

BAB II

LANDASAN TEORI

II.1. Pengertian Perancangan

Desain atau perancangan dalam pengembangan perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional memenuhi target, memenuhi kebutuhan secara implisit dan eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. (Rosa A.S, M. Shalahuddin ; 2011: 21).

Desain atau perancangan adalah mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem *fokus* pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan. (Rosa A.S, M. Shalahuddin ; 2011: 25).

Desain atau perancangan perangkat lunak adalah proses langkah yang *fokus* pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan perangkat lunak ke representasi desain agar dapat di implementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu di dokumentasikan. (Rosa A.S, M. Shalahuddin ; 2011: 27).

Perancangan adalah proses menuangkan ide dan gagasan berdasarkan teori-teori dasar yang mendukung. Proses perancangan dapat dilakukan dengan

cara pemilihan komponen yang akan digunakan, mempelajari karakteristik dan data fisiknya, membuat rangkaian skematik dengan melihat fungsi-fungsi komponen yang dipelajari, sehingga dapat dibuat alat yang sesuai dengan spesifikasi yang diharapkan. (<http://elib.unikom.ac.id/download.php?id=143047>: dikunjungi pada tanggal 28 Juli 2016).

II.2. Pengertian Dasar Sistem Kendali

Sistem kendali merupakan suatu sistem yang keluarannya dikendalikan pada suatu nilai tertentu atau untuk mengubah beberapa ketentuan yang telah ditetapkan dari masukan ke sistem. Untuk merancang suatu sistem yang dapat merespon suatu perubahan tegangan dan mengeksekusi perintah berdasarkan situasi yang terjadi, maka diperlukan pemahaman tentang sistem kendali (*control system*). Sistem kendali merupakan suatu kondisi dimana sebuah perangkat (*device*) dapat dikendalikan sesuai dengan perubahan situasi.

Kendali (pengendalian) adalah sebuah proses atau upaya untuk mencapai tujuan. Dengan demikian, sistem kendali adalah kombinasi dari beberapa komponen (*subsistem*) yang bekerja secara sinergi dan terpadu untuk memperoleh hasil yang diinginkan (tujuan). Sebagai contoh sederhana dan akrab dengan aktivitas sehari-hari dari konsep kendali adalah saat mengendarai kendaraan. Tujuan yang diinginkan dari proses tersebut adalah berjalannya kendaraan pada lintasan (*track*) yang diinginkan. Ada beberapa komponen yang terlibat di dalamnya, misalnya pedal gas, *speedometer*, mesin (penggerak), rem, dan pengemudi. Sistem kendali berkendara berarti kombinasi dari komponen-komponen tersebut yang menghasilkan berjalannya kendaraan pada lintasan yang

diinginkan. Ketika jalan lengang dan aturan memperbolehkan, pengendara mempercepat laju kendaraan dengan membuka pedal gas. Demikian pula, jika ada kendaraan lain di depan atau lampu penyeberangan berwarna merah maka pengendara menginjak rem dan menurunkan kecepatannya. Semua upaya itu dilakukan untuk mempertahankan kendaraan pada lintasan yang diinginkan.

Contoh lain dapat disebutkan berupa proses memindahkan barang oleh tangan kita. Pada proses tersebut, tujuannya adalah posisi atau letak barang yang diinginkan. Komponennya berupa tangan (dalam hal ini tentunya dengan otot tangan), mata, dan otak sebagai pengendali. Pada saat tangan bergerak untuk memindahkan barang, mata akan menangkap informasi tentang posisi pada saat itu. Informasi tersebut diproses oleh otak untuk disimpulkan apakah posisinya sudah benar atau tidak. Selanjutnya, apabila posisinya masih belum tercapai maka otak akan memerintahkan otot tangan untuk bergerak memindahkan barang ke posisi yang diinginkan. Secara umum dapat dikatakan semua proses yang terjadi di alam pada hakikatnya adalah sebuah sistem kendali.

Ada beberapa istilah yang biasanya dipakai dalam pembahasan sistem kendali, yaitu:

1. *Plant*, *Plant* berarti objek yang dikendalikan, misalnya pesawat terbang, reaktor kimia, tungku pembakaran, dan lain-lain.
2. *Input* (masukan), *Input* adalah sinyal yang masuk ke dalam sistem atau subsistem untuk selanjutnya diproses agar menghasilkan *output*.
3. *Output* (keluaran), *Output* berarti sinyal yang dihasilkan oleh proses yang terjadi dalam sistem atau subsistem.

4. *Command input* atau *reference input* (masukkan acuan) adalah tujuan dari proses yang terjadi dalam sistem kendali.
5. *Aktuator*, Subsistem ini berfungsi untuk menghasilkan *output* agar mampu menggerakkan *plant* ke arah yang diinginkan.
6. Pengendali (*controller*), Subsistem pengendali adalah “otak”nya sistem kendali yang berisi algoritma kendali sedemikian sehingga menghasilkan aturan-aturan pengendalian (*control law*).
7. *Sensor* (dan transduser), *Sensor* berfungsi mendeteksi keluaran yang terjadi untuk dibandingkan dengan masukan acuan. Nama *transduser* dipakai apabila dalam prakteknya melibatkan perubahan besaran.
8. *Komparator*, Subsistem ini berfungsi membandingkan dua sinyal yang masuk, yaitu menjumlahkan atau mengurangi.
9. *Disturbance* (gangguan) atau *Noise* (derau), Istilah ini dipakai untuk menamai sinyal yang masuk ke dalam sistem atau *plant*, tetapi bersifat “menggangu” pencapaian proses kendali ke arah tujuan yang diinginkan. (Asep Najmurokhman, 2010 : 1)

II.3. Pengertian Sensor

Dalam rangkaian elektronika untuk keperluan pengukuran atau deteksi, diperlukan suatu bagian yang disebut sensor. Sensor berfungsi untuk menubah besaran yang bersifat fisis atau suhu, tekanan, berat, atau intensitas cahaya menjadi besaran listrik (tegangan atau arus listrik).

Sensor adalah alat untuk mendeteksi atau mengukur sesuatu yang digunakan untuk mengubah variasi mekanis, *magnetic*, panas, dan kimia menjadi

tegangan dan arus listrik. Dalam lingkungan sistem pengendali dan robotika, sensor memberikan kesamaan yang menyerupai mata, pendengaran, hidung, lidah yang kemudian akan diolah oleh kontroler sebagai otaknya. (Petruzella : 2010)

Sensor memiliki suatu ukuran yang disebut sensitivitas. Sensitivitas menunjukkan seberapa besar pengaruh perubahan nilai besaran fisis yang diukur oleh sensor terhadap keluaran dari sensor tersebut. Misalnya, sebuah sensor suhu yang tegangan keluarannya berubah 0,1 V jika terjadi perubahan suhu sebesar 1°C . Maka sensor suhu tersebut dapat dikatakan memiliki sensitivitas sebesar $0,1\text{V}/1^{\circ}\text{C}$. (Endra Pitowarno ; 2006 : 56)

Sensor yang baik memiliki ciri-ciri sebagai berikut:

1. Peka terhadap besaran yang akan diukur.
2. Tidak peka terhadap besaran lain yang tidak akan diukur.
3. Keberadaan sensor tidak mempengaruhi besaran yang akan diukur.

II.4. Balancing Robot

Balancing robot (robot penyeimbang) beroda dua merupakan suatu robot mobile yang memiliki dua buah roda disisi kanan dan kirinya yang tidak akan seimbang apabila tanpa adanya kontroler. Balancing robot ini merupakan pengembangan dari model pendulum terbalik (*inverted pendulum*) yang diletakkan di atas kereta beroda. Menyeimbangkan robot beroda dua memerlukan suatu metode kontrol yang baik dan handal untuk mempertahankan posisi robot dalam keadaan tegak lurus terhadap permukaan bumi tanpa memerlukan pengendali lain dari luar.

II.5. Mikrokontroler ATmega328

Mikrokontroler adalah sebuah sistem mikroprosesor dimana didalamnya sudah terdapat *CPU*, *Read Only Memory (ROM)*, *Random Access Memory (RAM)*, *Input-Output*, *timer*, *interrupt*, *Clock* dan peralatan *internal* lainnya yang sudah saling terhubung dan terorganisasi dengan baik dalam satu *chip* yang siap dipakai (Heri Susanto, et al. 2013 : 3).

Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler.

Mikrokontroler digunakan dalam produk dan alat yang dikendalikan secara otomatis, seperti sistem kontrol mesin, *remote controls*, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan dengan mendesain menggunakan mikroprosesor memori, dan alat *input output* yang terpisah, kehadiran mikrokontroler membuat kontrol elektrik untuk berbagai proses menjadi lebih ekonomis. Dengan penggunaan mikrokontroler ini maka :

1. Sistem elektronik akan menjadi lebih ringkas
2. Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi
3. Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak

Agar sebuah mikrokontroler dapat berfungsi, maka mikrokontroler tersebut memerlukan komponen eksternal yang kemudian disebut dengan sistem minimum. Untuk membuat sistem minimal paling tidak dibutuhkan sistem *clock* dan reset, walaupun pada beberapa mikrokontroler sudah menyediakan sistem *clock internal*, sehingga tanpa rangkaian eksternal pun mikrokontroler sudah beroperasi.

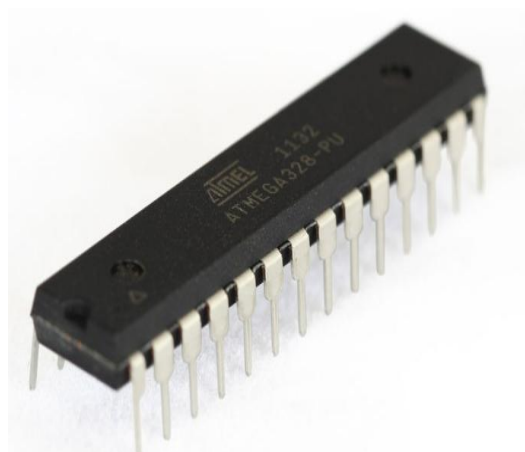
Yang dimaksud dengan sistem minimal adalah sebuah rangkaian mikrokontroler yang sudah dapat digunakan untuk menjalankan sebuah aplikasi. Sebuah IC mikrokontroler tidak akan berarti bila hanya berdiri sendiri. Pada dasarnya sebuah sistem minimal mikrokontroler AVR memiliki prinsip yang sama (Bernike Natalia Ginting, 2012 : 2).

II.5.1. Fitur AVR ATMega328

ATMega328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur *Reduce Instruction Set Computer (RISC)* dimana setiap proses eksekusi data lebih cepat dari pada arsitektur *Completed Instruction Set Computer (CISC)*. Mikrokontroler ini memiliki beberapa fitur antara lain (Menurut Baaret, 2013 : 3)

1. 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock*.
2. 32 x 8-bit *register* serba guna.

3. Kecepatan mencapai 16 *MIPS* dengan *clock* 16 MHz.
4. 32 KB *Flash memory* dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari *flash* memori sebagai *bootloader*.
5. Memiliki *EEPROM* (*Electrically Erasable Programmable Read Only Memory*) sebesar 1KB sebagai tempat penyimpanan data semi permanent.
6. Memiliki *SRAM* (*Static Random Access Memory*) sebesar 2KB.
7. Memiliki pin *I/O* digital sebanyak 14 pin 6 diantaranya *PWM* (*Pulse Width Modulation*) *output*.
8. *Master / Slave SPI Serial interface*.



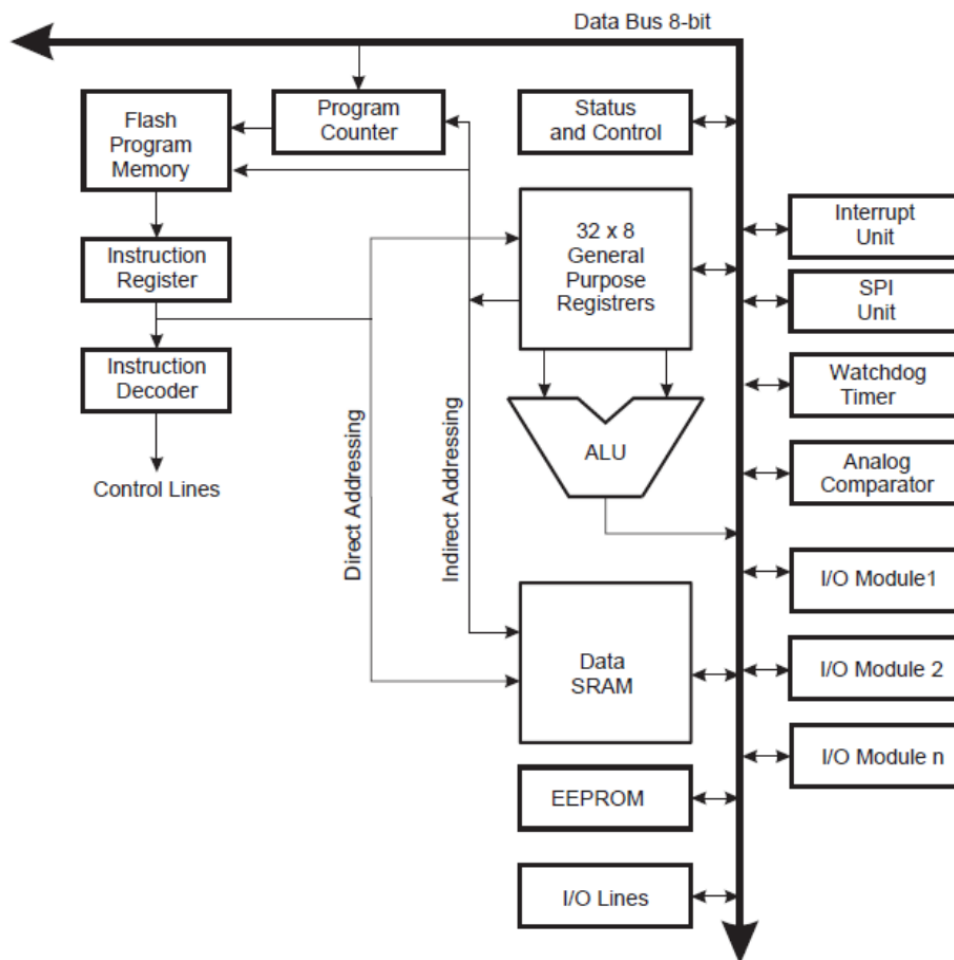
Gambar II.1. Mikrokontroler ATmega328

(Sumber: <http://www.protostack.com> dikunjungi pada tanggal 28 Juli 2016)

II.5.2. Arsitektur Mikrokontroler ATmega328

Seluruh mikrokontroler yang diimplementasikan pada produk Arduino menggunakan ATmega Keluaran AVR. Salah satunya, seri ATmega328 dengan sejumlah fitur di antaranya *On-Chip System Debug*, 5 ragam tidur (*Mode Sleep*), 6 saluran *ADC* yang mendukung reduksi *derau*, ragam hemat daya (*Power-save Mode*, *Power-down*), dan *Standby Mode*. (Jazia Eko Istiyanto, 2014 : 6).

Mikrokontroler ATmega328 memiliki arsitektur *Harvard*, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan kerja dan *parallelism*. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil dari memori program. Konsep inilah yang memungkinkan instruksi – instruksi dapat dieksekusi dalam setiap satu siklus *clock*. 32 x 8-bit *register* serba guna digunakan untuk mendukung operasi pada *ALU* (*Arithmetic Logic unit*) yang dapat dilakukan dalam satu siklus. 6 dari *register* serbaguna ini dapat digunakan sebagai 3 buah *register pointer* 16-bit pada mode pengalamatan tidak langsung untuk mengambil data pada ruang memori data. Ketiga *register pointer* 16-bit ini disebut dengan *register X* (gabungan R26 dan R27), *register Y* (gabungan R28 dan R29), dan *register Z* (gabungan R30 dan R31). Hampir semua instruksi AVR memiliki *format* 16-bit. Setiap alamat memori program terdiri dari instruksi 16-bit atau 32-bit. Selain *register* serba guna di atas, terdapat *register* lain yang terpetakan dengan teknik *memory mapped I/O* selebar 64 byte. Beberapa *register* ini digunakan untuk fungsi khusus antara lain sebagai *register control Timer/Counter*, Interupsi, *ADC*, *USART*, *SPI*, *EEPROM*, dan fungsi *I/O* lainnya. *Register – register* ini menempati memori pada alamat 0x20h – 0x5Fh. Berikut ini adalah tampilan *architecture* ATmega328 :



Gambar II.2. Tampilan Arsitektur ATmega328

(Sumber : Rizqi Ramadhan, 2012 : 5)

II.5.3. Konfigurasi Pin ATmega328

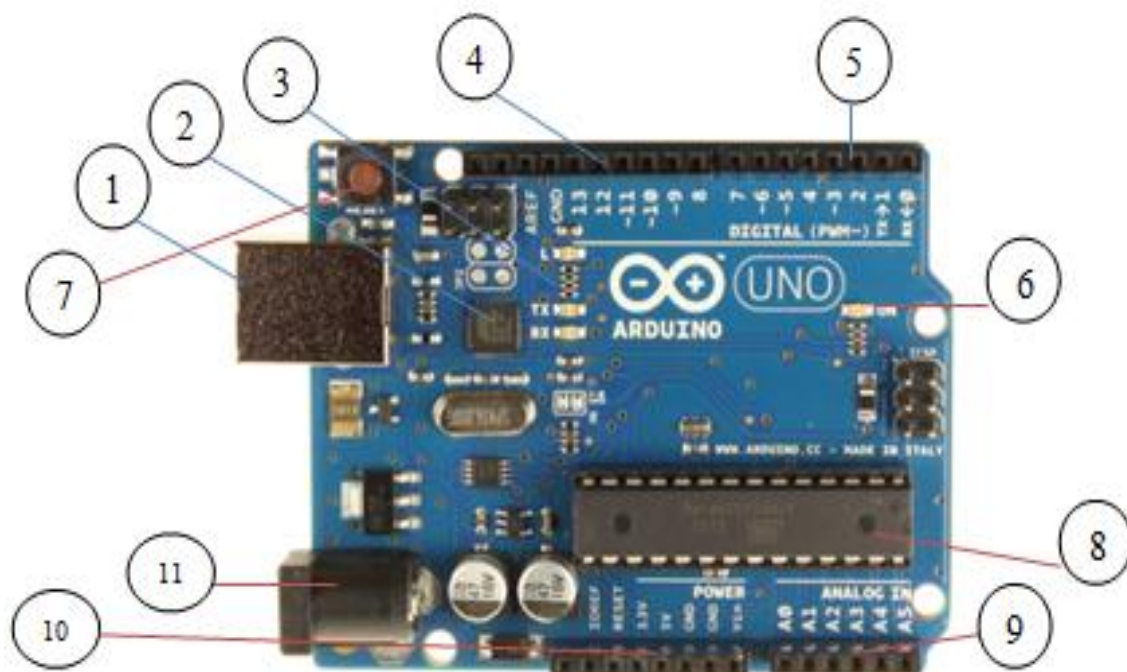
Masing-masing dari 14 pin digital arduino uno dapat digunakan sebagai masukan atau keluaran menggunakan fungsi `pinMode()`, `digitalWrite()` dan `digitalRead()`. Setiap pin beroperasi pada tegangan 5 volt. Setiap pin mampu menerima atau menghasilkan arus maksimum sebesar 40 mA dan memiliki resistor *pull-up internal* (diputus secara *default*) sebesar 20-30 KOhm. Sebagai tambahan, beberapa pin masukan digital memiliki kegunaan khusus yaitu (Jazi Eko Istiyanto, 2014 : 61):

1. Komunikasi serial: pin 0 (RX) dan pin 1 (TX), digunakan untuk menerima(RX) dan mengirim(TX) data secara serial.
2. External Interrupt: pin 2 dan pin 3, pin ini dapat dikonfigurasi untuk memicu sebuah interrupt pada nilai rendah, sisi naik atau turun, atau pada saat terjadi perubahan nilai.
3. *Pulse-width modulation (PWM)*: pin 3,5,6,9,10 dan 11, menyediakan keluaran *PWM* 8-bit dengan menggunakan fungsi *analogWrite()*.
4. *Serial Peripheral Interface (SPI)*: pin 10 (SS), 11 (MOSI), 12 (MISO) dan 13 (SCK), pin ini mendukung komunikasi SPI dengan menggunakan *SPI library*.
5. LED: pin 13, terdapat built-in LED yang terhubung ke pin digital 13. Ketika pin bernilai *HIGH* maka LED menyala, sebaliknya ketika pin bernilai *LOW* maka LED akan padam.

Arduino Uno memiliki 6 masukan analog yang diberi label A0 sampai A5, setiap pin menyediakan resolusi sebanyak 10 bit (1024 nilai yang berbeda). Secara *default* pin mengukur nilai tegangan dari *ground* (0V) hingga 5V, walaupun begitu dimungkinkan untuk mengganti nilai batas atas dengan menggunakan pin AREF dan fungsi *analogReference()*. Sebagai tambahan beberapa pin masukan analog memiliki fungsi khusus yaitu pin A4 (SDA) dan pin A5 (SCL) yang digunakan untuk komunikasi *Two Wire Interface (TWI)* atau *Inter Integrated Circuit (I2C)* dengan menggunakan *Wire library* yaitu (Jazi Eko Istiyanto, 2014 : 61).

II.6. Arduino Uno

Arduino Uno R3 adalah papan mikrokontroler berdasarkan ATMEGA8. Arduino ini memiliki 14 digital *input / output* pin (dimana 6 dapat digunakan sebagai *output* PWM), 6 input analog, *osillator kristal* 16 MHz, koneksi USB, *jack* DC, header ICSP, dan tombol reset. *Board* ini berisi semua yang diperlukan untuk mendukung mikrokontroler, cukup menghubungkannya ke komputer dengan kabel USB atau kekuasaan itu dengan adaptor AC-DC atau baterai untuk memulai. Arduino Uno R3 berbeda dari semua papan sebelumnya yang tidak menggunakan *chip* FTDI *driver* USB-to-serial. Sebaliknya, Arduino ini memiliki fitur Atmega16U2 (Atmega8U2 hingga versi R2) diprogram sebagai konverter USB-to-serial.



Gambar II.3. Minimum Sistem Arduino R3

Sumber : www.arduino.cc

Keterangan:

1. Port USB
2. IC Konverter Serial-USB (ATmega 8 U2)
3. Led untuk test output kaki D13
4. Kaki-kaki Input Output Digital (D8 – D13)
5. Kaki-kaki input Output Digital (D0 – D7)
6. LED Indikator catu daya
7. Tombol Reset
8. Mikrokontroler ATmega 8
9. Kaki-kaki input analog (A0 – A5)
10. Kaki-kaki catu daya (5V, GND)
11. Terminal Catudaya (6 – 9V)

II.7. Bluetooth

Bluetooth sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz *unlicensed ISM (Industrial, Scientific and Medical)* dengan menggunakan sebuah *frequency hopping tranceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara *host to host bluetooth* dengan jarak jangkauan layanan yang terbatas. *Bluetooth* sendiri dapat berupa *card* yang bentuk dan fungsinya hampir sama dengan card yang digunakan untuk *Wireless Local Area Network (WLAN)* dimana menggunakan frekuensi radio standar IEEE802.11, hanya saja pada *bluetooth* mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan *transfer* data yang lebih rendah.

Sistem *Bluetooth* terdiri dari sebuah *radio transceiver*, *baseband link Management dan Control*, *Baseband (processor core, SRAM, UART, PCM USB Interface)*, *flash* dan *voice code*. sebuah *link manager*. *Baseband link controller* menghubungkan perangkat keras radio ke *baseband processing* dan *layer protocol* fisik. *Link manager* melakukan aktivitas-aktivitas protokol tingkat tinggi seperti melakukan *link setup*, autentikasi dan konfigurasi. Bentuk fisik modul *Bluetooth HC-06* dapat dilihat pada gambar II.4 di bawah ini :



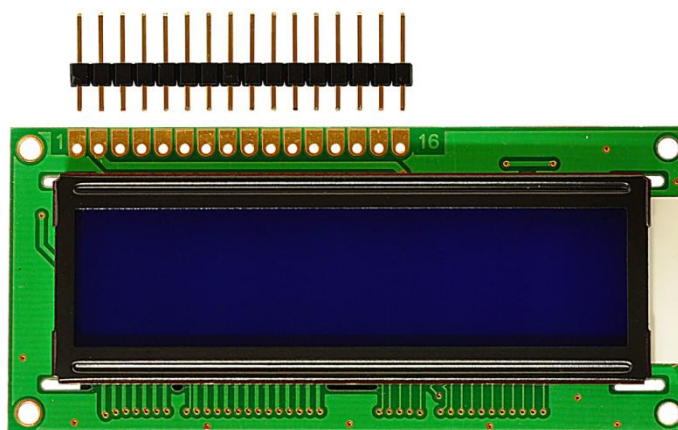
Gambar II.4. Bentuk Fisik Modul Bluetooth HC-06

Sumber : <http://www.botscience.net>

II.8. LCD 16 x 2

LCD (*Liquid Crystal Display*) merupakan suatu jenis media tampilan yang menggunakan *crystal* cair sebagai penampil utama. LCD adalah salah satu perangkat penampil yang sekarang ini mulai banyak digunakan. LCD memanfaatkan silikon atau galium dalam bentuk kristal cair sebagai pemancar cahaya. Pada layar LCD, setiap matrik adalah susunan dua dimensi piksel yang dibagi dalam baris dan kolom. Dengan demikian setiap pertemuan baris dan kolom adalah sebuah LED terdapat sebuah bidang latar (*backplane*), yang

merupakan lempengan kaca bagian belakang dengan sisi dalam yang ditutupi oleh lapisan elektroda transparan. Dalam keadaan normal, cairan yang digunakan memiliki warna cerah. Daerah-daerah tertentu pada cairan akan berubah warnanya menjadi hitam ketika tegangan diterapkan antara bidang latar dan pola elektroda yang terdapat pada sisi dalam lempeng kaca bagian depan. Bentuk fisik LCD 16x2 dapat dilihat pada gambar II.5



Gambar II.5. Bentuk Fisik LCD 16x2

Sumber : 20 Aplikasi mikrokontroler ATmega8535 & ATmega8535 menggunakan Bascom-AVR, Afrie Setiawan

LCD yang digunakan adalah jenis LCD yang menampilkan data dengan 2 baris tampilan pada *display*. Keuntungan dari LCD ini adalah

1. Dapat menampilkan karakter ASCII, sehingga dapat memudahkan dalam pembuatan program tampilan.
2. Mudah dihubungkan dengan *port I/O* karena hanya menggunakan 8 bit data.
3. Ukuran modul yang proporsional.
4. Daya yang digunakan relatif sangat kecil.

II.9. MPU6050

MPU6050 adalah *chip IC invense* yang di dalamnya terdapat sensor *accelerometer* dan *Gyroscope* yang sudah terintegrasi. Alasan menggunakan sensor ini adalah karena harganya relatif murah dimana sudah mendapatkan 2 sensor yang sudah terintegrasi.



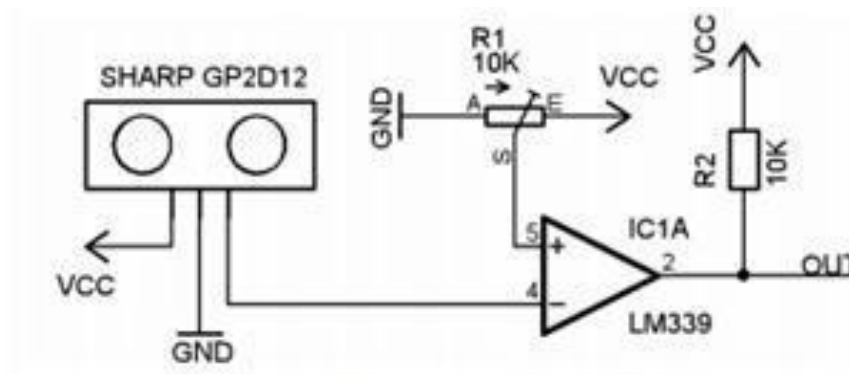
Gambar II.6. Modul GY521 MPU-6050
Sumber : *Datasheet* GY521 MPU-6050

Berikut adalah spesifikasi dari Modul GY521 MPU-6050 ini :

1. Berbasis *Chip* MPU-6050
2. *Supply* tegangan berkisar 3-5V
3. *Gyroscope range* + 250 500 1000 2000 ° / s
4. *Acceleration range*: $\pm 2 \pm 4 \pm 8 \pm 16$ g
5. *Communication standard* I2C
6. *Chip built-in* 16 bit AD *converter*, 16 bits data output
7. Jarak antar pin *header* 2.54 mm
8. Dimensi modul 20.3mm x 15.6mm

II.10. Sensor Sharp GP2D12

Sensor sharp GP2D12 digunakan untuk membaca jarak. Sensor ini menggunakan prinsip pantulan sinar infra merah. Dalam aplikasi ini nilai tegangan keluran dari sensor yang berbanding terbalik dengan hasil pembacaan jarak dikomparasi dengan tegangan referensi komparator. Rangkaian sistem komparator pembacaan jarak dengan sensor sharp GP2D12 ini disajikan pada gambar berikut:



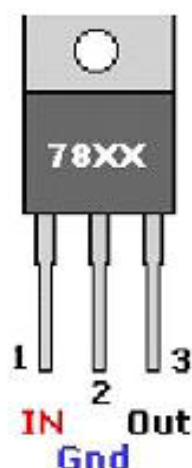
Gambar II.7. Sensor Sharp GP2D12
Sumber : *Datasheet* Sensor Sharp GP2D12

Prinsip kerja dari rangkaian komparator sensor sharp GP2D12 pada Gambar diatas adalah jika sensor mengeluarkan tegangan melebihi tegangan referensi, maka keluaran dari komparator akan berlogika rendah. Jika tegangan

referensi lebih besar dari tegangan sensor maka keluaran dari komparator akan berlogika tinggi. Selain menggunakan komparator, untuk mengakses sensor jarak sharp GP2D12 dapat dengan menggunakan prinsip ADC, atau dengan kata lain mengolah sinyal analog dari pembacaan sensor sharp GP2D12 ke bentuk digital dengan bantuan pemrograman. Untuk mengakses sensor ini dapat menggunakan fasilitas akses ADC.

II.11. Regulator Tegangan

Regulator seri 7805 adalah *regulator* untuk mendapatkan tegangan keluaran sebesar +5 volt, sedangkan *regulator* seri 7812 adalah untuk mendapatkan tegangan keluaran sebesar +12 volt. Agar rangkaian *regulator* dengan IC tersebut dapat bekerja dengan baik, tegangan *input* harus lebih besar dari tegangan *output* *regulator*-nya. Bentuk Fisik dari *regulator* 78xx dapat dilihat pada gambar II.8. (Fredy Indra Oktaviansyah: 2011).



Gambar II.8. Bentuk Fisik dari Regulator 78xx

Sumber : Datasheet LM 78xx

II.12. Bahasa Pemrograman

II.12.1. Bahasa Pemrograman C

Struktur penulisan bahasa C secara umum terdiri atas empat blok, yaitu:

1. *Header*,
2. Deklarasi konstanta global dan atau variabel,
3. Fungsi dan atau prosedur (bisa di bawah program utama),
4. Program utama.

Secara umum, pemrograman C paling sederhana dilakukan dengan hanya menuliskan program utamanya saja, yaitu:

```

/* fungsi utama */
void main()
{
    Statemen-statemen;
}
/* fungsi-fungsi lain yang ditulis oleh pemrogram komputer */
Fungsi_fungsi_lain()
{
    Statemen-statemen;
}

```

(M. Ary Heryanto, ST dan Ir. Wisnu Adi P. (2008: 8))

II.12.2. Pemrograman *IDE* Arduino)

Arduino memiliki open-source yang memudahkan untuk menulis kode dan meng-upload board ke arduino. Arduino *IDE* (*Integrated Development Enviroment*) ini merupakan media yang digunakan untuk memberikan informasi kepada arduino sehingga dapat memberikan output sesuai dengan apa yang diinginkan. Aplikasi arduino IDE ini dapat dijalankan di windows, Mac OS X, dan linux (Moh. Kamalul Wafi, 2014: 2). berikut merupakan gambaran tampilan arduino IDE :

```

Blink | Arduino 1.0
File Edit Sketch Tools Help

Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000); // wait for a second
}

1 Arduino Uno on /dev/ttyACM1

```

Gambar II.9. Tampilan Arduino IDE

Sumber: arduino.stackexchange.com dikunjungi pada tanggal 29 Juli 2016

Dalam arduino terhubung dengan arduino *IDE* ini dengan hanya menekan tombol RESET. tombol ini dirancang untuk menjalankan program yang telah di upload ke arduino uno, tombol ini juga terhubung dengan ATMEga 328 melalui kapasitor 100nf. *IDE (Integrated Development Enviroment)* arduino merupakan pemograman dengan menggunakan bahasa C. Setiap program *IDE* arduino (biasa disebut *sketch*) mempunyai dua buah fungsi yang harus ada, yaitu :

a. `void setup() { }`

Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program *IDE* Arduino dijalankan untuk pertama kalinya.

b. *void loop()* { }

Fungsi ini akan dijalankan setelah setup (fungsi *void setup*) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (power) dilepaskan.

Compiler merupakan modul yang berfungsi mengubah bahasa *processing* (kode pemrograman) kedalam kode biner karena kode biner adalah satu-satunya bahasa program yang dipahami oleh mikrokontroler. Sedangkan *upload* program adalah modul yang berfungsi memasukkan kode biner kedalam mikrokontroler. Pada *software* Arduino *IDE* memiliki fitur *compiler* sedangkan untuk *upload* program menggunakan *USBisp* yang dihubungkan ke port *ISP* pada papan rangkaian mikrokontroler. Pada proses ini akan merubah bahasa pemrograman dari digital ke bahasa analog yang dapat dipahami mikrokontroler (Anandya Bagus Venesa dan Wibowo Basuki Dwi, 2004 : 5).

II.12.3. Android

Android adalah sistem operasi untuk perangkat selular yang berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi.

Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

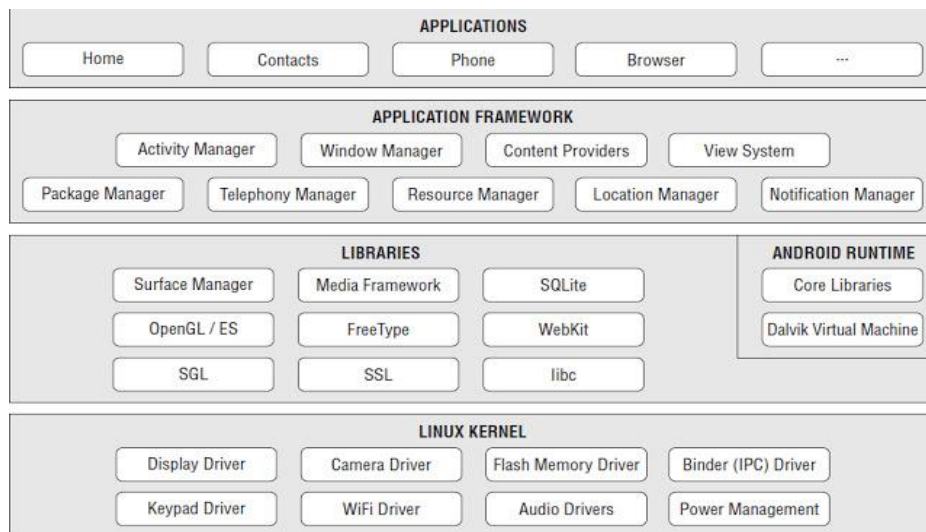
II.12.3.1. Fitur – fitur Android

Android tersedia secara open source bagi manufaktur perangkat keras untuk modifikasinya sesuai kebutuhan. Meskipun konfigurasi perangkat Android tidak sama antara satu perangkat dengan perangkat lain. Fitur yang tersedia Android adalah :

1. Penyimpanan (*Storage*) : menggunakan SQLite yang merupakan database *relational* yang ringan untuk menyimpan data.
2. Koneksi (*Connectivity*) : mendukung *GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth* (termasuk *A2DP* dan *AVRCP*), *Wifi, LTE, dan WiMax*.
3. Pesan (*Messaging*) : mendukung SMS dan MMS.
4. Mendukung Media : audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
5. *Web Browser* : menggunakan *open source WebKit* termasuk di dalamnya *engine Chrome V8 JavaScript*.
6. *Hardware* : terdapat *Accelerometer Sensor, Camera, Digital Compass, Proximity Sensor* dan *GPS*.
7. *Multi touch* : mendukung layar *multi touch*.
8. *Multi tasking* : mendukung aplikasi *multi tasking*.

II.12.3.2. Arsitektur Android

Agar lebih mudah memahami bagaimana Android bekerja, berikut ini bagan tingkatan – tingkatan sistem operasi Android :



Gambar II.10. Arsitektur Sistem Operasi Android

Sumber : *Pemograman Aplikasi Android*

Secara garis besar system operasi Android terbagi menjadi lima tingkatan :

1. **Applications** pada tingkat inilah kita akan bekerja, contoh aplikasi ini banyak ditemui, seperti : *Phone, Contack, Browse* dan lain – lain. Seperti aplikasi Android pada umumnya yang dapat di *download* dan di *install* dari Market Android. Semua aplikasi yang anda buat terleteak pada tingkat *Applications*.
2. **Application Framework** adalah semacam kumpulan class built-in yang tertanam dalam sistem operasi Android sehingga pengembang dapat memanfaatkannya untuk aplikasi yang sedang dibangun.
3. **Libraries** berisi semua kode program yang menyediakan layanan – layanan utama sistem operasi Android. Sebagai contoh library SQLite yang menyediakan dukungan database sehingga aplikasi Android dapat menggunakannya untuk menyimpan data. Library WebKit yang menyediakan fungsi – fungsi *browsing web*, dan lain – lain.

4. **Android Runtime** kedudukannya setingkat dengan libraries, Android *Runtime* menyediakan kumpulan pustaka inti yang dapat diaktifkan oleh pengembang untuk menulis kode aplikasi Android dengan bahasa pemrograman Java. *Dalvik Virtual Machine* aktif setiap kali aplikasi Android berproses (aplikasi Android dikompilasi menjadi *Dalvik executable*). *Dalvik* adalah mesin semu yang dirancang khusus untuk Android yang dapat mengoptimalkan daya *battery* perangkat bergerak dengan memori dan *CPU* terbatas.
5. **Linux kernel** adalah kernel dasar Android. Tingkat ini berisi semua *driver* perangkat tingkat rendah untuk komponen – komponen *hardware* perangkat Android.

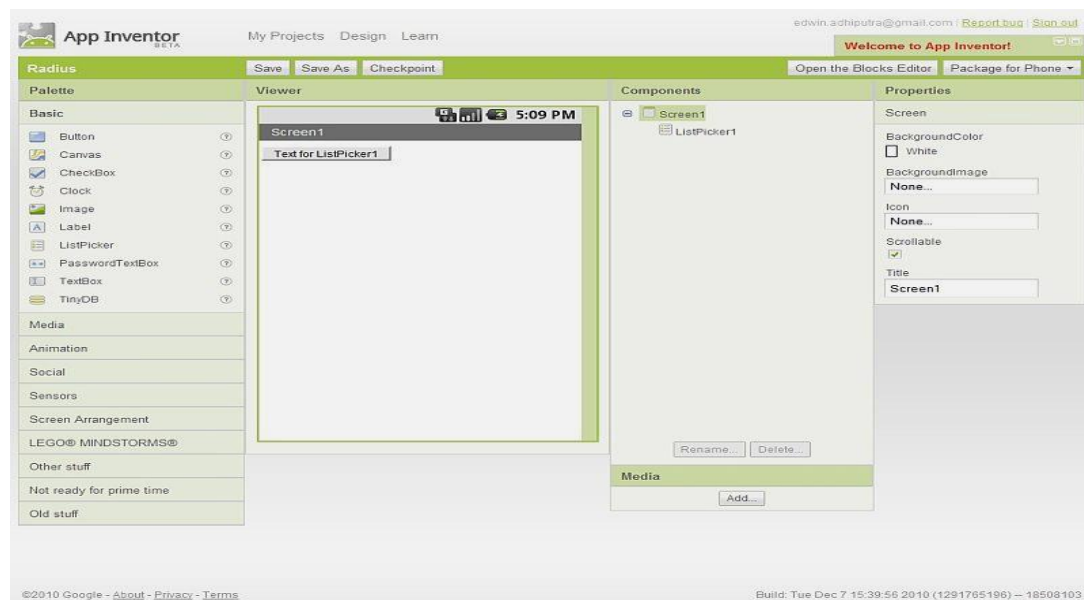
II.12.4. App Inventor

App Inventor for Android adalah aplikasi yang pada dasarnya disediakan oleh Google dan sekarang di-*maintenance* oleh Massachusetts Institute of Technology (MIT). App Inventor memungkinkan semua orang untuk membuat *software* aplikasi untuk sistem operasi android. Pengguna dapat menggunakan tampilan grafis GUI dan fitur drag and drop visual objek untuk membuat sebuah aplikasi dapat berjalan pada sistem operasi Android.

App Inventor adalah sebuah pemrograman visual yang digunakan untuk mengembangkan aplikasi berbasis Android dengan dukungan fitur berupa drag and drop tool. Anda dapat mendesain user *interface* dari sebuah aplikasi dengan menggunakan web GUI (*Graphical User Interface*) *builder*, kemudian anda dapat

menspesifikasikan behavior aplikasi dengan memasang *block* yang sesuai dengan kebutuhan anda.

App Inventor menggunakan *Kawa Language Framework* dan *Kawa's dialect* yang dikembangkan oleh Per Botner. Kedua aplikasi tersebut didistribusikan sebagai bagian dari GNU Operating System oleh *Free Software Foundation*. Kedua aplikasi tersebut dijadikan sebagai compiler dan menerjemahkan *Visual Block Programming* untuk di implementasikan pada *platform* Android.



Gambar II.11. Tampilan Awal App Inventor

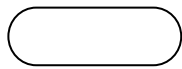





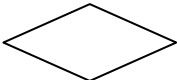


Sumber : Wahana Komputer : 2013 : 3

II.13. Flowchart

Prinsip kerja dari pemodelan di atas dapat digambarkan melalui *flowchart*. *Flowchart* adalah gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol dan dengan demikian setiap simbol menggambarkan proses tertentu. Hubungan antar

proses digambarkan dengan garis penghubung (Zarlis et al, 2007). Flowchart disusun dengan simbol-simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Adapun tabel simbol *flowchart* pada Tabel II.1:

Tabel II.1. Tabel Simbol Flowchart

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inialisasi/pemberian harga awal
	PROSES	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Sumber : Dr. Suarga, M. sc., M. Math. , Ph D. , 2012.