**BAB II**

**LANDASAN TEORI**

1. **Pengertian Rancang Bangun**

“Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisis dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan” (Pressman, 2012).

“Rancang sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru” (Mc Leod, 2005). “Perancangan adalah kegiatan yang memiliki tujuan dan mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik” (Ladjamudin, 2005). Sedangkan pengertian “bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun menggantikan atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian” (Pressman, 2012). Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisis ke dalam bentuk perangkat lunak kemudian menciptakan sebuah sistem baru ataupun memperbaiki sistem yang sudah ada baik secara keseluruhan maupun sebagian.

1. **Pengertian Sistem**

Suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. “Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen yang lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi didalam sistem tersebut” (Tata Sutabri, 2012).

* 1. **Mikrokontroler**

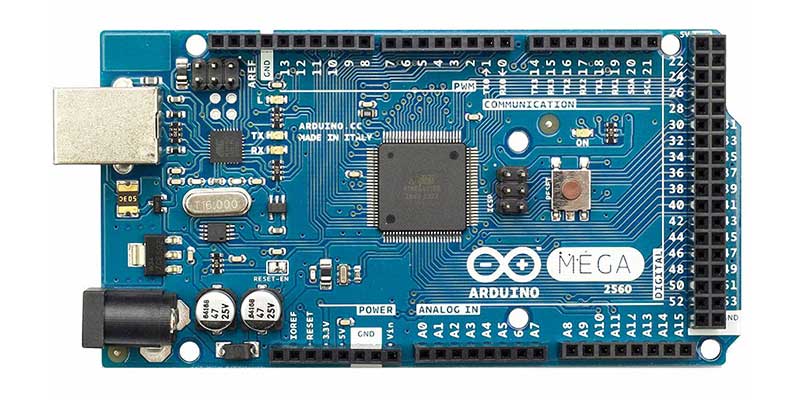
Menurut Dian Artanto (2008) dalam bukunya “*Mikrokontroller* merupakan sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip* IC sehingga sering juga disebut *single chip microcomputer*”.

* + 1. **Mikrokontroler Arduino**

**“**Arduino adalah sebuah pengendali *micro board* tunggal yang memiliki sifat terbuka (*open source*) yang diturunkan dari platform berbasis *wiring*” (Feri Djuandi,2011: 2). Pengendali ini dirancang untuk mempermudah penggunaan dalam berbagai bidang elektronik. *Hardware* arduino mengandung prosesor jenis Atmel AVR, dan memiliki bahasa pemrograman tersendiri. IDE adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam *memory microcontroller*. Ada banyak projek dan alat - alat yang dikembangkan oleh akademisi dan profesional dengan menggunakan arduino, selain itu juga ada banyak modul-modul pendukung yang dibuat oleh pihak lain untuk bisa disambungkan dengan Arduino. Arduino berevolusi menjadi sebuah platform karena arduino menjadi pilihan dan acuan bagi banyak praktisi. Salah satu yang membuat arduino banyak digunakan adalah karena sifatnya *open source*, baik untuk *hardware* maupun *software*-nya. Komponen utama didalam papan arduino adalah sebuah *microcontroller* 8 bit dengan merk Atmega yang dibuat oleh perusahaan Atmel Corporation.

* 1. **Arduino Mega**
     1. **Arduino ATMega 2560**

Dalam melakukan penelitian ini penulis menggunakan Arduino ATMega 2560 sebagai mikrokontroller. Arduino ATMega 2560 adalah sebuah papan mikrokontroler berbasis ATMega 2560 *(datasheet).* Mempunyai 54 pin digital *input/output* (dimana 14 pun dapat diguanakan sebagai keluaran PWM), 16 pin input analog, 2 UARTs (*Hardware serial ports*), sebuah *crystal oscillator* 16 MHz, sebuah penghubung USB, sebuah colokan listrik, ICSP *header*, dan tombol kembali. Setiap isi dari arduino ATMega 2560 membutuhkan dukungan mikrokontroler, koneksi mudah antara asrduino ATMega 2560 ke komputer dengan sebuah kabel USB atau daya dengan AC to DC adaptor atau baterai untuk memulai.

****

**Gambar 2.1** Arduino Mega 2560

(Sumber : [www.arduino.cc](http://www.arduino.cc))

Arduino Mega 2560 memiliki jumlah pin terbanyak dari semua papan pengembangan Arduino. Mega 2560 memiliki 54 buah digital pin yang dapat digunakan sebagai input atau output, dengan menggunakan fungsi pinMode(), digitalWrite(), dan digital(Read). Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus sebesar 20mA, dan memiliki tahanan pull-up sekitar 20-50k ohm (secara default dalam posisi disconnect). Nilai maximum adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan chip mikrokontroller.Beberapa pin memiliki fungsi khusus :

* **Serial**, memiliki 4 serial yang masing-masing terdiri dari 2 pin. Serial 0 : pin 0 (RX) dan pin 1 (TX). Serial 1 : pin 19 (RX) dan pin 18 (TX). Serial 2 : pin 17 (RX) dan pin 16 (TX). Serial 3 : pin 15 (RX) dan pin 14 (TX). RX digunakan untuk menerima dan TX untuk transmit data serial TTL. Pin 0 dan pin 1 adalah pin yang digunakan oleh chip USB-to-TTL ATmega16U2
* **External Interrups**, yaitu pin 2 (untuk interrupt 0), pin 3 (interrupt 1), pin 18 (interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2). Dengan demikian Arduino Mega 2560 memiliki jumlah interrupt yang cukup melimpah : 6 buah. Gunakan fungsi attachInterrupt() untuk mengatur interrupt.
* **PWM**: Pin 2 hingga 13 dan 44 hingga 46, yang menyediakan output PWM 8-bit dengan menggunakan fungsi analogWrite()
* **SPI** : Pin 50 (MISO), 51 (MOSI), 52 (SCK), dan 53 (SS) mendukung komunikasi SPI dengan menggunakan SPI Library
* **LED** : Pin 13. Pada pin 13 terhubung built-in led yang dikendalikan oleh digital pin no 13. Set HIGH untuk menyalakan led, LOW untuk memadamkan nya.
* **TWI** : Pin 20 (SDA) dan pin 21 (SCL) yang mendukung komunikasi TWI dengan menggunakan Wire Library

Arduino Mega 2560 R3 memiliki 16 buah input analog. Masing-masing pin analog tersebut memiliki resolusi 10 bits (jadi bisa memiliki 1024 nilai). Secara default, pin-pin tersebut diukur dari ground ke 5V, namun bisa juga menggunakan pin AREF dengan menggunakan fungsi analogReference(). Beberapa in lainnya pada board ini adalah :

* **AREF** : Sebagai referensi tegangan untuk input analog.
* **Reset** : Hubungkan ke LOW untuk melakukan reset terhadap mikrokontroller. Sama dengan penggunaan tombol reset yang tersedia.

Arduino Mega juga memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, berkomunikasi dengan Arduino lainnya, atau dengan mikrokontroller lain nya. Chip Atmega2560 menyediakan komunikasi serial UART TTL (5V) yang tersedia di pin 0 (RX) dan pin 1 (TX). Chip ATmega16U2 yang terdapat pada board berfungsi menterjemahkan bentuk komunikasi ini melalui USB dan akan tampil sebagai Virtual Port di komputer. Firmware 16U2 menggunakan driver USB standar sehingga tidak membutuhkan driver tambahan.

Chip ATmega2560 juga mendukung komunikasi I2C (TWI) dan SPI. Di dalam Arduino Software (IDE) sudah termasuk Wire Library untuk memudahkan anda menggunakan bus I2C. Untuk menggunakan komunikasi SPI, gunakan SPI library.

* + 1. **Arsitektur Arduino Mega 2560**

Arduino Mega 2560 terbentuk dari prosessor yang dikenal dengan Mikrokontroler ATMega 2560 yang memiliki beberapa fitur/spesifikasi yang menjadikannya sebagai solusi pengendali yang efektif untuk berbagai keperluan. Berikut spesifikasi Arduino Mega 2560 yang ditunjukan pada tabel 2.1 :

**Tabel 2.1** spesifikasi Arduino Mega 2560

|  |  |
| --- | --- |
| **Nama** | **Spesifikasi** |
| Tegangan Operasi | 5 V |
| Tegangan Input | 7-12 V |
| Pin Digital I/O | 54 pin (14 pin merupakan keluaran dari PWM) |
| Pin Input Analog | 16 pin |
| Arus DC pin I/O | 14 mA |
| Arus DC pin 3.3V | 50 mA |
| Flash Memory | 256 Kb (8 Kb digunakan untuk bootloader) |
| SRAM | 8 Kb |
| EEPROM | 4 Kb |
| Clock Speed | 16 MHz |

1. **Pengertian Sensor**

Sensor dan transducer merupakan peralatan atau komponen yang mempunyai peranan penting dalam sebuah sistem pengaturan otomatis. Ketepatan dan kesesuaian dalam memilih sensor akan sangat menentukan kinerja dari sistem pengaturan secara otomatis. Secara umum berdasarkan fungsi dan penggunaannya sensor dapat dikelompokan menjadi 3 bagian yaitu :

1. Sensor *thermal* (panas)
2. Sensor mekanis
3. Sensor *optic* (cahaya)

Sensor thermal adalah sensor yang digunakan untuk mendeteksi gejala perubahan panas/temperature/suhu pada suatu dimensi benda atau dimensi ruang tertentu. Contohnya; bimetal, termistor, termokopel, RTD, photo transistor, photo dioda, photo multiplier, photovoltaik, infrared pyrometer, hygrometer, dsb.

Sensor mekanis adalah sensor yang mendeteksi perubahan gerak mekanis, seperti perpindahan atau pergeseran atau posisi, gerak lurus dan melingkar, tekanan, aliran, level dsb. Contoh; strain gage, linear variable deferential transformer (LVDT), proximity, potensiometer, load cell, bourdon tube, dsb.

Sensor optic atau cahaya adalah sensor yang mendeteksi perubahan cahaya dari sumber cahaya, pantulan cahaya ataupun bias cahaya yang mengernai benda atau ruangan. Contoh; photo cell, photo transistor, photo diode, photo voltaic, photo multiplier, pyrometer optic, dsb.

Dengan definisi seperti ini maka sensor merupakan alat elektronik yang banyak dipakai dalam kehidupan manusia saat ini. Hampir seluruh kehidupan sehari-hari saat ini tidak ada yang tidak melibatkan sensor. Tidak mengherankan jika sensor banyak disebut sebagai panca indera-nya alat elektronik (William de Cooper, 2004).

* + 1. **Sensor Ultrasonik**

Gelombang ultrasonik merupakan gelombang yang umum digunakan untuk radar untuk mendeteksi keberadaan suatu benda dengan memperkirakan jarak antara sensor dan benda tersebut. Dalam penelitian ini, penulis menggunakan sensor ultrasonik tipe HC-SR04. Sensor ultasonik dapat dilihat pada Gambar 2.2.

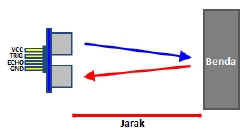
  
**Gambar 2.2** Sensor ultrasonik

(Sumber : www.arduino.cc)

Sensor ultrasonik adalah sebuah sensor yang berfungsi untuk mengubah besaran *fisis* (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik). Gelombang ultrasonik adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi ultrasonik tidak dapat didengar oleh telinga manusia. Bunyi ultrasonik dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi ultrasonik bisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi ultrasonik di permukaan zat padat hampir sama dengan reflektivitas bunyi ultrasonik di permukaan zat cair. Akan tetapi, gelombang bunyi ultrasonik akan diserap oleh tekstil dan busa.

* + - 1. **Cara Kerja Sensor Ultasonik**

Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan *piezoelektrik* dengan frekuensi tertentu. *Piezoelektrik* ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi 40kHz) ketika sebuah *osilator* diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima (Gambar 2.3).



**Gambar 2.3** Cara Kerja Sensor Ultrasonik

(Sumber : http://www.elangsakti.com)

Karena kecepatan bunyi adalah 340 m/s, maka rumus untuk mencari jarak berdasarkan ultrasonik adalah :

Dari rumus diatas, S merupakan jarak antara sensor ultrasonik dengan benda (bidang pantul), dan t adalah selisih antara waktu pemancaran gelombang oleh *transmitter* dan waktu ketika gelombang pantul diterima *receiver*.

HC-SR04 merupakan sensor ultrasonik siap pakai, satu alat yang berfungsi sebagai pengirim, penerima, dan pengontrol gelombang ultrasonik. Alat ini bisa digunakan untuk mengukur jarak benda dari 2cm - 4m dengan akurasi 3mm. Dengan demikian, untuk menghitung jarak yang hanya maksimal 4 m maka rumus di atas harus dimodifikasi atau disesuaikan satuannya.

Mikrokontroler bisa bekerja pada order *mikrosekon* (1s = 1.000.000 µs) dan satuan jarak bisa kita ubah ke satuan cm (1m = 100 cm). Oleh sebab itu, rumus di atas bisa diperbaharui menjadi:

Sensor ultasonik HC-SR04 memiliki 4 pin, pin Vcc, Gnd, Trigger, dan Echo. Pin Vcc untuk listrik positif dan Gnd untuk ground-nya. Pin Trigger untuk trigger keluarnya sinyal dari sensor dan pin Echo untuk menangkap sinyal pantul dari benda.

* + 1. **Sensor PIR (*Passive Infra Red*)**

Sensor PIR (*Passive Infra Red*) adalah sensor yang digunakan untuk mendeteksi adanya pancaran sinar infra merah. Sensor PIR bersifat pasif, artinya sensor ini tidak memancarkan sinar infra merah tetapi hanya menerima radiasi sinar infra merah dari luar.

[](http://4.bp.blogspot.com/-cEp8HXVoAEs/Th5pWtiB2XI/AAAAAAAAAE0/Rjb3Skf6qpw/s1600/PIR.jpg)

**Gambar 2.4** Sensor PIR (*Passive Infra Red*)

(Sumber : www.arduino.cc)

Sensor ini biasanya digunakan dalam perancangan detektor gerakan berbasis PIR. Karena semua benda memancarkan energi radiasi, sebuah gerakan akan terdeteksi ketika sumber infra merah dengan suhu tertentu (misal: manusia) melewati sumber infra merah yang lain dengan suhu yang berbeda (misal: dinding), maka sensor akan membandingkan pancaran infra merah yang diterima setiap satuan waktu, sehingga jika ada pergerakan maka akan terjadi perubahan pembacaan pada sensor.Sensor PIR terdiri dari beberapa bagian yaitu *Fresnel Lens, IR Filter, Pyroelectric sensor, amplifier*, dan *comparator*.

1.  *Fresnel Lens*

Lensa Fresnel pertama kali digunakan pada tahun 1980an. Digunakan sebagai lensa yang memfokuskan sinar pada lampu mercusuar. Penggunaan paling luas pada lensa Fresnel adalah pada lampu depan mobil, di mana mereka membiarkan berkas parallel secara kasar dari pemantul parabola dibentuk untuk memenuhi persyaratan pola sorotan utama. Namun kini, lensa Fresnel pada mobil telah ditiadakan diganti dengan lensa plain polikarbonat. Lensa Fresnel juga berguna dalam pembuatan film, tidak hanya karena kemampuannya untuk memfokuskan sinar terang, tetapi juga karena intensitas cahaya yang relative konstan diseluruh lebar berkas cahaya.

2. IR *Filter*

IR Filter dimodul sensor PIR ini mampu menyaring panjang gelombang sinar infrared pasif antara 8 sampai 14 mikrometer, sehingga panjang gelombang yang dihasilkan dari tubuh manusia yang berkisar antara 9 sampai 10 mikrometer ini saja yang dapat dideteksi oleh sensor. Sehingga Sensor PIR hanya bereaksi pada tubuh manusia saja.

3.  *Pyroelectric sensor*

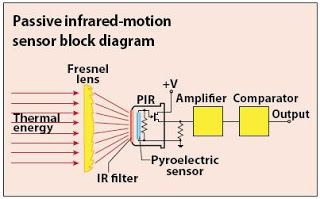
Seperti tubuh manusia yang memiliki suhu tubuh kira-kira 32 derajat celcius, yang merupakan suhu panas yang khas yang terdapat pada lingkungan. Pancaran sinar inframerah inilah yang kemudian ditangkap oleh *Pyroelectric sensor* yang merupakan inti dari sensor PIR ini sehingga menyebabkan *Pyroelectic sensor* yang terdiri dari *galium nitrida, caesium nitrat*dan *litium tantalate*menghasilkan arus listrik. Mengapa bisa menghasilkan arus listrik? Karena pancaran sinar inframerah pasif ini membawa energi panas. Material *pyroelectric* bereaksi menghasilkan arus listrik karena adanya *energy* panas yang dibawa oleh infrared pasif tersebut. Prosesnya hampir sama seperti arus listrik yang terbentuk ketika sinar matahari mengenai *solar cell*.

4. Amplifier

Sebuah sirkuit amplifier yang ada menguatkan arus yang masuk pada material pyroelectric.

5. Comparator

Setelah dikuatkan oleh amplifier kemudian arus dibandingkan oleh *comparator* sehingga mengahasilkan output.

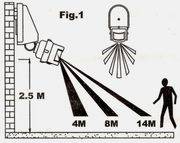
[](http://2.bp.blogspot.com/-ErpjAtPP-XI/Th5pSHkLAJI/AAAAAAAAAEo/tBDHGSTKjeY/s1600/cara-kerja-pir.jpg)

**Gambar 2.5** Block Diagram Sensor PIR

(Sumber : http://www.sainsdanteknologiku.com/)

* + - 1. **Cara kerja pembacaan sensor PIR**

Pancaran infra merah masuk melalui lensa Fresnel dan mengenai sensor pyroelektrik, karena sinar infra merah mengandung energi panas maka sensor pyroelektrik akan menghasilkan arus listrik. Sensor pyroelektrik terbuat dari bahan gallium nitride (GaN), cesium nitrat (CsNo3) dan litium tantalate (LiTaO3). Arus listrik inilah yang akan menimbulkan tegangan dan dibaca secara analog oleh sensor. Kemudian sinyal ini akan dikuatkan oleh penguat dan dibandingkan oleh komparator dengan tegangan referensi tertentu (keluaran berupa sinyal 1-bit). Jadi sensor PIR hanya akan mengeluarkan logika 0 dan 1, 0 saat sensor tidak mendeteksi adanya pancaran infra merah dan 1 saat sensor mendeteksi infra merah. Sensor PIR didesain dan dirancang hanya mendeteksi pancaran infra merah dengan panjang gelombang 8-14 mikrometer. Diluar panjang gelombang tersebut sensor tidak akan mendeteksinya. Untuk manusia sendiri memiliki suhu badan yang dapat menghasilkan pancaran infra merah dengan panjang gelombang antara 9-10 mikrometer (nilai standar 9,4 mikrometer), panjang gelombang tersebut dapat terdeteksi oleh sensor PIR. (Secara umum sensor PIR memang dirancang untuk mendeteksi manusia). Sensor PIR memiliki jangkauan jarak yang bervariasi, tergantung karakteristik sensor. Proses penginderaan sensor PIR dapat digambarkan sebagai berikut:

[](http://2.bp.blogspot.com/-APpU8BedqT4/Th5pUMjRF_I/AAAAAAAAAEs/KJ4uHJH29ik/s1600/t_11103385_1.jpg)

**Gambar 2.6** Proses penginderaan sensor PIR

(Sumber : http://www.sainsdanteknologiku.com/)

Pada umumnya sensor PIR memiliki jangkauan pembacaan efektif hingga 5 meter, dan sensor ini sangat efektif digunakan sebagai *human detector*.

* + 1. **Switch Magnetik PE-905**

Switch magnetik merupakan saklar yang dapat merespon medan magnet yang berada disekitarnya. Magnetik switch ini seperti halnya sensor limit switch yang diberikan tambahan plat logam yang dapat merespon adanya magnet. (Heranudin, Rancang Bangun Sistem Keamanan Ruangan Menggunakan Radio Frequency Identifikasi (RFID) Berbasis Mikrokontroler AT89C51, 2008 hal: 5). Salah satu contoh switch magnetik dapat dilihat pada gambar 2.7.



**Gambar** **2.7** Switch Magnetik

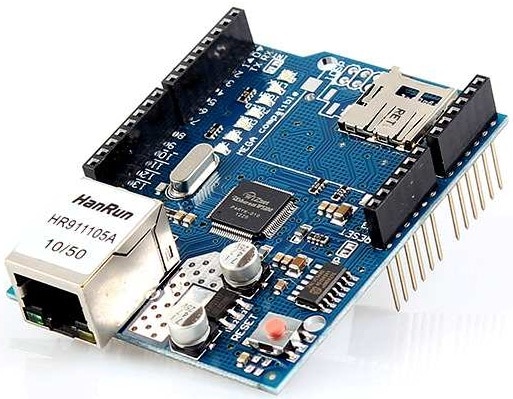
(Sumber: Jurnal Thesis Tri Putri Handayani 2015.)

Switch magnetik tersebut biasa digunakan untuk pengamanan pada pintu dan jendela. Dalam pemasangannya switch magnetik ini dapat dipasang dengan cara ditanam di bagian pintu atau hanya ditempelkan saja di jendela. Pemasangannya pun dapat dilakukan pada pintu atau jendela dengan berbagai bahan, dapat di pasang pada pintu atau jendela yang terbuat dari kayu atau dari logam, seperti aluminium. Spesifikasi switch magnetik, switch magnetic yang digunakan type PE-905, suatu perangkat pengendalian otomatis, sangat cocok untuk menggunakan di sirkuit sampai tegangan maksimal 4 Volt dan arus sampai 100mA sampai 500mA.

* + 1. **Modul Ethernet W5100**

Ethernet shield merupakan sebuah shield Ethernet mikrokontroler Arduino. Ethernet shield ini merupakan sarana pengembangan TCP/IP berbasi modul jaringan W5200 yang berfungsi sebagai jembatan antara mikrokontroler dengan jaringan internet atau Ethernet tanpa memerlukan bantuan komputer. Ethernet shield ini cocok untuk aplikasi-aplikasi embedded yang membutuhkan komunikasi dengan jaringan internet atau Ethernet, seperti serial to Ethernet converter, web server, smart house, dsb.

Ethernet shield adalah sebuah shield modul jaringan yang menyertakan chip w5200 (TCP/IP hardware chip), Ethernet PHY (IP101A), dan MAG-Jack (RJ45 dengan X’FMR). Modul ini merupakan pilihan tepat dan murah jika ingin membangun system berbasis internet, Gambar dibawah ini menampilkan modul Ethernet shield.



**Gambar** **2.7** Switch Magnetik (www.arduino.cc)

Untuk menghubungkan ethernet shield dengan komputer, hub, atau router gunakan kabel ethernet standar (CAT5 atau CAT6 dengan konektor RJ45). Board Arduino berkomunikasi dengan W5100 dan SD card menggunakan bus SPI (melalui ICSP header). Bus ini terwakili oleh pin 11, 12, dan 13. Pin 10 digunakan untuk mengaktifkan chip W5100. Ada beberapa LED yang digunakan sebagai indikator pada Ethernet Shield ini, yaitu :

1. PWR : mengindikasikan ada tidaknya tegangan yang mensupply shield

2. LINK : mengindikasikan koneksi jaringan, dan berkedip saat lalu-lintas data

3. FULLD : mengindikasikan bahwa koneksi jaringan merupakan full duplex

4. 100M : mengindikasikan koneksi jaringan 100 Mb/s

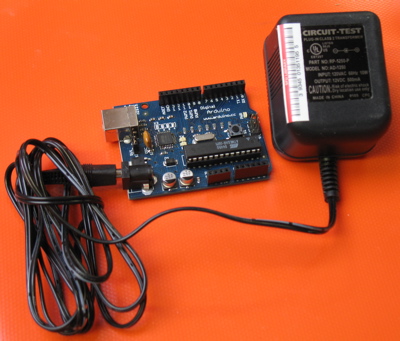
5. RX : berkedip ketika shield menerima data

6. TX : berkedip ketika shield mengirim data

7. COLL : berkedip ketika network collisions terdeteksi

* 1. **Catu Daya**

Secara umum, istilah “catu daya” biasanya berarti suatu sistem penyearah-filter (*rectifier-filter*) yang mengubah AC menjadi DC murni. Banyak rangkaian catu daya yang berlainan yang dapat digunakan untuk pekerjaan tersebut. Komponen dasar yang digunakan untuk rangkaian yang lebih sederhana adalah *transformator*, penyearah, *resistor*, *kapasitor*, dan *induktor*. Catu yang diatur secara lebih kompleks dapat menambahkan *transistor* atau *trioda* sebagai pengindera-tegangan dan pengontrolan tegangan, ditambah dengan *dioda zener* atau tabung VR untuk menyediakan tegangan acuan (*reference*).

****

**Gambar 2.9** Catu Daya   
(Sumber : www.playground.arduino.cc)

* 1. ***Buzzer***

*Buzzer* adalah sebuah komponen elektronika yang dapat mengubah sinyal listrik menjadi getaran suara. Pada umumnya, *Buzzer* yang merupakan sebuah perangkat audio ini sering digunakan pada rangkaian anti-maling, Alarm pada Jam Tangan, Bel Rumah, peringatan mundur pada Truk dan perangkat peringatan bahaya lainnya. Jenis *Buzzer* yang sering ditemukan dan digunakan adalah *Buzzer* yang berjenis Piezoelectric, hal ini dikarenakan Buzzer Piezoelectric memiliki berbagai kelebihan seperti lebih murah, relatif lebih ringan dan lebih mudah dalam menggabungkannya ke Rangkaian Elektronika lainnya. Buzzer yang termasuk dalam keluarga Transduser ini juga sering disebut dengan Beeper.

Efek Piezoelectric (Piezoelectric Effect) pertama kali ditemukan oleh dua orang fisikawan Perancis yang bernama Pierre Curie dan Jacques Curie pada tahun 1880. Penemuan tersebut kemudian dikembangkan oleh sebuah perusahaan Jepang menjadi Piezo Electric Buzzer dan mulai populer digunakan sejak 1970-an.

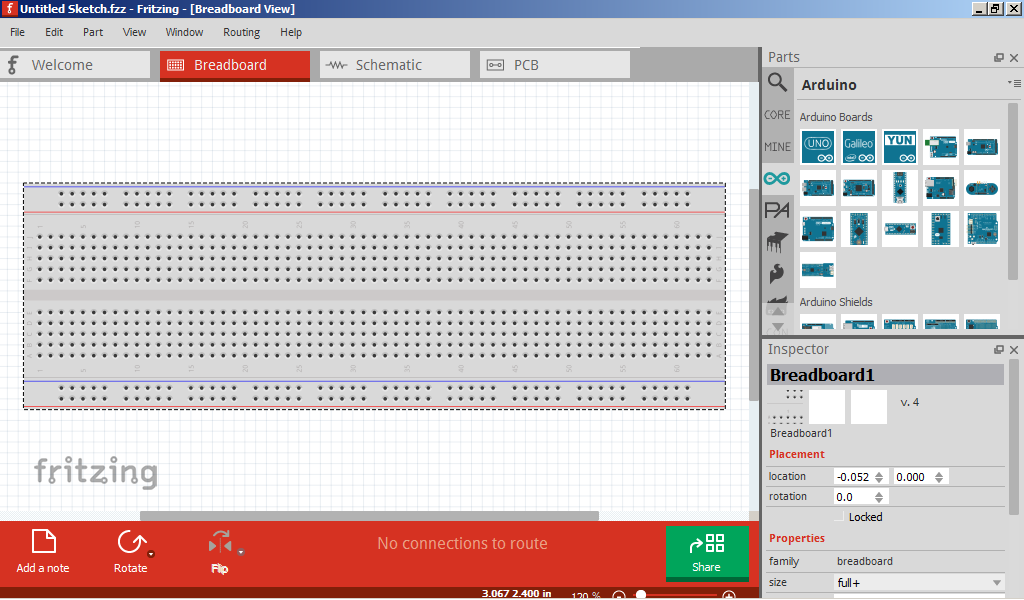


**Gambar 2.10** *Buzzer*

(Sumber : http://www.teknikelektronika.com)

* 1. ***Breadboard***

*Breadboard* adalah *board* yang digunakan unutuk membuat rangkaian elektronika sementara dengan tujuan uji coba atau *prototiype* tanpa harus menyolder. Dengan memanfaatkan *breadboard,* komponen-komponen elektronika yang dipakai dapat digunakan kembali untuk membuat rangkaian yang lain. *Breadboard* umumnya terbuat dari bahan plastik dengan banyak lubang-lubang diatasnya. Lubang-lubang *breadboard* diatur sedemikian rupa membentuk pola sesuai dengan pola jaringan koneksi di dalamnya. *Breadboard* dapat dilihat pada gambar 2.11.



**Gambar 2.11** *Breadboard*

(Sumber : http://www.rustamaji.net)

* 1. **Kabel *Jumper***

Kabel *jumper* adalah suatu istilah kabel yang berdiameter kecil, didalam dunia elektronik digunakan untuk menghubungkan dua titik atau lebih dan dapat juga untuk menghubungkan dua komponen atau lebih.

* + 1. **Jenis-jenis Kabel *Jumper***

Ada beberapa jenis kabel *jumper* yang dibedakan berdasarkan konektor kabelnya, yaitu :

1. *Male-male*

Kabel *jumper* jenis ini digunakan untuk koneksi *male to male* pada kedua ujung kabelnya.



**Gambar 2.12** Kabel Jumper *Male to Male*

(Sumber : http://www.robotedukasi.com)

1. *Male-female*

Kabel *jumper* jenis ini digunakan untuk koneksi *male to female* pada kedua ujung kabelnya. Diaman *male* merupakan konektor *plug* dan *female* merupakan konektor *socket*

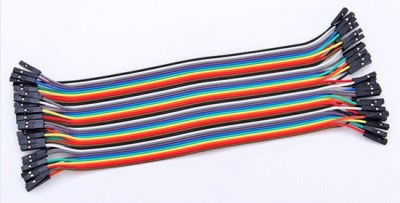


**Gambar 2.13** Kabel Jumper *Male to female*

(Sumber : http://www.robotedukasi.com)

1. *Female-female*

Kabel *jumper* jenis ini digunakan untuk koneksi *famale to famale* pada kedua ujung kabelnya



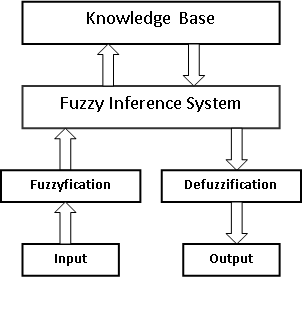
**Gambar 2.14** Kabel Jumer *Female to female*

(Sumber : http://www.robotedukasi.com)

* 1. **Metode *Fuzzy Logic***

Logika fuzzy diperkenalkan pertama kali pada tahun 1965 oleh Prof Lutfi A. Zadeh seorang peneliti di Universitas California di Barkley dalam bidang ilmu komputer. Professor Zadeh beranggapan logika benar salah tidak dapat mewakili setiap pemikiran manusia, kemudian dikembangkanlah logika fuzzy yang dapat mempresentasikan setiap keadaan atau mewakili pemikiran manusia. Perbedaan antara logika tegas dan logika fuzzy terletak pada keanggotaan elemen dalam suatu himpunan. Jika dalam logika tegas suatu elemen mempunyai dua pilihan yaitu terdapat dalam himpunan atau bernilai 1 yang berarti benar dan tidak pada himpunan atau bernilai 0 yang berarti salah. Sedangkan dalam logika fuzzy, keanggotaan elemen berada di interval [0,1].

Logika fuzzy memiliki beberapa komponen yang harus dipahami seperti himpunan fuzzy, fungsi keanggotaan, operator pada himpunan fuzzy, *inferensi fuzzy* dan *defuzzifikasi*. Berikut adalah gambar skema dasar dari *Fuzzy logic* :



**Gambar 2.15** Skema Dasar Metode *Fuzzy*

(Sumber : Rinaldi Munir 2007)

Berikut adalah penjelasan dari skema metode fuzzy :

1. *Input fuzzy* berupa bilangan *crisp* (tegas) yang dinyatakan dalam himpunan input.
2. *Fuzzifikasi* merupakan proses untuk mengubah bilangan crisp menjadi nilai keanggotaan dalam himpunan *fuzzy*.
3. *Fuzzy inference system* merupakan bagian pengambilan kesimpulan (*reasoning*) dan keputusan.
4. *Knowledge base* berisi aturan-aturan yang biasanya dinyatakan dengan perintah **IF …. THEN….**
5. *Defuzzification* merupakan proses untuk merubah nilai *output* fuzzy menjadi nilai *crisp*.

Contoh berikut akan menjelaskan bagaimana konsep “jarak” yang digolongkan “jauh” dalam pengertian fuzzy/samar dan *crisp* (tegas). Misalnya diberikan suatu definisi bahwa setiap jarak kurang dari 50 cm dikatakan “dekat” dan jarak 100 cm atau lebih adalah “jauh”.

0

1

10 20 30 40 50 60 70 80 90 100 110

Derajat   
Keanggotaan

µ[x]

(Jarak)

**Gambar 2.16** Konsep “Jauh” Dalam Pengertian Tegas (Crisp)

(Sumber : Rinaldi Munir 2007)

Dalam pengertian crisp (tegas), batas-batas antara “dekat” dan “jauh” sangat jelas, setiap jarak (50,…,90) adalah tidak “jauh” (Gambar 2.16). Tidak ada derajat jarak, sedangkan dalam fuzzy setiap anggota memiliki nilai berdasarkan pada derajat keanggotaan. Adapun konsep “jarak” yang digolongkan “jauh” dalam pengertian fuzzy :

0

1

80 90 100 110 120 130

0.7

Derajat   
Keanggotaan

µ[x]

(Jarak)

**Gambar 2.17** Konsep “Jauh” Dalam Pengertian Fuzzy

(Sumber : Rinaldi Munir 2007)

Gambar 2.17 memperlihatkan bahwa anggota jarak yang berjarak 90 cm, derajat keanggotaannya bernilai 0.7, sedangkan anggota yang berjarak 100 cm derajat keanggotaannya bernilai 1, sedangkan anggota yang bejarak kurang dari 100 cm memiliki derajat yang berlainan. Derajat keanggotaan ini, menunjukkan seberapa dekat nilai tiap-tiap jarak dalam anggota himpunan itu dalam konsep “jauh”. Dapat dikatakan bahwa anggota yang berjarak 90 cm adalah 70% (0.7) mendekati “jauh”, atau dengan bahasa alami “agak jauh”.

Dalam metode fuzzy ada beberapa komponen diantaranya sebagai berikut :

1. Himpunan Fuzzy

Tahun 1965, Profesor L.A. Zadeh memperkenalkan teori himpunan fuzzy, yang secara tidak langsung mengisyaratkan bahwa tidak hanya teori probabilitas saja yang dapat merepresentasikan ketidakpastian. Teori himpunan fuzzy adalah merupakan perluasan dari teori logika Boolean yang menyatakan tingkat angka 1 atau 0 atau pernyataan benar atau salah, sedangkan pada teori logika fuzzy terdapata nilai keangggotaan suatu *item* x dalam suatu himpunan A, yang sering ditulis dengan µA[x], memiliki 2 kemungkinan  yaitu : satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan. Dari variabel yang telah dimunculkan, kemudian disusunlah domain himpunan fuzzy.

1. Fungsi Keanggotaan (*Membership Function)*

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Ada Berbagai macam jenis pendekatan fungsi yang dapat digunakan, antara lain *representative linear*, *representative* kurva lonceng, *representative* kurva normal, *representative* kurva bahu, dan *representative* kurva segitiga. Dalam penyelesaian masalah dimakalah ini, kami menggunakan *representative* kurva segitiga dalam penentuan fungsi keanggotaan.

1. Aplikasi Fungsi Implikasi

Tahap ini merupakan penentuan aturan dari sistem logika kabur. Aturan-aturan dapat dibentuk untuk menyatakan relasi antara input dan output. Tiap aturan merupakan suatu implikasi. Operator yang digunakan untuk menghubungkan antara dua input adalah operator *AND,* dan yang memetakan antara *input-output* adalah *IF THEN.*

1. Komposisi Aturan

Komposisi Aturan yang kami gunakan adalah Metode MIN-MAX dengan menggunakan operator AND. Disini operator AND berhubungan dengan operasi interseksi pada himpunan. Solusi humpunan Fuzzy sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

1. Penegasan *(Defuzzycation)*

*Input* dari proses *defuzzifikasi* adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut. Sehingga jika diberikan suatu himpunan fuzzy dalam range tertentu, maka harus dapat diambil suatu nilai *crsip* tertentu sebagai output.

Dalam proses penyelesaian ini perlu adanya peran  *Fuzzy Expert Systems* yang pada kesempatan ini menggunakan metode Tsukamoto yaitu setiap konsekuen pada aturan yang berbentuk *IF-Then* harus direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil *inferensi* dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan α-predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

* 1. **Teori Perancangan Perangkat Lunak**
     1. **Definisi *Flowchart***

*Flowchart* adalah bagan - bagan yang mempunyai arus yang menggambarkan langkah - langkah penyelesaian suatu masalah. Flowchart merupakan cara penyajian dari suatu algoritma (Ladjamuddin B, Al-Bahra bin. 2006). Ada dua macam flowchart yang menggambarkan proses dengan computer, yaitu :

1. System flowchart, bagan yang memperlihatkan urutan proses dalam sistem dengan menunjukan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data.
2. Bagan yang memperlihatkan urutan instruksi yang digambarkan dengan simbol tertentu memecahkan masalah salam suatu program.

Flowchart disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol - simbol yang digunakan dalam menyusun flowchart, yaitu sebagai berikut :

**Tabel 2.2** Simbol-Simbol Pada *Flowchart*

(Ladjamuddin B, Al-Bahra bin. 2006)

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Fungsi** |
|  | Terminator | Sebagai symbol ‘START’ atau ‘END’ untuk memulai dan mengakhiri flowchart. |
|  | Arrow | Sebagai penunjuk arah dan alur proses |
|  | Preparation | Digunakan untuk memberikan nilai awal |
|  | Proses | Digunakan untuk menuliskan proses yang diperlukan, misalnya operasi aritmatika. |
|  | Input / output data | Digunakan untuk menuliskan proses menerima data atau mengeluarkan data. |
|  | Conditional / Decision | Digunakan untuk menyatakan proses yang membutuhkan keputusan. |
|  | On Page Connector | Digunakan untuk menyatukan beberapa arrow. |
|  | Off Page Connector | Digunakanuntuk menghubungkan flowchart yang harus digambarkan pada halaman yang berbeda. |
|  | Display | Digunakan untuk menampilkan data ke monitor. |

* + 1. ***Unified Modelling Language* (UML)**
       1. **Definisi *Unified Modelling Language* (UML)**

*“Unified Modelling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal., yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemroograman berorientasi objek” (Salahudin M, Rosa A.S. 2011). UML diaplikasikan untuk maksud tertentu, biasanya untuk :

1. Merancang perangkat lunak
2. Sarana komunikasi antara perangakat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

Blok pembangun utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis*timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambar-kan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dalam mengaplikasikan beragam sistem. Intinya, UML merupakan alat komunikasi yang konsisten dalam mendukung para pengembang sistem saat ini. Sebagai perancang sistem, pasti akan menjumpai uml, baik kita sendiri yang membuat atau sekedar membaca diagram uml buatan orang lain.

* + - 1. **Komponen *Unified Modelling Language* (UML)**

UML menyediakan beberapa diagram visual yang menunjukan berbagai aspek dalam sistem. Diagram mengemukakan banyak hal, penggunaan notasi yang teridefinisi baik dan ekpresif adalah penting pada proses pengembangan perangkat lunak. Ada beberapa diagram yang disediakan dalam UML, antara lain :

1. *Use CaseDiagram*

*“Use Case Diagram* adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* diagram mendeskripsikan interaksi tipikal antara pengguna sistem dengan sistem itu sendiri, dengan memberikan sebuah narasi tentang bagaimana sistem tersebut digunakan. *Use Case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem yang ditekankan adalah apa yang diperbuat sistem bukan bagaimana” (Salahudin M, Rosa A.S. 2011).

*Use Case* Diagram menggambarkan *external view* dari sistem yang akan kita buat modelnya. Komponen *Use Case* Diagram dapat kita lihat pada tabel 2.3

**Tabel 2.3** Simbol – Simbol Pada *Use Case Diagram*

( Sumber : Salahudin M, Rosa A.S. 2011)

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama Simbol** | **Keterangan** |
|  | *Actor* | Sebagai subjek yang berinteraksi dengan system |
|  | *Use Case* | Sebagai kegiatan yang dapat dilakukan oleh pengguna pada sistem. |
|  | *Association* | Sebagai penghubung anatar actor dengan use case. |
|  | *Include* | Sebagai penghubung antara *use case* yang membutuhkan *use case* lain. |
|  | *System* | Sebagai cakupan wilayah sistem. |

1. *Activity Diagram*

“*Activity Diagram* adalah teknik untuk menggambarkan logika prosedural, proses bisnis dan jalur kerja. Dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antar diagram ini dan notasi diagram alir adalah diagram ini mendukung behavior paralel”. Komponen *ActivityDiagram* dapat kita lihat pada tabel 2.4 (Salahudin M, Rosa A.S. 2011)

**Tabel 2.4** Simbol – Simbol Pada *Activity Diagram*

(Sumber : Salahudin M, Rosa A.S. 2011)

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama Simbol** | **Keterangan** |
|  | *Swimlane* | Menunjukan siapa yang bertanggung jawab dalam melakukan aktivitas dalam suatu diagram. |
|  | *Activity* | Memperlihatkan bagaimana masing – masing kelas antarmuka saling berinteraksi satu sama lain |
|  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi. |
|  | *Initial State* | Menunjukan dimana aliran kerja di mulai. |
|  | *Activity Final Node* | Menunjukan dimana aliran kerja di akhiri. |
|  | *Decesion Node* | Menunjunkan suatu keputusan yang mempunyai satu atau lebih transisi dan dua atau lebih transisi sesuai dengan suatu kondisi. |
|  | *Control Flow* | Menunjukan bagaimana kendali suatu aktivitas terjadi pada aliran kerja dalam tindakan tertentu. |

1. *Class Diagram*

“*ClassDiagram* merupakan spesifikasi yang jika diinstasiasikan akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek” (Salahudin M, Rosa A.S. 2011).

**Tabel 2.5** Simbol Pada *Class Diagram*

(Sumber : Salahudin M, Rosa A.S. 2011)

|  |  |  |
| --- | --- | --- |
| Simbol | Nama Simbol | Keterangan |
|  | Class | Sebagai kelas yang di gunakan pada sistem. |
|  | Generalization | Menunjukan *inheritance* hubungan pada kelas |

1. *Sequence Diagram*

“Sebuah *Sequence* diagram secara khusus menjabarkan aktifitas sebuah skenario tunggal. Diagram tersebut menunjukan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek didalam usecase diagram. *Sequence* diagram menunjukan interaksi dengan menampilkan partisipan dengan garis alir secara vertikal dan pengurutan pesan dari atas kebawah” (Salahudin, M. Rosa A.S. 2011). Komponen *SequenceDiagram* dapat kita lihat pada tabel 2.6

**Table 2.6** Simbol - Simbol Pada *Sequence Diagram*

(Sumber : Salahudin M, Rosa A.S. 2011)

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama Simbol** | **Keterangan** |
|  | *Life Line* | Objek entity, antarmuka yang saling berinteraksi. |
|  | *Actor* | Menggambarkan user/pengguna |
| Message() | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi – informasi tentang aktifitas yang terjadi. |
|  | *Boundary* | Digunakan untuk menggambarkan sebuah form. |
|  | *Control Class* | Digunakan untuk menghubungkan boundary dengan table. |
|  | *Entity Class* | Digunakan untuk menggabarkan hubungan kegiatan yang akan dilakukan. |

* 1. **Teori Bahasa Pemrograman**
     1. **Bahasa Pemrograman C**

“Bahasa C adalah bahasa pemrograman yang dapat dikatakan berada diantara bahasa beraras rendah (bahasa yang berorientasi pada mesin) dan beraras tinggi (bahasa yang berorientasi pada manusia)” (Henry Febrian, 2016). Bahasa beraras rendah, misalnya bahasa assembler, bahasa ini ditulis dengan sandi yang dimengerti oleh mesin saja, oleh karena itu hanya digunakan bagi yang memprogram mikroprosesor. Bahasa beraras rendah merupakan bahasa yang membutuhkan kecermatan yang teliti bagi pemrogram karena perintahnya harus rinci, ditambah lagi masing-masing pabrik mempunyai sandi perintah sendiri. Bahasa tinggi relatif mudah digunakan, karena ditulis dengan bahasa manusia sehingga mudah dimengerti dan tidak tergantung mesinnya. Bahasa beraras tinggi biasanya digunakan pada komputer.

Pencipta bahasa C adalah Brian W. Kernighan dan Denis M. Ritchi, sekitar tahun 1972. Penulisan program dalam bahasa C dilakukan dengan membagi dalam blok-blok, sehingga bahasa C disebut dengan bahasa terstruktur. Bahasa C dapat digunakan di berbagai mesin dengan mudah, mulai dari PC sampai dengan mainframe, dengan berbagai sistem operasi misalnya DOS, UNIX, VMS dan lain-lain.

* + - 1. **Penulisan Bahasa Pemrograman C**

“Akar dari bahasa C adalah dari bahasa BCPL yang dikembangkan oleh Martin Richards pada tahun 1967” (Wirdasari, 2010). Bahasa C adalah bahasa yang standar, artinya suatu program ditulis dengan versi bahasa C tertentu akan dapat dikompilasi dengan versi bahasa C yang lain dengan sedikit modifikasi.

Program Bahasa C tidak mengenal aturan penulisan dikolom tertentu, jadi bisa dimulai dari kolom manapun. Namun demikian, untuk mempermudah pembacaan program dan untuk keperluan dokumentasi, sebaiknya penulisan bahasa C diatur sedemikian rupa sehingga mudah dan enak dibaca.

Berikut contoh penulisan Program Bahasa C:

#include <NewPing.h>

main ()

{

//isi program//

}

Program dalam bahasa C selalu berbentuk fungsi seperti ditunjukkan dalam **main ().** Program yang dijalankan berada di dalam tubuh program yang dimulai dengan tanda kurung buka **{** dan diakhiri dengan tanda kurung tutup **}**. Semua yang tertulis di dalam tubuh program ini disebut dengan blok.

Tanda () digunakan untuk mengapit argumensuatu fungsi. Argumen adalah suatu nilai yang akan digunakan dalam fungsi tersebut. Dalam fungsi **main** diatas tidak ada argumen, sehingga tak ada data dalam (). Dalam tubuh fungsi antara tanda **{** dan tanda **}** ada sejumlah pernyataan yang merupakan perintah yang harus dikerjakan oleh prosesor. Setiap pernyataan diakhiri dengan tanda titik koma

Baris pertama **#include <…>** bukanlah pernyataan, sehingga tak diakhiri dengan tanda titik koma (;). Baris tersebut meminta kompiler untuk menyertakan file yang namanya ada di antara tanda **<…>** dalam proses kompilasi. *File-file* ini (ber-ekstensi .h) berisi deklarasi fungsi ataupun *variable*. File ini disebut *header*. File ini digunakan semacam perpustakaan bagi pernyataan yang ada di tubuh program.

* + 1. **Bahasa Pemrograman Java**

Java adalah bahasa pemrograman serbaguna. Java dapat digunakan untuk membuat suatu program sebagaimana membuatnya seperti bahasa lain. Java juga mendukung sumberdaya internet yang sangat populer yaitu WEB, selian itu java juga mendukung aplikasi klien-server dalam jaringan lokal dan luas. (Abdul Kadir, 2003).

Java dikembangkan oleh sun microsystem pada agustus 1991 dengan nama semula *OAK***.** *OAK* adalah sejenis pohon jati yang terlihat dari jendela tempat pembuatnya, james Gosling bekerja. Ada yang mengatakan oak adalah singkatan dari “*Object Application Kernel”.* pada januari 1995 karena nama Oak dianggap kuranh komersial maka diganti dengan nama Java.

Dalam sejumlah literatur disebutkan bahwa java merupakan hasil perpaduan sifat dari sejumlah bahasa pemrograman yaitu C, C++, Object-C, SmallTalk, dan CommonLISP. Selain itu java juga dilengkapi dengan unsur keamanan. Yang tak kalah penting java menambahkan paradigma pemrograman yang sederhana. Jika anda telah mengenal C atau C++ yang mengandalkan pointer dan merasakan keruwetannya, java justru meninggalkan sehingga memperoleh kemudahaan java saat menggunakannya. Java memiliki kelebihan, sebagai berikut :

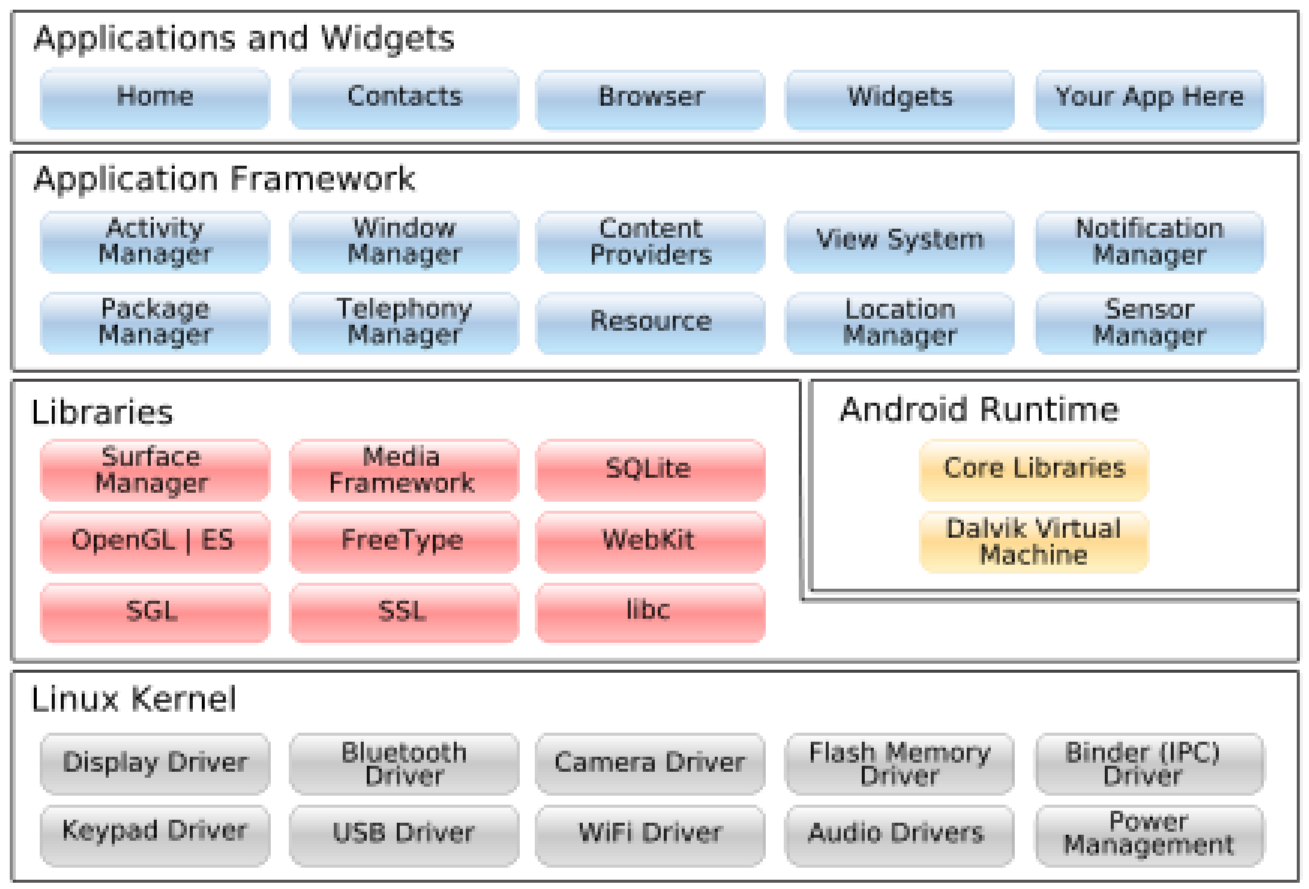
1. *Multiplatform*. Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform* / sistem operasi komputer, sesuai dengan prinsip *tulis sekali, jalankan di mana saja*. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows.
2. OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek) yang artinya semua aspek yang terdapat di Java adalah Objek. Java merupakan salah satu bahasa pemrograman berbasis oebjek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut *Object*. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir.
3. Perpustakaan Kelas Yang Lengkap, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya.
   * 1. **Android**
        1. **Sejarah Android**

Android adalah sebuah sistem operasi utuk perangkat mobile berbasis linux yang mencangkup sistem operasi, *middleware* dan aplikasi. Awalnya, Google Inc. membeli android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk *smartphone*. kemudian untuk mengembangkan android, dibentuklah *Open Handset Alliance* (OHA).

Sejak tahun 2008, Android secara bertahap telah melakukan sejumlah pembaruan untuk meningkatkan kinerja sistem operasi, menambahkan fitur baru, dan memperbaiki bugyang terdapat pada versi sebelumnya. Setiap versi utama yang dirilis dinamakan secara alfabetis berdasarkan nama-nama makanan pencuci mulut atau cemilan bergula; misalnya, versi 1.5 bernama Cupcake, yang kemudian diikuti oleh versi 1.6 Donut. Versi terbaru adalah 5.0 Lollipop, yang dirilis pada 15 Oktober 2014. (Nazruddin Safaat H. 2012).

* + - 1. **Arsitektur Android**

Secara garis besar arsitektur android dapat dijelaskan dan digambarkan sebagai berikut :



**Gambar 2.18** Arsitektur Sistem Operasi Android

(Nazruddin Safaat H.2012)

1. *Application and Widgets*

*Aplication* dan *widget* ini adalah *layer* dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Dilayer ini terdapat aplikasi inti termasuk email, program SMS, kalender, peta, browser, kontak, dan lain-lain

1. *Application Frameworks*

Android adalah “*Open Development Platform*” yaitu android menawarkan kepada pengembang untuk memeberi kemampuan kepada pengembang unuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *services* *background*, mengatur alarm dan menambahkan notifikasi, dan sebagainya.

1. *Libraries*

*Libraries* ini adalah *layer* dimana fitur fitur android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan diatas *kernel*, *layer* ini meliputi berbagai *library* c/c++ inti seperti libc dan SSL, serta :

1. *Libraries* media untuk pemutaran media audio dan video.
2. *Libraries* untuk manajemen tampilan.
3. *Libraries* mencangkup SGL dan OpenGL untuk grafis 2D dan 3D.
4. *Libraries* SQLite untuk dukungan database.
5. *Libraries* SSL dan *Webkit* terintegrasi dengan *web* *browser* dan *security*.
6. *Libraries* *LiveWebcore* mencangkup modern web browser dengan *engine embedded web view.*
7. *Android Run Time*

*Layer* yang membuat aplikasi android dapat berjalan dimana saja dalam prosesnya menggunakan implementasi linux. Didalam *Android* *Run* *Time* terbagi menjadi dua bagian yaitu :

1. *Core* *Libraries*: aplikasi yang dibangun dalam bahasa java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh core libraries.
2. *Dalvik Virtual Machine* : virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi fungsi secara efisien, untuk melakukan *threading* dan manajemen tingkat rendah.
3. *Linux Kernel*

*Linux kernel* adalah *layer* dimana inti dari *operating system* dari android itu berada. Berisi file-file sistem yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi android lainnya.

* 1. **Database**
     1. **Pengertian Database dan DBMS**

Sebelum membahas secara lebih mendalam mengenai *database* ada baiknya mengetahui terlebih dahulu pengertian *database* dan *DBMS* sebagai berikut :

1. *Database* merupakan sekumpulan data yang disusun secara logis dan dikendalikan secara sentral. *Database* memiliki bagian-bagian penting misalnya tabel yang digunakan untuk menyimpan data (Ema Utami, 2008)
2. *DBMS*  merupakan kepanjangan dari *Database Management System* yaitu perangkat lunak yang dirancang untuk dapat melakukan utilitas dan mengelola koleksi data dalam jumlah yang besar. *DBMS* juga dirancang untuk melakukan manipulasi data secara lebih mudah. Penyimpanan data oleh *DBMS* disesuaikan dengan bentuk model datanya, beberapa contoh *DBMS* adalah PostgreSQL, MySQL, DB2, Oracle, SQL Server dan lain-lain.
   * 1. **Pengertian SQL**

Dalam pemrograman database, salah satu pemrograman yang harus kita kuasai adalah SQL. SQLmerupakan bahasa komputer standar yang digunakan untuk berkomunikasi dengan sistem manajemen basis data relasional (RDBMS). SQLsering disebutkan sebagai kepanjangan dari *Structured Query Language* yang merujuk sejarah SQLyakni pertama kali dikembangkan oleh *IBM* dalam *SEQUEL-XRM* dan *System-R* Project (1974-1977). Dimana *SEQUEL* merupakan kepanjagan dari *Structured English Query Language)* yang akhirnya dihilangkan *English*nya menjadi SQL namun masih banyak orang yang melafalkan SQL karena sejarah tersebut (Ema Utami, 2008)

* 1. **Tool Perangkat Lunak**
     1. **Pengertian Eclipse**

Eclipse adalah sebuah software development environment multibahasa yang berfitur extensible plugin. Extensible plugin berarti pengembang dapat mengembangkan beberapa jenis aplikasi, misalkan, Java, C++ atau Python dalam satu aplikasi IDE dengan cara menambahkan plugin. Dan eclipse adalah *integration* *development* *environment* (IDE) yang digunakan untuk pengembangan Java/Android (Nazaruddin Safaat H. 2012).

* + 1. **SQLite**

Menurut Nazaruddin Safaat android memiliki fasilitas untuk membuat database yang dikenal dengan SQLite yaitu salah satu software yang embedded yang sangat popular, kombinasi SQL interface dan penggunaan memory yang sangat sedikit dengan kecepatan yang sangat cepat. SQLite di android termasuk dalam Android runtime, sehingga setiap versi dari android dapat membuat database dengan SQLite (Nazaruddin:2012).

Dalam sistem android terdapat beberapa teknik untuk melakukan penyimpanan data. Teknik yang umum digunakan adalah sebagai berikut:

1. Shared prefences yaitu menyimpan data beberapa nilai (value) dalam bentuk groups key yang dikenal dengan prefences.

2. Files yaitu menyimpan data dalam file, dapat berupa menulis ke file atau membaca dari file.

3. SQLite Database, yaitu menyimpan data dalam bentuk database.

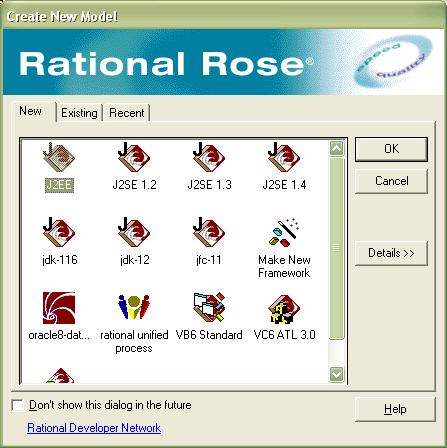
4. Content Providers, yaitu menyimpan data dalam bentuk content providers service.

* + 1. ***Rational Rose***

“*Rational rose* merupakan sebuah perangkat lunak pemodelan secara visual yang memiliki banyak kemampuan (*power—full*) untuk pembentukan sistem berorientasi objek yang menggunakan *Unified Modelling Language* (UML)” (Adi Nugaroho. 2009).

Istilah-istilah yang diguanakan dalam *rational rose*, yaitu :

1. *View* : menunjukan perbedaan dari berbagai aspek – aspek suatu sistem yang dimodelkan
2. Diagram : merupakan graph yang menjelaskan tentang isi dari sebuah *view*.
3. Elemen : elemen model konsep diguankan diagram merupakan elemen-elemen model yang menyatakan konsep-konsep berorientasi objek secara umum.

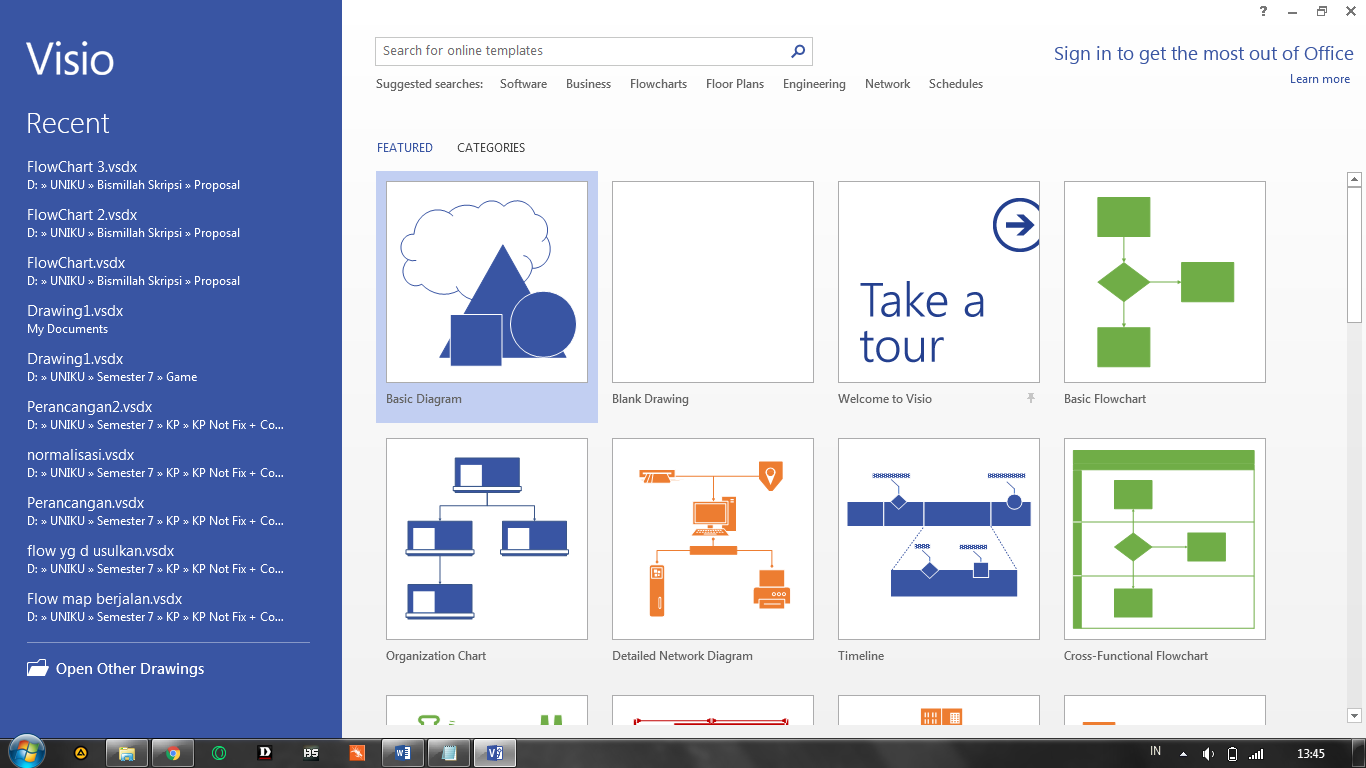


**Gambar 2.19** Tampilan Awal *Rational Rose* 2000

(Adi Nugroho. 2009)

* + 1. ***Microsoft Visio***

*Microsoft Visio* merupakan suatu program aplikasi computer yang didesain khusus untuk membantu dalam membuat diagram seperti, *Flowchart, Grant Chart, Data Flow,* Gambar Jaringan, Gambar Denah Bangunan, dan juga pembuatan Gambar Teknik, serta desain lainnya. (Chandra, Cato dan Ir. Teddy Marcus Zakaria, 2008) Berikut ini adalah Gambar 2.20 Aplikasi *Microsoft visio* yang penulis gunakan :

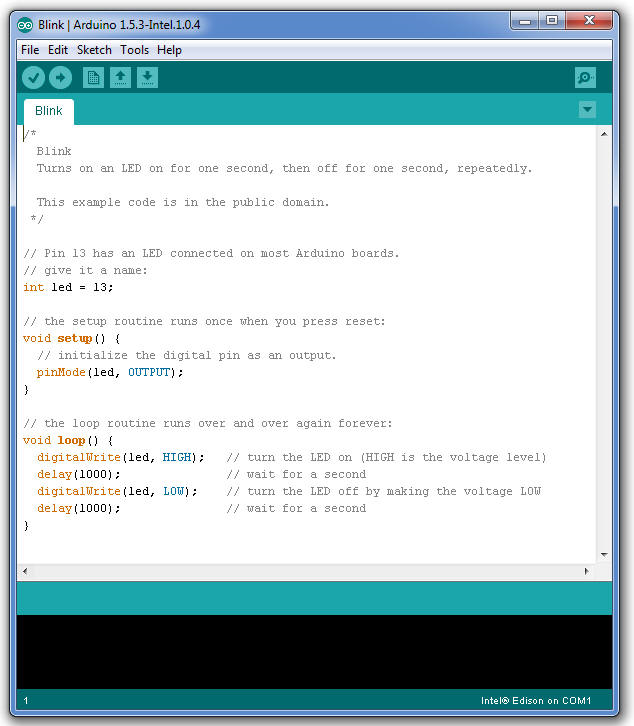


**Gambar 2.20** Tampilan Awal *Microsoft Visio* 2013

*Microsoft Visio* memiliki keunggulan disbanding program *word* atau *excel* dalam pembuatan diagram. *Microsoft Visio* memiliki banyak *template* yang di dalam terdapat *shape / connector.*

* + 1. **Arduino IDE**

IDE itu merupakan kependekan dari Integrated Developtment 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 dibenamkan 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. ([www.sinauarduino.com/](http://www.sinauarduino.com/))



**Gambar 2.21** Arduino IDE

(Sumber : www.sinauarduino.com)

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*](https://processing.org/) yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

* 1. **Pengujian Sistem** 
     1. **Blackbox Testing**

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Pengujian black box berusaha menemukan kesalahan dalam kategori sebagai berikut: fungsi-fungsi yang tidak benar atau hilang, (2) kesalahan interface, (3) kesalahan dalam struktur data atau akses databae eksternal, (4) kesalahan kinerja, (5) inisialisasi dan kesalahan terminasi (Roger S. Pressman, 2012).

Klasifikasi Black box testing mencakup beberapa pengujian, yaitu (Janner Simarmata, 2010) :

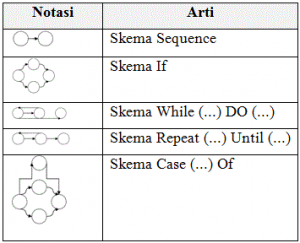
* + - 1. Pengujian fungsional (*functional testing*). Pada jenis pengujian ini, perangkat lunak diuji untuk persyaratan fungsional.
      2. Pengujian tegangan (*stress testing*). Pengujian ini berkaitan dengan kualitas aplikasi di dalam lingkungan. Ide nya adalah untuk menciptakan sebuah lingkungan yang lebih menuntut aplikasi, tidak seperti saat aplikasi dijalankan pada beban kerja normal.
      3. Pengujian beban (*load testing*). Pada pengujian ini, aplikasi akan diuji dengan beban berat atau masukan, seperti yang terjadi pada pengujian situs web, untuk mengetahui apakah aplikasi/situs gagal atau kinerjanya menurun.
      4. Pengujian khusus (*ad hoc testing*). Jenis pengujian ini dilakukan tanpa penciptaan rencana pengujian (test plan) atau kasus pengujian (test case). Pengujian khusus membantu dalam menentukan lingkup dan durasi dari berbaga pengujian lainnya dan juga membantu para penguji dalam mempelajari aplikasi sebelum memulai pengujian dengan pengujian lainnya.
      5. Pengujian penyelidikan (*exploratory testing*). Pengujian penyelidikan mirip dengan pengujian khusus dan dilakukan untuk mempelajari atau mencari aplikasi. Pengujian ini merupakan pendekatan yang menyenangkan untuk pengujian.
      6. Pengujian usabilitas (*usability testing*). Pengujian ini dilakukan jika antarmuka pengguna dari aplikasinya penting dan harus spesifik untuk jenis pengguna tertentu.
      7. Pengujian asap (*smoke testing*). Pengujian ini dilakukan untuk memeriksa apakah aplikasi tersebut sudah siap untuk pengujian yang lebih besar dan bekerja dengan baik tanpa cela sampai tingkay ang paling diharapkan.
      8. Pengujian pemulihan (*recovery testing*). Pengujian ini pada dasarnya dilakukan untuk memeriksa seberapa cepat dan baiknya aplikasi bias pulih terhadap semua jenis crash atau kegagalan hardware, masalah bencana, dan lain-lain.
      9. Pengujian volume (*volume testing*). Pengujian ini dilakukan terhadap efisiensi dari aplikasi. Jumlah data yang besar diproses melalui aplikasi (yang sedang diuji) untuk memeriksa keterbatasan ekstrem dari sistem.
      10. Pengujian domain (*domain testing*). Pengujian ini merupakan penjelasan yang paling sering menjelaskan teknik pengujian.
      11. Pengujian skenario (*scenario testing*). Pengujian ini adalah pengujian yang realistis, kredibel dan memotivasi stakeholder, tantangan untuk program dan mempermudah penguji untuk melakukan evaluasi.
      12. Pengujian regresi (*regression testing*). Pengujian ini adalah gaya pengujian yang berfokus pada pengujian ulang (*retesting*) setelah ada perubahan.
      13. Penerimaan pengguna (*user acceptance*). Pada pengujian ini, perangkat lunak akan diserahkan kepada pengguna untuk mengetahui apakah perangkat lunak memenuhi harapan pengguna dan bekerja seperti yang diharapkan.
      14. Pengujian alfa (*alpha testing*). Pada pengujian ini pengguna akan diundang ke pusat pengembangan. Pengguna akan menggunakan aplikasi dan pengembangan mencatat setiap masukan atau tindakan yang dilakukan oleh pengguna.
      15. Pengujian beta (*beta testing*). Pada pengujian ini, perangkat lunak didistribusikan sebagi sebuah versi beta dengan pengguna yang menguji aplikasi di situs mereka.
    1. **Whitebox Testing**

“Pengujian kotak putih atau disebut juga pengujian kotak kaca (*glass box testing*) merupakan sebuah filosofi perancangan *test case* yang menggunakan strukutur kontrol yang dijelaskan sebagai bagian dari perancangan peringkat komponen untuk menghasilkan *test case*”. (S. Pressman, 2012)**.** Prinsip dari keluaran tipe testing ini adalah: Klasifikasi white *box* testing mencakup beberapa pengujian, yaitu :

1. Pengujian unit. Pengujian ini berada pada tingkat yang sangat dasar seperti ketika unit kode dikembangkan atau fungsi tertentu dibangun. Pengujian unit berkaitan dengan unit secara keseluruhan.
2. Analisis statis dan dinamis. Analisis statis dilibatkan melalui kode untuk mengetahui segala kemungkinan cacat dalam kode, sedangkan analisis dinamis akan melibatkan pelaksanaan kode dan penganalisisan hasilnya.
3. Cakupan pernyataan. Dalam hal ini jenis pengujian kode dijalankan dengan setiap pernyataan dari aplikasi yang dijalankan minimal sekali. Hal tersebut membantu dalam memastikan semua pernyataan untuk dijalankan tanpa efek samping.
4. Cakupan cabang. Pengujian ini membantu pemvalidasian semua cabang di dalam kode dan memastikan bahwa tidak ada yang mengarah ke percabangan perilaku abnormal dari aplikasi.
5. Pengujian mutasi. Pada pengujian ini aplikasi diuji untuk kode yang telah dimodifikasi setelah pemasangan bug/cacat tertentu. Hal ini juga membantu dalam menemukan kode dan strategi pengodean yang dapat membantu dalam mengembangkan fungsi secara efektif.

Teknik *white box* sebagai berikut :

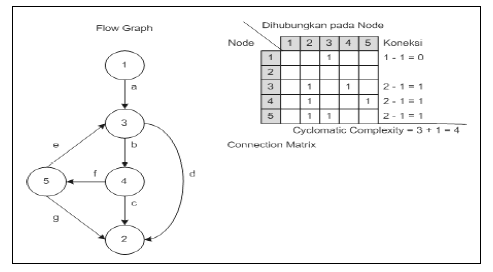
1. Pengujian basis *path*. Metode ini memungkinkan penguji dapat mengukur kompleksitas logis dari desain procedural dan menggunakannya sebagai pedoman untuk menetapkan himpunan basis dari semua jalur eksekusi.
2. Notasi Diagram Alir. Notasi yang digunakan untuk menggambarkan jalur eksekusi adalah notasi diagram alir (atau grafik program), yang menggunakan notasi lingkaran (simpul atau node) dan anak panah (link atau edge).



**Gambar 2.22** Notasi Diagram Alir

(Sumber : Janner Simarmata, 2010)

1. Kompleksitas Siklomatis adalah metrics perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program, nilai yang didapat akan menentukan jumlah jalur independen dalam himpunan path, serta akan memberi nilai batas atas bagi jumlah pengujian yang harus dilakukan, untuk memastikan bahwa semua pernyataan telah dieksekusi sedikitnya satu kali.
2. Matriks Grafis (Graph Matrik). Bentuk struktur data yang sering digunakan untuk mengambarkan pengujian adalah dengan matriks grafis. Matriks grafis adalah matriks bujursangkar yang berukuran sama dengan jumlah simpul pada grafik alir. Inputan dalam matriks harus bersesuaian dengan arah sisi dengan simpul.



**Gambar 2.23** Matriks Grafis

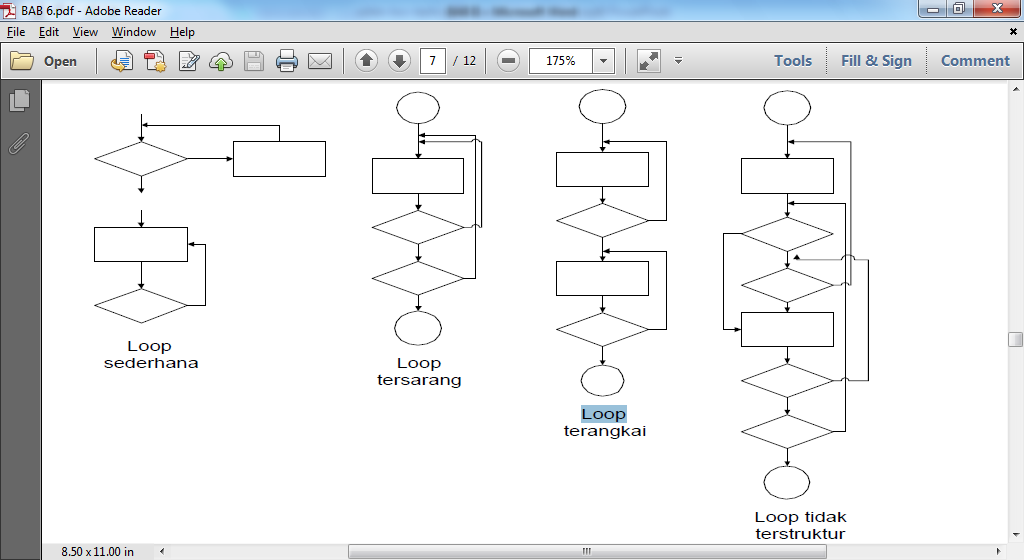
(Sumber : Janner Simarmata, 2010)

1. Pengujian Struktur Kontrol. Teknik pengujian basis path merupakan salah satu dari sejumlah teknik untuk pengujian struktur control, namun pengujian basis path tidak memadai untuk beberapa kasus uji.
2. Pengujian Kondisi. Pengujian kondisi menggunakan kondisi logis sederhana yang terdapat dalam program. Bila suatu kondisi tidak benar, maka akan terdapat paling tidak satu komponen dari kondisi yang salah, sehingga tipe kesalahan pada suatu kondisi meliputi :

* Kesalahan operator Boolean
* Kesalahan variable Boolean
* Kesalahan tanda kurung Boolean
* Kesalahan operator relasional
* Kesalahan persamaan aritmatika

1. Pengujian Aliran Data. Metode pengujian aliran data melakukan pengujian dengan mengggunakan definisi variable dalam program, efektif digunakan untuk melindungi kesalahan, tetapi akan memiliki cakupan pengukuran dan pemilihan jalur uji yang kompleks.
2. Pengujian *Loop*. *Loop* atau skema pengulangan sering digunakan dalam pembuatan program. Terdapat beberapa macam *loop*, yaitu:

* *Loop* Sederhana. Pengujian yang dilakukan harus memperhatikan hal-hal yaitu mengabaikan keseluruhan *loop*, hanya terdapat satu jalur yang melewati *loop*, suatu variable akan melewati loop jika bernilai lebih besar dari nilai yang ditentukan, harus memperhatikan batasan variable pada loop. (kondisi : n-1, n, n+1)
* *Loop* Tersarang. Jumlah pengujian akan bertambah secara geometris sesuai jumlah persarangan *loop* yang ada. Untuk menyederhanakan pengujian, maka harus diperhatikan hal-hal berikut yaitu memulai pengujian dari loop terdalam, dengan memulai pengujian dari nilai minimum, melakukan pengujian loop sederhana dari loop terdalam hingga loop terluar dengan memperhatikan parameter yang digunakan.
* Loop Terangkai. Pengujian untuk loop terangkai harus disesuaikan dengan independensi variable antara loop tersebut. Jika variable yang digunakan dalam loop kedua tidak bergantung dengan loop pertama, maka digunakan pengujian loop sederhana, sedangkan bila loop kedua bergantung secara nilai dengan loop kedua, maka lakukan pengujian tersarang.
* Loop Tidak Terstruktur. Sama sekali tidak dianjurkan untuk digunakan dalam membuat program.



**Gambar 2.24** Pengujian Loop

(Sumber : Janner Simarmata, 2010)