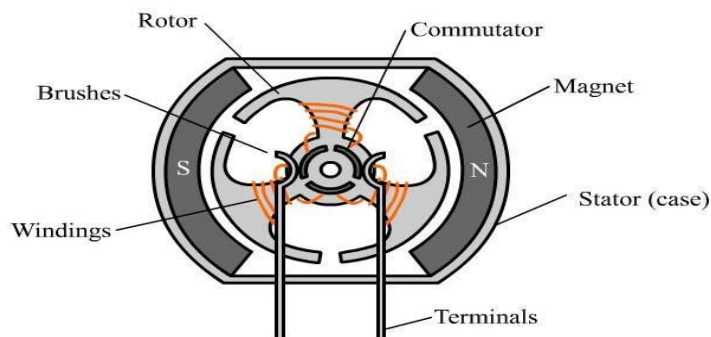
2.8.1 Motor DC (*Direct Current*)

Motor DC (*Direct Current*) adalah peralatan elektromekanik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik. Motor DC merupakan jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya. Dengan memberikan beda tegangan pada kedua terminal tersebut, motor akan berputar pada satu arah, dan bila polaritas dari tegangan tersebut dibalik maka arah putaran motor akan terbalik pula. Polaritas dari tegangan yang diberikan pada dua terminal menentukan arah putaran motor sedangkan besar dari beda tegangan pada kedua terminal menentukan kecepatan motor .



Gambar 2.5 Motor DC

(Sumber : <http://wahyu-umiq.blogspot.com/2013/10/motor-dc.html>)



Gambar 2.6 Bagian *Motor DC (Direct Current)*

(Sumber : <http://wahyu-umiq.blogspot.com/2013/10/motor-dc.html>)

Motor DC memiliki 3 bagian atau komponen utama untuk dapat berputar. Yang ditunjukkan seperti gambar di bawah ini :

1. **Kutub medan.** Secara sederhana bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan dinamo yang menggerakkan *bearing* pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan yaitu kutub utara dan kutub selatan.
2. **Rotor.** Bila arus masuk menuju rotor (bagian motor yang bergerak), maka arus ini akan menjadi elektromagnet. Rotor yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, rotor berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan dinamo.
3. **Komutator.** Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikkan arah arus listrik dalam dinamo. *Commutator* juga membantu dalam transmisi arus antara dinamo dan sumber daya

2.9. Baterai

Baterai adalah alat elektro kimia yang berfungsi untuk menyimpan tenaga listrik dalam bentuk tenaga kimia. Tenaga listrik yang tersimpan akan dialirkan lagi untuk memberikan arus listrik pada lampu posisi, lampu indikator, lampu rem belakang dan klakson.

Konstruksi baterai terdiri dari kotak baterai yang didalamnya terdapat elektrolit asam sulfat, elektrode positif, dan elektrode negatif. Baterai terdiri dari dua jenis yaitu baterai yang hanya dapat dipakai sekali saja atau single use dan baterai yang dapat di isi ulang atau rechargeable.



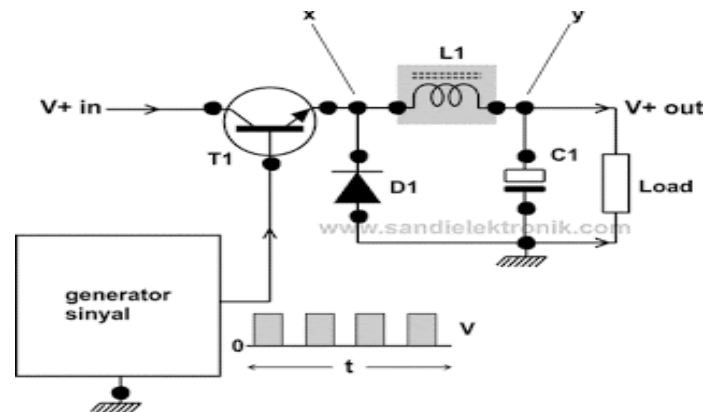
Gambar 2.7 Macam-macam jenis baterai
(Sumber : www.wikipedia.org/baterai)

Gambar 2.8 merupakan macam-macam jenis baterai. Baik baterai primer maupun baterai sekunder, kedua-duanya bersifat mengubah energi kimia menjadi energi listrik. Baterai primer hanya bisa dipakai sekali, karena menggunakan reaksi kimia yang bersifat tidak bisa dibalik *irreversible reaction*. Sedangkan baterai sekunder dapat diisi ulang karena reaksi kimianya bersifat bisa dibalik *reversible reaction*.

2.10. Buck-converter.

Buck-converter adalah konverter penurun tegangan khusus yang menerapkan sistem Switching Mode Power Supply (SMPS). Ia adalah konverter dengan efisiensi yang lebih tinggi jika dibandingkan dengan power-supply penurun tegangan biasa (sistem linier). Efisiensinya dapat mencapai lebih dari 90%.

Buck-converter memanfaatkan sifat induktor terhadap guncangan listrik berfrekwensi tinggi dan bekerja dengan adanya denyut-denyut tegangan (sebagaimana layaknya SMPS). Karena itu di dalam sebuah rangkaian buck-converter selalu terdapat generator sinyal, transistor penguat, dioda, kondensator dan induktor. Konsep dasar rangkaiannya dapat digambarkan sebagai berikut :



Gambar 2.8 Rangkaian BUCK Konverter

(Sumber : <https://www.sandielektronik.com/2016/01/buck-converter.html>)

Kelebihan dari sistem Buck converter antara lain :

1. Efisiensi yang tinggi
2. Rangkaiannya sederhana
3. Tidak memerlukan transformer
4. Tingkatan stress pada komponen switch yang rendah
5. Riak (ripple) pada tegangan keluaran juga rendah sehingga penyaring atau filter yang diperlukan pun relatif kecil

Kekurangan dari sistem Buck Converter antara lain :

- a. Tidak adanya isolasi antara masukan dan keluaran
- b. Hanya satu keluaran yang dihasilkan
- c. Tingkat ripple yang tinggi pada arus masukan
- d. Metode ini sering digunakan pada aplikasi yang membutuhkan sistem yang berukuran kecil[7].

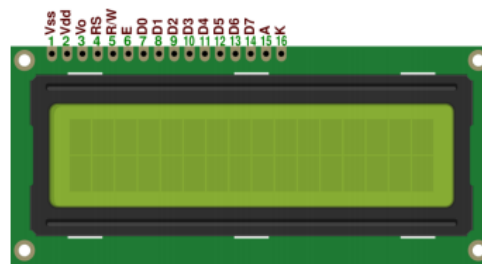
2.11. LCD (*Liquid Crystal Display*)

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. Adapun fitur yang disajikan dalam LCD ini adalah: - Terdiri dari 16 karakter dan 2 baris. - Mempunyai 192 karakter tersimpan. - Terdapat karakter generator terprogram. - Dapat dialamati dengan mode 4-bit dan 8-bit. - Dilengkapi dengan back light. Proses inisialisasi

pin arduino yang terhubung ke pin LCD RS, Enable, D4, D5, D6, dan D7, dilakukan dalam baris LiquidCrystal (2, 3, 4, 5, 6, 7), dimana lcd merupakan variable yang dipanggil setiap kali intruksi terkait LCD akan digunakan. Definisi pin lcd 16x2 dapat dilihat ditabel 2.1 dan gambar 2.17 adalah device LCD. ^[24]

Pin	Diskripsi
1	Ground
2	Vcc
3	Pengatur Kontras
4	Register Select
5	Read / Write LCD Register
6	Enable
7-14	Data I / O Pins
15	VCC + LED
16	Ground – LED

Tabel 2.1 Spesifikasi LCD 16 x 2



Gambar 2.9 LCD (Liquid Crystal Display) 16x2

Berikut ada beberapa fungsi-fungsi dari library LCD :

1. `begin()`

Untuk `begin()` digunakan dalam inialisasi interface ke LCD dan mendefinisikan ukuran kolom dan baris LCD. Pemanggilan `begin()` harus dilakukan terlebih dahulu sebelum memanggil instruksi lain dalam library LCD. Untuk syntax penulisan instruksi `begin()` ialah sebagai berikut. `lcd.begin(cols,rows)` dengan `lcd` ialah nama variable, `cols` jumlah kolom LCD, dan `rows` jumlah baris LCD.

2. `clear()`

Instruksi `clear()` digunakan untuk membersihkan pesan text. Sehingga tidak ada tulisan yang ditampilkan pada LCD.

3. `setCursor()`

Instruksi ini digunakan untuk memposisikan cursor awal pesan text di LCD. Penulisan syntax `setCursor()` ialah sebagai berikut. `lcd.setCursor(col,row)` dengan `lcd` ialah nama variable, `col` kolom LCD, dan `row` baris LCD.

4. `print()`

Sesuai dengan namanya, instruksi `print()` ini digunakan untuk mencetak, menampilkan pesan text di LCD. Penulisan syntax `print()` ialah sebagai berikut. `lcd.print(data)` dengan `lcd` ialah nama variable, `data` ialah pesan yang ingin ditampilkan. ^[24]