BAB 2
LANDASAN TEORI

2.1 Sistem Monitoring

Sistem monitoring atau sistem pengawasan adalah suatu upaya yang sistematik untuk menetapkan kinerja berdasarkan standar pada perencanaan untuk merancang sistem umpan balik informasi, untuk membandingkan kinerja actual dengan standar yang telah ditentukan, untuk menetapkan apakah telah terjadi suatu penyimpangan tersebut, serta untuk mengambil tindakan perbaikan yang diperlukan untuk menjamin bahwa semua sumber daya perusahaan atau organisasi telah digunakan seefektif dan seefisien mungkin guna mencapai tujuan perusahaan atau organisasi.[5]

2.2 Nata De Coco

Nata de coco merupakan salah satu produk olahan air kelapa yang sudah cukup populer di Indonesia. Di Indonesia sendiri nata de coco mulai dikenal sekitar tahun 1980-an. Hingga saat ini produk nata de coco sangat digemari oleh masyarakat Indonesia, sehingga tidak heran bila nata de coco bisa ditemukan dimana saja karena produk ini bisa digunakan sebagai bahan campuran es krim, cocktail buah, sirup dan makanan ringan lainnya. [6]

Air kelapa adalah bahan baku utama untuk membuat nata de coco. Air kelapa yang baik untuk digunakan dalam pembuatan nata de coco adalah air kelapa yang berasal dari kelapa tua. Air kelapa yang berasal dari buah kelapa yang masih muda atau kelapa tua yang sudah bertunas tidak bisa digunakan untuk membuat nata de coco. Air kelapa yang berasal dari buah kelapa yang masih muda belum cukup mengandung mineral yang diperlukan untuk pertumbuhan dan aktifitas bakteri *Acetobacter xylinum.* Sebaliknya, air kelapa yang telah bertunas mengandung minyak berlebihan yang dapat menghambat pertumbuhan bakteri *Acetobacter xylinum.*Kualitas nata de coco yang baik ditentukan oleh beberapa elemen seperti bahan baku, penambahan sumber nitrogen, penambahan sumber karbon, starter nata, wadah fermentasi dan sanitasi.[6]

Bahan baku

Bahan baku pembuatan nata de coco harus air kelapa murni, tidak tercampur air ataupun kotoran lainnya. Air kelapa tidak harus segar asalkan jangan lebih dari 8 hari penyimpanan karena telah berubah sifatnya akibat adanya fermentasi dan kontaminasi bakteri.

Sumber karbon

Gula dalam pembuatan nata de coco berfungsi sebagai sumber karbon atau energi. Semua jenis gula bisa digunakan sebagai sumber karbon baik itu glukosa, sukrosa, maupun maltosa. Adapun jumlah gula yang dianjurkan sebanyak 2%, karena penggunaan gula 2% akan menghasilkan rendemen *nata de coco* tidak jauh berbeda dengan penambahan 5%.

Wadah Fermentasi

Untuk mendapatkan rendemen yang tinggi sebaiknya digunakan wadah berbentuk segi empat (nampan) dengan tinggi 5 cm – 10 cm sehingga permukaan cukup luas.

Sanitasi

Kebersihan semua peralatan, bahan dan tempat produksi merupakan syarat mutlak untuk mencegah terjadinya kontaminasi karena bakteri *Acetobacter xylinum* sangat sensitif terhadap perubahan sifat-sifat fisik dan kimia lingkungan

Sumber nitrogen

Nitogen merupakan salah satu bahan yang dapat merangsang pertumbuhan dan aktifitas bakteri *Acetobacter xylinum.* Sumber nitrogen yang biasa digunakan adalah amonium sulfat (ZA) karena mudah didapat dan relatif murah.

Tetapi sesuai PK BPOM RI NO 7 tahun 2015 yang menerangkan pelarangan penggunaan pupuk ZA dalam proses pembuatan nata de coco. Oleh karena itu diperlukan sumber nitrogen alternatif yang bersumber dari tumbuhan yaitu kacang hijau, tauge, atau tahu. Teknologi pengolahan nata de coco merupakan teknologi sederhana, namun dalam proses pembuatannya terdapat beberapa titik kritis yang harus diperhatikan.[6]

Gambar 2.1 Nata De Coco

Macam-macam nata selain nata de coco sebagai berikut :

1. Nata de pina adalah nata yang dibuat dari buah atau limbah nanas yang berupa kulit, empulur dan mata nanas serta buah nanas masak optimum. Bahan diblender dengan tambahan air. Air digunakan sebagi media untuk nata dengan penambahan sumber nitrogen dan karbon (Iskandar dkk, 2010).

Gambar 2.2 Nata De Pina

2. Nata De Soya adalah nata yang dibuat dengan bibit Whey tahu (limbah cair tahu) dapat digunakan sebagai media pada pembuatan nata, karena masih mengandung sumber nitrogen yang dapat digunakan untuk pertumbuhan bakteri A. xylinum (Nugraheni, 2008)

Gambar 2.3 Nata De Soya

2.2.1 Cara Pembuatan

1. Air kelapa disaring supaya bersih dari kotoran, kemudian masak air kelapa yang sudah disaring tersebut.
2. Panaskan air kelapa sesuai perlakuan sampai mendidih. lalu ditambahkan gula pasir sebanyak 5%, asam cuka 0.01% mililiter. lalu aduk sampai rata.[7]
3. Jika sudah mendidih sampai 100 dejarat, dinginkan air kelapa.
4. Setelah didinginkan larutan siap dituangi starter (bibit bakteri nata) sebanyak 10% dari volume media bakteri. Dan tambahkan asam sitrat sesuai dengan yang ditentukan.
5. Masukkan air kelapa kedalam wadah yang sudah di sterilkan oleh alkohol, lalu tutup wadah dengan koran, dan tunggu proses fermentasinya.
6. Fermentasi larutan sampai tumbuh lembaran natanya kurang lebih 7 hari. 8. Cuci bersih dan potong-potong. 9. Analisis untuk memperoleh data yang diperlukan. [8]

2.2.2 Fermentasi

Fermentasi memiliki arti yang berbeda bagi ahli biokimia dan mikrobiologi industry. Arti fermentasi pada bidang biokimia dihubungkan dengan pembangkitan energi oleh katabolisme senyawa organic. Pada bidang mikrobiologi industri, fermentasi mempunyai arti yang menggambarkan setiap proses untuk menghasilkan produk dari pembiakan mikrooganisme. Dari uraian tersebut dapat diartikan bahwa fermentasi mempunyai pengertian suatu proses terjadinya perubahan kimia pada suatu substrat organic melalui aktivitas enzim yang dihasilkan oleh mikroorganisme. [9] Berdasarkan sumber mikrooganisme, proses fermentasi dibagi 2 (dua) yaitu :

1. Fermentasi spontan, adalah fermentasi bahan pangan dimana dalam pembuatannya tidak ditambahkan mikroorganisme dalam bentuk starter atau ragi, tetapi mikroorganisme berkembang baik secara spontan karena lingkungan hidupnya dibuat sesuai untuk pertumbuhannya, dimana aktivitas dan pertumbuhan bakteri asam laktat dirangsang karena adanya garam, contohnya pada pembuatan sayur asin.[9]

Gambar 2.4 Sayur Asin

2. Fermentasi tidak spontan adalah fermentasi yang terjadi dalam bahan pangan yang dalam pembuatannya ditambahakan mikroorganisme dalam bentuk starter atau ragi, dimana mikroorganisme tersebut akan tumbuh dan berkembangbiak secara aktif merubah bahan yang difermentasi menjadi produk yang diinnginkan contohnya pembuatan nata de coco dan oncom.[9]

Gambar 2.5 Oncom

Faktor-faktor yang mempengaruhi fermentasi adalah sebagai berikut

1. Nutrisi.

   Aktivitas A. xylinum dalam menghasilkan nata dipengaruhi oleh kandungan glukosa dalam substrat atau media yang digunakan. Penambahan gula pasir ke dalam air kelapa dimaksudkan untuk memenuhi kebutuhan nutrisi dan karbon bagi A. xylinum. Ekstrak tauge ditambahkan sebagi sumber nitrogen, namun sumber ini tidak mutlak diperlukan karena nitrogen dapat diasimilasi dari protein yang terkandung dalam air kelapa.

2. Tingkat Keasaman (pH). Lapisan nata dapat terbentuk lebih tebal pada pH optimal 3,5-4. Pada pH netral, nata yang terbentuk cenderung tipis dan terbentuk setelah minimal 10 hari waktu inkubasi. Pengaturan pH pada pembuatan nata dapat dilakukan dengan penambahan cuka makan

sehingga pH media lebih asam. Keasaman yang rendah meningkatkan pertumbuhan A. xylinum dan mencegah kontaminasi jenis bakteri lain.[10]

Berikut tabel 2.1 hasil uji beda faktor pH pada faktor lama fermentasi sebagai berikut.

Tabel 2.1 Hasil Uji Beda Faktor pH Media,
Faktor Lama Fermentasi

pH Media	Perlakuan Lama Fermentasi 8 hari	Perlakuan Lama Fermentasi 11 hari	Perlakuan Lama Fermentasi 14 hari
pH 3,5	0,792	1,252	1,438
pH 4,0	1,113	1,369	1,704
pH 4,5	1,181	1,447	1,527
Rata-rata	1,029	1,356	1,556

Perlakuan pH media menunjukkan pengaruh sangat nyata terhadap tebal, berat, warna, volume media sisa dan pH media sisa nata de coco. Perlakuan lama fermentasi juga menunjukkan pengaruh pada tebal, berat, tekstur, warna, volume sisa. Semakin besar pH media dan semakin lama fermentasi maka akan menghasilkan nata de coco yang semakin tebal dan berat, tekstur yang semakin kenyal, warna semakin gelap dan media sisa semakin rendah, meski pada pH 4,5 hasilnya tidak berbeda nyata dengan pH 4. [11]

3. Temperatur optimum fermentasi adalah 28^{o}-30^{o}C atau pada suhu kamar. Pada temperatur ini dihasilkan nata yang paling tebal dibandingkan.[12]
4. Volume starter jumlah larutan induk (percent inokulum) besar sekali pengaruhnya terhadap ketebalan nata yang dihasilkan. Dengan semakin besar larutan induk, semakin banyak jumlah bakteri Acetobacter Xylinum yang ada. Dari hasil penelitian diketahui bahwa untuk mendapatkan ketebalan yang maksimum, diperlukan 20 % inokulum dalam media fermentasi.[12]

2.3 Internet

Internet adalah singkatan dari *Interconnected Networking* yang apabila diartikan dalam Bahasa Indonesia berarti kumpulan komputer yang terhubung di dalam beberapa rangkaian jaringan. Internet merupakan salah satu hasil dari kecanggihan dan kemajuan ilmu pengetahuan dan teknologi buatan manusia. Internet merupakan sebutan untuk sekumpulan jaringan komputer yang dapat menghubungkan berbagai situs akademik, pemerintahan, komersial, organisasi, hingga perorangan. Lebih lanjut dijelaskan bahwa internet mampu untuk menyediakan akses untuk layanan telekomunikasi dan berbagai sumber daya informasi untuk jutaan pemakaiannya yang tersebar di seluruh dunia. Internet memliki berbagai macam layanan-layanan internet meliputi komunikasi secara langsung seperti email dan juga chatting, diskusi seperti *Usenet News*, email dan juga milis serta sumber daya informasi yang terdistrubusi (*World Wide Web*, *Gopher*), *remote login*, dan lalu lintas file (Telnet, FTP), dan lain-lainnya [13]

2.3.1 Konsep Internet of Things (IoT)

Internet of Things atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data, remote kontrol, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang semuanya tersambung ke jaringan lokal dan global melalui sensor yang tertanam dan selalu aktif. Pada dasarnya, *Internet of Things* mengacu pada benda yang dapat diidentifikasikan secara unik sebagai representasi virtual dalam struktur berbasis Internet. Istilah Internet of Things awalnya disarankan oleh Kevin Ashton pada tahun 1999 dan mulai terkenal melalui Auto-ID Center di MIT. Dengan semakin berkembangnya infrastruktur internet, maka kebutuhan hidup menuju babak berikutnya, dimana bukan hanya *smartphone* atau komputer saja yang dapat terkoneksi dengan internet. Namun berbagai macam benda nyata akan terkoneksi dengan internet. Sebagai contohnya dapat berupa mesin produksi, mobil, peralatan elektronik, peralatan yang dapat dikenakan manusia (wearables),

dan termasuk benda nyata apa saja yang semuanya tersambung ke jaringan lokal dan global menggunakan sensor dan atau aktuator yang tertanam [14].

Gambar 2.6 Ilustrasi Internet Of Things

Dapat kita simpulkan bahwa *internet of things* membuat suatu koneksi antara mesin dengan mesin, sehingga mesin-mesin tersebut dapat berinteraksi dan bekerja secara independen sesuai dengan data yang diperoleh dan diolahnya secara mandiri. Tujuannya adalah untuk membuat manusia berinteraksi dengan benda dengan lebih mudah, bahkan supaya benda juga bisa berkomunikasi dengan benda lainnya.

Teknologi *internet of things* sangat luar biasa. Jika sudah direalisasikan, teknologi ini tentu akan sangat memudahkan pekerjaan manusia.
Manusia tidak akan perlu lagi mengatur mesin saat menggunakannya, tetapi mesin tersebut akan dapat mengatur dirinya sendiri dan berinteraksi dengan mesin lain yang dapat berkolaborasi dengannya.

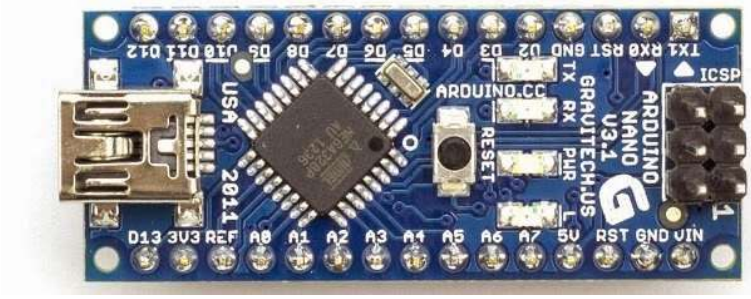
Cara kerja dari *internet of things* cukup mudah. Setiap benda harus memiliki sebuah IP Address. IP Address adalah sebuah identitas dalam jaringan yang membuat benda tersebut bisa diperintahkan dari benda lain dalam jaringan yang sama. Selanjutnya, IP address dalam benda-benda tersebut akan dikoneksikan ke jaringan internet. Saat ini, koneksi internet sudah sangat mudah kita dapatkan. Dengan demikian, kita dapat memantau benda tersebut bahkan memberi perintah kepada benda tersebut. Sebagai contoh, jika ada speaker yang memiliki IP address dan terkoneksi internet di Amerika Serikat, kita dapat memerintahkan speaker tersebut untuk menyalakan musik walaupun kita berada di Indonesia. Yang kita

perlukan hanyalah koneksi internet. Lalu apa hubungannya dengan *internet of things*, Setelah sebuah benda memiliki IP address dan terkoneksi dengan internet, pada benda tersebut juga dipasang sebuah sensor. Sensor pada benda memungkinkan benda tersebut memperoleh informasi yang dibutuhkan,benda tersebut dapat mengolah informasi itu sendiri, bahkan berkomunikasi dengan benda-benda lain yang memiliki IP address dan terkoneksi dengan internet juga. Akan terjadi pertukaran informasi dalam komunikasi antara benda-benda tersebut.[15]

2.4 Arduino

Arduino adalah salah satu jenis Microcontroller yang paling populer di dunia. Paling banyak digunakan di dunia. Arduino ini menggunakan chip AVR sebagai microcontrollernya. Karena ukurannya yang kecil, microcontroller sering digunakan untuk mengendalikan rangkaian lampu LED, membuat MP3 Player, DVD, Televisi, AC, dan untuk membuat sebuah projek yang kita butuhkan seperti alarm motor misalkan . Dan tentu saja bisa juga untuk membuat projek robot. Dan dalam robot sendiri kita sudah ketahui bahwa microcontroller, berfungsi sebagai otaknya Ada beberapa tipe arduino sebagai berikut :

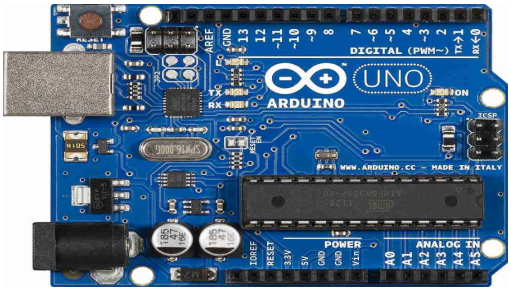
Arduino Nano. Sepertinya namanya, Nano yang berukulan kecil dan sangat sederhana ini, menyimpan banyak fasilitas. Sudah dilengkapi dengan FTDI untuk pemograman lewat Micro USB. 14 Pin I/O Digital, dan 8 Pin input Analog (lebih banyak dari Uno). Dan ada yang menggunakan ATMEGA168, atau ATMEGA328.[16]



Gambar 2.7 Arduino Nano

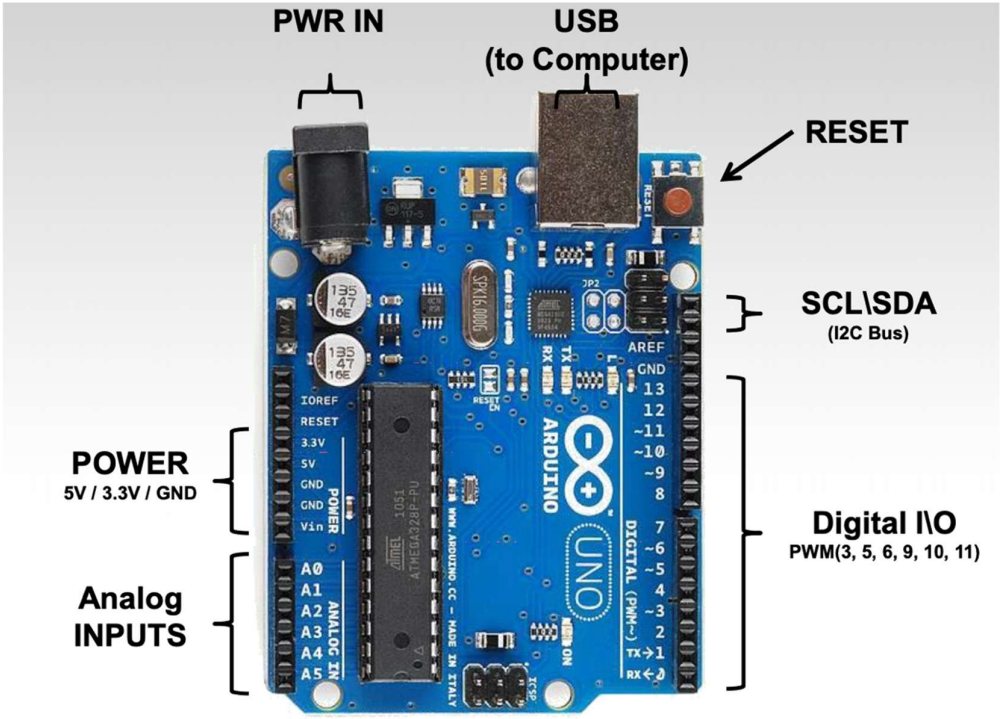
Arduino Uno adalah sebuah board mikrokontroler yang didasarkan pada Atmega328. Arduino Uno mempunyai 14 Input/output dengan 6 input PWM, 6

input analog, sebuah osilator kristal 16 Mhz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino Uno memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkan ke sebuah komputer dengan sebuah kabel USB dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.[16]



Gambar 2.8 Arduino Uno

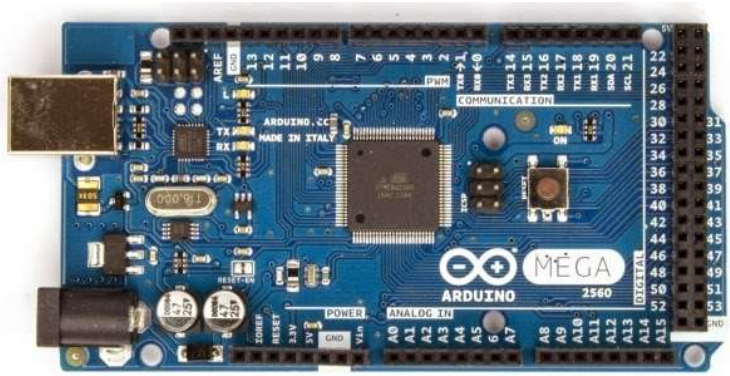
Penjelasan port yang tersedia pada Arduino Uno sebagai berikut :



Gambar 2.9 Port pada Arduino Uno

Arduino Mega. Mirip dengan Arduino Uno, sama-sama menggunakan USB type A to B untuk pemogramannya. Tetapi Arduino Mega, menggunakan Chip

yang lebih tinggi ATMEGA2560. Dan tentu saja untuk Pin I/O Digital dan pin input Analognya lebih banyak dari Uno.[16]

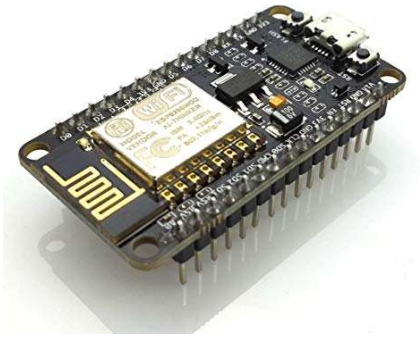


Gambar 2.10 Arduino Mega

Dengan penggunaan yang cukup sederhana, anda tinggal menghubungkan power dari USB ke PC anda atau melalui adaptor AC/DC ke jack DC.

2.5 NodeMCU

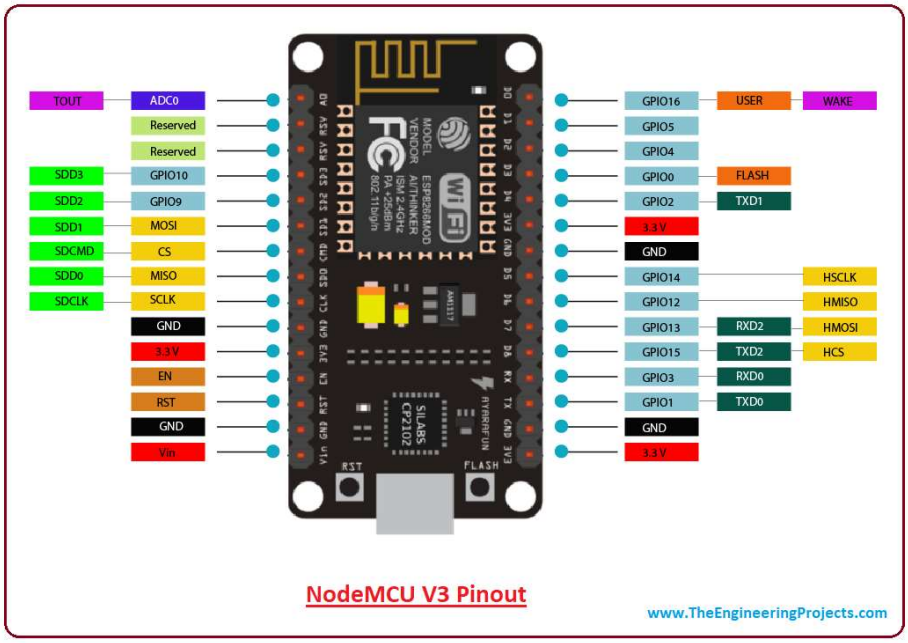
NodeMCU pada dasarnya adalah pengembangan dari ESP 8266 dengan firmware berbasis e-Lua. Pada NodeMcu dilengkapi dengan micro usb port yang berfungsiuntuk pemorgaman maupun power supply. Selain itu juga pada NodeMCU di lengkapi dengan tombol push button yaitu tombol reset dan flash. NodeMCU menggunakan bahasa pemorgamanan Lua yang merupakan package dari esp8266. Bahasa Lua memiliki logika dan susunan pemorgaman yang sama dengan c hanya berbeda syntax. Jika menggunakan bahasa Lua maka dapat menggunakan tool Lua loader maupun Lua uploder.Selain dengan bahasa Lua NodeMCU juga support dengan sofware Arduino IDE dengan melakukan sedikit perubahan board manager pada Arduino IDE.



Gambar 2.11 Modul NodeMCU

(Sumber : Dian Mustika P.2017)

Port-port yang tersedia pada nodeMCU seperti gambar dibawah.



Gambar 2.12 Port port nodeMCU

2.6 Sensor

Sensor adalah suatu perangkat yang mendeteksi perubahan energi yang berada di alam seperti energi listrik, energi fisika, energi kimia, energi biologi, energi mekanik dan sebagainya. Di dalam sebuah sensor terdapat transduser yang berfungsi untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi besaran listrik berupa tegangan, resistansi dan arus listrik.

2.6.1 Sensor pH

Sensor pH adalah alat elektronik yang digunakan untuk mengukur pH kadar keasaman atau alkalinitas ataupun basa dari suatu larutan meskipun *probe* khusus terkadang digunakan untuk mengukur pH zat semi padat. Sensor pH yang biasa terdiri dari pengukuran probe pH elektroda gelas yang terhubung ke pengukuran pembacaan yang mengukur dan menampilkan pH yang terukur. Prinsip kerja dari alat ini yaitu semakin banyak electron pada sampel maka akan semakin bernilai asam begipun sebaliknya.

Probe pH mengukur pH seperti aktifitas yang mengelilingi bohlam kaca berdinding tipis pada ujungnya. Probe ini menghasilkan tegangan rendah sekitar 60mV per unit pH yang diukur dan ditampilkan sebagai pembacaan nilai pH.

Sensor pH PH-4502C Modul sensor ini merupakan module yang berfungsi untuk mendeteksi tingkat ph air yang dimana outputnya berupa tegangan analog.

Gambar 2.13 Sensor pH PH-4052C

Sensor pH PH-4502C Modul sensor ini merupakan module yang berfungsi untuk mendeteksi tingkat ph air yang dimana outputnya berupa tegangan analog sama seperti sensor pH PH-4502C, hanya biasanya sensor PH-4502C digunakan untuk skala besar dan harga lebih mahal.

Gambar 2.14 Sensor pH SEN0169

2.6.2 Sensor Ultrasonik

Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek tertentu di depannya, frekuensi kerjanya pada daerah diatas gelombang suara dari 40 KHz hingga 400 KHz. Sensor ultrasonik terdiri dari dari dua unit, yaitu unit pemancar dan unit penerima. Sensor jarak ultrasonik HC-SR04 adalah sensor 40 KHz. HC-SR04 merupakan sensor ultrasonik yang dapat digunakan untuk mengukur jarak antara penghalang dan sensor [25]. 2 Komponen utama sebagai penyusunnya yaitu *ultrasonic transmitter* dan *ultrasonic receiver*. Fungsi dari *ultrasonic transmitter* adalah memancarkan gelombang ultrasonik dengan frekuensi 40 KHz kemudian *ultrasonic receiver* menangkap hasil pantulan gelombang ultrasonik yang mengenai suatu objek. Waktu tempuh gelombang ultrasonik dari pemancar hingga sampai ke penerima sebanding dengan 2 kali jarak antara sensor dan bidang pantul. Gambar merupakan betuk sensor ultrasonic HC-SR04.

Gambar 2.15 Sensor Ultrasonik HC-SR04

2.6.3 Sensor Suhu

Sensor suhu LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Sensor Suhu LM35 yang dipakai dalam penelitian ini berupa komponen elektronika elektronika yang diproduksi oleh National Semiconductor. LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, LM35 juga mempunyai keluaran impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus serta tidak memerlukan penyetelan lanjutan.

Meskipun tegangan sensor ini dapat mencapai 30 volt akan tetapi yang diberikan kesensor adalah sebesar 5 volt, sehingga dapat digunakan dengan catu daya tunggal dengan ketentuan bahwa LM35 hanya membutuhkan arus sebesar 60 µA hal ini berarti LM35 mempunyai kemampuan menghasilkan panas (self-heating) dari sensor yang dapat menyebabkan kesalahan pembacaan yang rendah yaitu kurang dari 0,5 ºC pada suhu 25 ºC .[17]

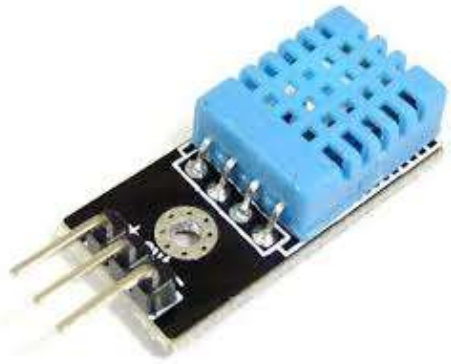
Gambar 2.16 Sensor suhu LM35

Sensor DS18B20 menggunakan sebuah kabel protokol bus eksklusif maxim yang menerapkan komunikasi bus menggunakan satu sinyal kontrol. Garis kontrol memerlukan resistor pull up lemah karena semua perangkat yang terhubung ke bus melalui 3 bagian atau open-drain port (pin DQ pada sensor DS18B20). Dalam sistem bus ini, mikroprosesor (perangkat master) mengidentifikasi dan mengalamatkan perangkat pada bus menggunakan kode unik 64-bit dari masing-masing perangkat. Karena masing-masing perangkat memiliki kode unik, jumlah perangkat yang dapat diatasi pada satu DS18B20 bus hampir tak terbatas. [18]

Gambar 2.17 Sensor Suhu DS18B20

Sensor DHT11 adalah module sensor yang berfungsi untuk mensensing objek suhu dan kelembaban yang memiliki output tegangan analog yang dapat diolah lebih lanjut menggunakan mikrokontroler.

Module sensor ini tergolong kedalam elemen resistif seperti perangkat pengukur suhu seperti contohnya yaitu NTC. Kelebihan dari module sensor ini dibanding module sensor lainnya yaitu dari segi kualitas pembacaan data sensing yang lebih responsif yang memliki kecepatan dalam hal sensing objek suhu dan kelembaban, dan data yang terbaca tidak mudah terinterverensi.[17]



Gambar 2.18 Sensor Suhu DHT11

2.7 Solenoid Valve

Solenoid valve berfungsi menghentikan atau meneruskan aliran refrigeran dalam suatu sistem refrigerasi, dimana pengaturannya dilakukan oleh arus listrik. Solenoid valve terdiri dari sebuah kumparan yang berbentuk silinder dimana pada bagian tengahnya terdapat sebuah inti besi yang mudah dibuat magnet yang disebut dengan plunger. Apabila kumparan dialiri arus listrik maka kumparan menjadi elektromagnet sehinggga akan mengangkat/menarik plunger ke tengah kumparan dan akibatnya akan membuka katup. Apabila aliran listrik dimatikan maka medan magnet kumparan akan hilang dan plunger karena beratnya sendiri akan turun sehingga menutup katup.

Gambar 2.19 Solenoid Valve SV44-24V

Solenoid Valve SNS 2L terbuat dari metal dan untuk harga lebih mahal dan daya tahan lebih lama.

Gambar 2.20 Solenoid Valve SNS 2L

2.8 Konsep Perancangan Berorientasi Objek

Pendekatan perancangan berorientasi objek menganalogikan sistem aplikasi seperti kehidupan yang memiliki elemen-elemen objek didalamnya. Dalam membangun sistem yang berorientasi objek, langkah awal yang dilakukan adalah melakukan proses analisis dan perancangan yang berorientasi pada objek-objek yang terlibat didalam sistem. Tujuan dilakukannya proses tersebut adalah untuk mempermudah pengembang dalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut yang kemudian dimodelkan dalam sistem nyata. Perusahaan software Rational Software, telah membentuk konsorsium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language* (UML) sebagai Bahasa standar dalam *Object Oriented Analysist Design* (OOAD).

2.8.1 Unified Modelling Language

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. *UML* merupakan himpunan struktur dan teknik untuk pemodelan desain program

berorientasi objek (*OOP*) serta aplikasinya. *UML* adalah metodologi untuk mengembangkan sistem *OOP* dan sekelompok perangkat (*tool*) untuk mendukung pengembangan sistem tersebut. *UML* merupakan sistem arsitektur yang bekerja dalam *OOAD* dengan satu bahasa yang konsisten untuk menentukan, visualisasi, konstruksi dan mendokumentasikan artifact yang terdapat dalam sistem. Artifact adalah sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*. Artifact dapat berupa model, deskripsi atau software.[18]

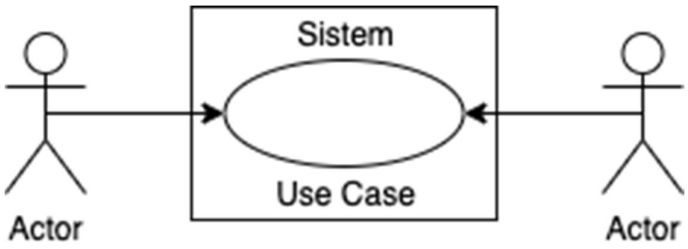
2.8.2 Diagram UML

UML dalam mendokumentasikan rancangan menyediakan berbagai macam diagram untuk memodelkan aplikasi perangkat lunak berorientasi objek . Namun, dalam penelitian ini hanya menggunakan empat macam diagram saja untuk memodelkannya. Empat macam diagram yang digunakan diantaranya yaitu:

1. Use Case Diagram

   Use case diagram adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatan *use case* adalah serangkaian *scenario* yang digabungkan bersama-sama oleh tujuan umum pengguna. *Use case* merupakan alat bantu guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Tidak selalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah sistem. Karena sistem pengembangan tradisional sering ceroboh dalam melakukan analisis, akibatnya pengguna seringkali sulit menemukan jawaban tatkala dimintai masukan tentang sesuatu. Diagram *use case* menunjukan 3 aspek dari sistem yaitu : *actor*, *use case* dan *sistem* atau

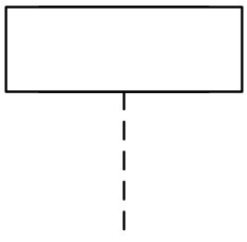
sub sistem boundary. *Actor* mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan *use case*.



Gambar 2.21 Use Case Diagram

2. Sequence Diagram

   Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukan sejumlah contoh objek dan *message* (pesan) yang diletakan diantara objek-objek ini di dalam *use case*. Komponen utama *sequence diagram* terdiri atas objek yang ditulisan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukan dengan *progress vertical*. Objek diletakan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Mereka diatur dalam urutan guna menyederhanakan diagram, istilah objek dikenal juga dengan *participant*, setiap *participant* terhubung dengan garis titik-titik yang disebut *lifeline*. Sepanjang *lifeline* ada kotak yang disebut *activation*, *activation* mewakili sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*.

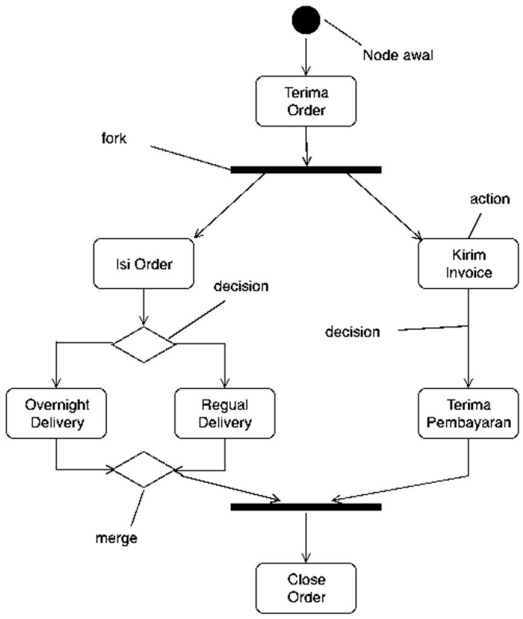


Gambar 2.22 Participant pada sebuah Sequence Diagram

3. Activity Diagram

   Activity diagram adalah bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika prosedural, proses bisnis dan aliran

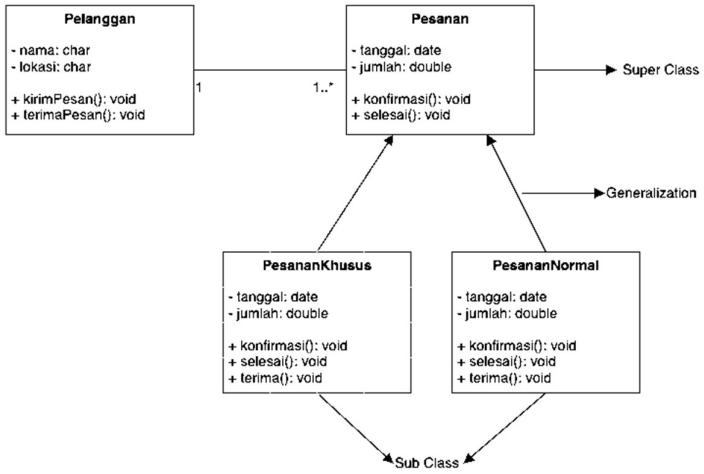
kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram*. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaanya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku pararel sedangkan *flowchart* tidak bisa. Tujuan dari *activity diagram* adalah untuk menangkap tingkah laku dinamis dari sistem dengan cara menunjukan aliran pesan dari satu aktifitas ke aktifitas lainya. Secara umum *activity diagram* digunakan untuk menggambarkan diagram alir yang terdiri dari banyak aktifitas dalam sistem dengan beberapa fungsi tambahan seperti percabangan, aliran pararel, *swim lane*, dan sebagainya. Sebelum menggambarkan sebuah *activity diagram*, perlu adanya pemahaman yang jelas tentang elemen yang akan digunakan dalam *activity diagram*. Elemen utama dalam *activity diagram* adalah aktifitas itu sendiri. Aktifitas adalah fungsi yang dilakukan oleh sistem setelah akitifitas terindentifikasi, selanjutnya yang perlu diketahui adalah bagaimana semua elemen tersebut berasosiasi dengan *constraint* dan kondisi lalu perlu penjabaran tata letak dari keseluruhan aliran agar bisa ditransofrmasikan ke *activity diagram*.



Gambar 2.23 Contoh Activity Diagram Sederhana

4. Class Diagram

   Class diagram adalah diagram statis. Diagram ini mewakili pandangan statis dari suatu aplikasi. *Class diagram* tidak hanya digunakan untuk memvisualisasikan, menggambarkan dan mendokumentasikan berbagai aspek sistem tetapi juga membangun kode eksekusi dari aplikasi perangkat lunak. *Class diagram* menggambarkan *atribut*, *operation* dan juga *constraint* yang terjadi pada sistem. *Class diagram* banyak digunakan dalam pemodelan sistem OO karena mereka adalah satu-satunya diagram UML yang dapat dipetakan langsung dengan bahasa berorientasi objek. *Class diagram* menunjukan koleksi *class*, antarmuka, asosiasi, kolaborasi, dan *constraint*, dikenal juga sebagai diagram structural.



Gambar 2.24 Contoh Class Diagram sederhana

2.8.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *System Analyst* dalam tahap analisis persyaratan proyek pengembangan sistem. ERD merupakan alat peraga yang memberikan dasar untuk desain *database* relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk *database*.[19]

Dalam *Entity Relationship Diagram* (ERD) terdapat beberapa komponen, yaitu sebagai berikut :

1. Entitas (*Entity*)

   Entitas adalah suatu objek yang memiliki hubungan dengan objek lain. Dalam ERD digambarkan dengan bentuk persegi panjang.

2. Hubungan (Relationship)

   Merupakan keterangan entitas dapat berhubungan dengan entitas lain, hubungan ini disebut dengan entity relationship yang digambarkan dengan garis. Terdapat 4 tipe relasi dalam *database* yaitu :

   a. One-to-One

      Artinya satu data memiliki satu data pasangan di tabel lain. Jadi, jika dua tabel berelasi one-to-one artinya setiap *record* di entitas pertama hanya akan berhubungan dengan satu *record* di entitas kedua begitu pula sebaliknya. Conrohnya relasi antara tabel pegawai dan alamat pegawai. Satu dokumen pegawai hanya berhubungan dengan satu *record* alamat pegawai begitu pula sebaliknya.

   b. One-to-Many

      Artinya satu data memiliki beberapa data pasangan di tabel lain. Jadi, misalkan terdapat relasi antara tabel satu dan tabel dua yang berarti satu dokumen pada tabel satu boleh berelasi (mempunyai) dengan banyak dokumen pada tabel dua. Namun, satu dokumen pada tabel dua hanya boleh berelasi dengan satu dokumen saja pada tabel satu.

   c. Many-to-One

      Artinya beberapa data memiliki satu data pasangan di tabel lain. Jadi, misalkan terdapat relasi antara tabel satu dengan tabel dua, dapat diartikan bahwa satu dokumen pada tabel satu hanya boleh berelasi (mempunyai) dengan satu dokumen pada tabel dua. Tetapi, satu dokumen pada tabel dua boleh berelasi dengan banyak dokumen pada tabel satu.

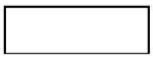
d. Many-to-Many

Artinya beberapa data memiliki beberapa data pasangan di tabel lain. Jadi, jika tabel satu berelasi dengan tabel dua dengan relasi many-to- many berarti bahwa ada banyak dokumen di entitas satu dan entitas dua yang saling berhubungan satu sama lain.

3. Atribut

Atribut adalah elemen dari entitas yang berfungsi sebagai deskripsi karakter entitas dan digambarkan dengan bentuk elips [19].

Tabel 2.2 Tabel Entity Relationship Diagram (ERD)

No	Simbol	Keterangan
1		Entitas
2		Atribut
3		Hubungan
4		Garis

2.9 Smartphone

Smartphone adalah telepon pintar yang memiliki kemampuan seperti komputer. Smartphone diklasifikasikan sebagai high end mobile phone yang dilengkapi dengan kemampuan mobile computing. Dengan kemampuan mobile computing tersebut, smartphone memiliki kemampuan yang tak bisa dibandingkan dengan ponsel biasa. Smartphone yang pertama kali muncul merupakan kombinasi dari fungsi suatu personal digital assistant (PDA) dengan telepon genggam ataupun telepon dengan kamera.Seiring dengan perkembangannya, kini smartphone juga mempunyai fungsi sebagai media player portable, low end digital compact camera, pocket video camera dan GPS. Smartphone modern juga dilengkapi dengan layar touch screen resolusi tinggi, browser yang mampu menampilkan full web seperti pada PC, serta akses data WiFi dan internet broadband.

2.9.1 Karakteristik Smartphone

Beberapa karakteristik yang umum ada pada smartphone yaitu:

1. Mobile OS

   Mobile OS yang sering digunakan pada smartphone adalah:

   Symbian OS, IPhone OS, Windows Mobile OS, RIM Blackberry, Linux, Palm OS, Android.
2. OpenSource
3. WebFeature
4. Enhanced Hardware

   Fitur hardware eksternal seperti layar sentuh lebar dan sensitif, built-in keyboard, resolusi kamera tinggi, sisi kamera depan untuk video conferences.

MobilePC Pada umumnya smartphonememiliki prosesor yang cukup tinggi., selain itu memiliki penyimpanan memori yang besar dan memiliki RAM tambahan yang cukup besar seperti sebuah PC desktop atau laptop.

2.9.2 Android

Menurut Nasruddin Safaat H (Pemrograman aplikasi mobile smartphone dan tablet PC berbasis android 2012:1) android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak. Awalnya, GoogleInc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [13].

2.9.2.1 Versi Android

Android memiliki sejumlah pembaharuan semenjak rilis aslinya. Pembaharuan ini dilakukan untuk memperbaiki bug dan menambah fitur-fitur yang baru. Berikut merupakan versi-versi yang dimiliki Android sampai saat ini:

Versi Android	API Level	Nickname
Android 1.0	1	
Android 1.1	2	
Android 1.5	3	Cupcake
Android 1.6	4	Donut
Android 2.0	5	Eclair
Android 2.01	6	Eclair
Android 2.1	7	Eclair
Android 2.2	8	Froyo (Frozen Yogurt)
Android 2.3	9	Gingerbread
Android 2.3.3	10	Gingerbread
Android 3.0	11	Honeycomb
Android 3.1	12	Honeycomb
Android 3.2	13	Honeycomb
Android 4.0	14	Ice Cream Sandwich
Android 4.0.3	15	Ice Cream Sandwich
Android 4.1	16	Jelly Bean

Gambar 2.25 Versi-versi Android

2.9.2.2 Arsitektur Android

Android merupakan kernel Linux yang menyediakan dan mengatur alur proses aplikasi. Gambar 2.1merupakan struktur dari sistem operasi Android.

Gambar 2.26 Arsitektur Android

(Sumber : http://developer.android.com/guide/basics/what-is-android.html)

Arsitektur Android terdiri atas:

1. Application

   Application merupakan bagian yang memuat aplikasi – aplikasi yang dapat digunakan oleh pengguna perangkat Android. Pada bagian Application ini Android memasukkan satu set aplikasi inti yang meliputi email client, program SMS, kalendar, peta, browser, dan kontak.Selain aplikasi inti seperti yang terdapat pada arsitektur Android, aplikasi - aplikasi tambahan yang diinstal sendiri oleh penguna juga akan menempati bagian Applicationdan memiliki hak akses yang sama terhadap ApplicationFramework.(Developer Android, 2012).

2. Application Framework

   ApplicationFramework merupakan bagian yang dapat digunakan oleh pengembang aplikasi dalam membuat aplikasi android. Dengan menyediakan pengembangan platform yang terbuka, Android menawarkan kemampuan pengembangan untuk membangun aplikasi yang sangat kaya dan inovatif. Pengembang diberikan kebebasan untuk mengambil keuntungan dari perangkat keras, mengakses informasi, run background services, set alarm, serta menambahkan pemberitahuan ke status bar. Dalam mengembangkan aplikasi, pengembang memiliki akses ke framework API (Application Programming Interface) yang sama dengan yang dapat diakses oleh aplikasi –aplikasi inti dari android. (Developer Android, 2012)

3. Libraries

   Libraries merupakan kumpulan kode yang dapat digunakan oleh komponen atau program lain. Penulisan kode pada Libraries ditulis menggunakan bahasa pemrograman C/C++.

Beberapa Libraries yang terdapat pada Android yaitu :

1. System C Library, implementasi BSD yang diturunkan dari sistem librarystandar C (libc).
2. Media Libraries, berdasarkan OpenCORE PacketVideo; library dapat mendukung pemutaran dan perekaman audio dan format video, serta

file gambar, termasuk MPEG4, H.264, MP3, AAC, AMR, JPG, dan PNG.

3. Surface Manager, library yang mengelola penggambaran dalam bentuk komposisi 2D, grafis 3D.
4. LibWebCore
5. SGL(Scalable Graphics Library), library yang menangani pengelolaan grafis 2D.
6. 3D libraries, libraryyang menangani pengelolaan grafis 3D.
7. FreeType, library yang menangani pengelolaan rendering font.
8. SQLite, digunakan untuk pengelolaan database.

(Developer Android, 2012).

4. Android Runtime

Android Runtime memiliki dua bagian utama, yaitu:

a. CoreLibraries
b. Dalvik Virtual Machine(DVM)

Aplikasi Android ditulis dengan menggunakan bahasa pemrograman Java. DalvikVirtual Machine (DVM) adalah sebuah virtual machine yang digunakan untuk menterjemahkan instruksi –instruksi program Java ke dalam instruksi yang dimengerti oleh sistem operasi. Namun dalam platform Android virtual machine yang digunakan bukan Java Virtual Machine (JVM) tetapi Dalvik Virtual Machine (DVM). Dalvik Virtual Machine adalah sebuah virtual machine yang dioptimasi untuk perangkat yang memiliki memori kecil, sumber daya terbatas, dan kemampuan proses yang kecil. Dalvik Virtual Machine mengeksekusi file dalam bentuk Dalvik executable (. Dex).(Andry, 2011).

5. Linux Kernel

Dalam bagian ini android menggunakan modifikasi dari Linux Kernel versi 2.6. Bagian ini bertanggung jawab untuk mengelola dan berkomunikasi dengan perangkat keras dimana android berjalan. Pemilihan Linux Kernel sebagai inti dari android adalah karena dukungan dan kestabilannya terhadap berbagai macam

komponen perangkat keras. Pada bagian ini disediakan driver (program pengendali) perangkat keras, pengelolaan memori, pengelolaan proses, pengelolaan jaringan, dan keamanan. (Developer Android, 2012).

2.10 Database

Database atau basis data merupakan sebuah koleksi atau kumpulan dari data yang bersifat mekanis, terbagi, terdefinisi secara formal serta terkontrol. Pengontrolan dari sistem database tersebut adalah terpusat, yang biasanya dimiliki dan juga dipegang oleh suatu organisasi. Pemanfaatan basis data untuk pengolahan data, juga memiliki tujuan – tujuan lain. Secara lengkap tujuan pemanfaatan basis data adalah sebagai berikut :

1. Kecepatan dan Kemudahan (*speed*).

   Dengan menggunkan basis data pegambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokan, mengurutkan bahkan perhitungan dengan matematika.

2. Kebebasan (data independence).

   Jika sebuah program telah selesai dibuat, dan ternyata ada perubahan isi atau struktur data. Maka dengan basis data, perubahan ini hanya perlu dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

3. Keakuratan (*accuracy*).

   Penerapan secara ketat aturan tipe data, domain data, keunikan data, hubungan antar data, dan lain-lain menekan ketidakakuratan dalam pemasukan atau penyimpanan data.

4. Ketersediaan (*availability*).

   Dengan basis data kita dapat mem-backup data, memilah-milah data mana yang masih diperlukan dan data mana yang perlu disimpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi suatu organisasi dari waktu ke waktu membutuhkan media penyimpanan yang semakin besar.

5. Keamanan (*security*).

   Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna. Pengguna diberi hak akses yang berbeda-beda sesuai dengan kepentingan dan posisinya. Basis data bisa diberikan password untuk membatasi orang yang mengaksesnya.

6. Kebersamaan Pemakaian (*sharability*).

   Karena cukup dengan satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja. Jika ada perubahan data alamat mahasiswa misalnya, maka tidak perlu kita memperbarui semua data di masing-masing bagian, tetapi cukup hanya disatu basis data.[20]

Database disimpan didalam tabel, dan tabel mengandung data yang berhubungan, atau *entity*, seperti misalnya orang, produk, pesanan, dan sebagainya. Tujuannya adalah menjaga table tetap kecil dan dapat dikelola, serta *entity-entity* yang terpisah disimpan dalam tabel-tabel tersendiri. Tentu saja *entity* tidak dapat independen satu sama lain. Di dalam sebuah *database*, setiap tabel memiliki sebuah *field* yang memiliki nilai unik untuk setiap baris [21].

MySQL adalah salah satu aplikasi sistem manajemen *database* relasional yang handal dalam mengelola *databases* yang sederhana maupun komplek. *MySQL* mempunyai dua macam lisensi yang dikeluarkan oleh *MySQL AB*, suatu perusahaan Swedia, lisensi tersebut yaitu :

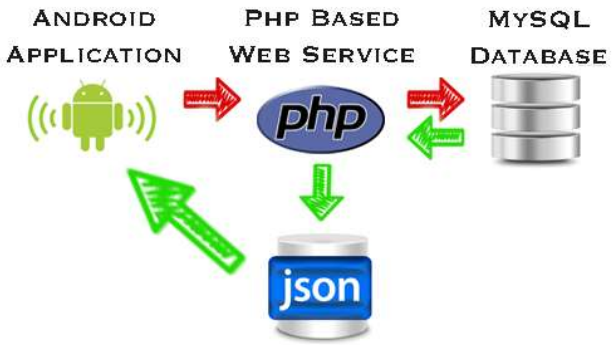
1. Open Source software : MySQL tersedia via GNU GPL (General Public License) untuk yang gratis.
2. *Commercial License* : tersedia bagi siapa saja yang menyukai GPL, jika ingin mengembangkan dan menggunakan *MySQL* sebagai bagian dari *software* produk baru maka pengembang harus membeli *license commercial* ini. Beberapa keunggulan dari *databases MySQL* diantaranya:
   a. Cepat : tujuan utama dari pengembangan *MySQL* adalah kecepatan dalam mengakses dan mengolah *databases*.
   b. Tidak mahal : dibawah *Open Source software license* maka

siapapun dapat menggunakan aplikasi *MySQL* secara gratis.

c. Mudah digunakan : kita dapat membangun dan berinteraksi dengan *databases* MySQL cukup dengan pernyataan sederhana didalam bahasa SQL.

d. Dapat berjalan pada beberapa sistem operasi : seperti Windows, Linux, Mac OS, Unix (solaris, AIX, DEC unix) FreeBSD, OS/2, Irix, dan lainnya.

3. Aman : MySQL adalah sistem oktorisasi fleksibel yang mengijinkan beberapa atau semua privilege *databases* untuk pengguna khusus atau kelompok pengguna [21].



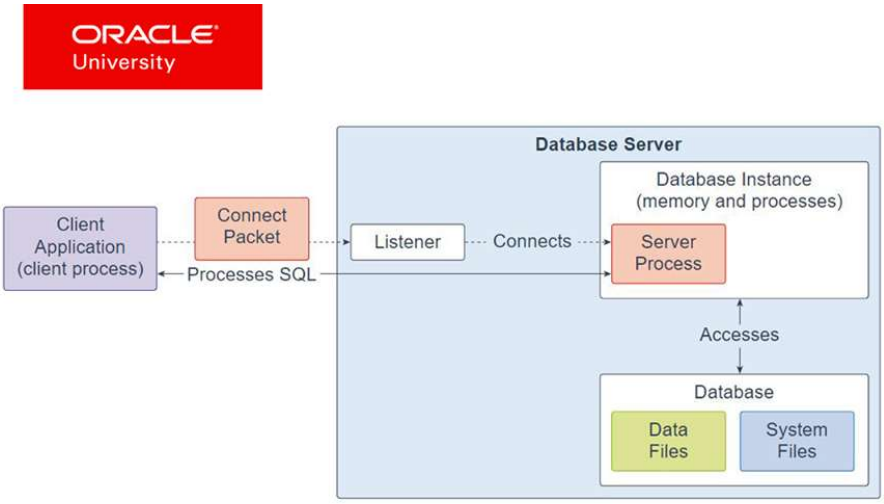
Gambar 2.27 Arsitektur Sistem MySQL

Oracle adalah relational database management system (RDBMS) untuk mengelola informasi secara terbuka, komprehensif dan terintegrasi.Oracle Server menyediakan solusi yang efisien dan efektif karenakemampuannya dalam hal sebagai berikut:

1. Dapat bekerja di lingkungan client/server (pemrosesan tersebar)
2. Menangani manajemen space dan basis data yang besar
3. Mendukung akses data secara simultan
4. Performansi pemrosesan transaksi yang tinggi
5. Menjamin ketersediaan yang terkontrol

Lingkungan yang terreplikasi Database merupakan salah satu komponen dalam teknologi informasiyang mutlak diperlukan oleh semua organisasi yang ingin mempunyai suatusistem informasi yang terpadu untuk menunjang kegiatan

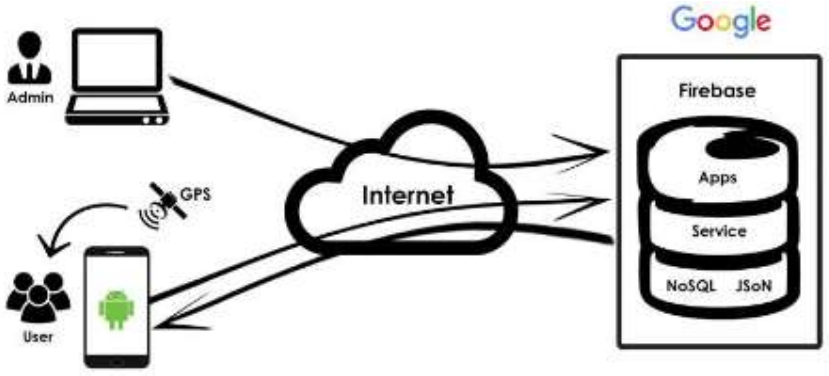
organisasi demimencapai tujuannya. Karena pentingnya peran database dalam sisteminformasi, tidaklah mengherankan bahwa terdapat banyak pilihan software Database Management System (DBMS) dari berbagai vendor baik yang gratis maupun yang komersial. [22]



Gambar 2.28 Arsitektur Sistem Oracle

Firebase adalah API yang disediakan google untuk penyimpanan dan penyelarasan data ke dalam aplikasi Android, iOS, atau web. Realtime database adalah salah satu fasilitas yang menyimpan data ke database dan mengambil data darinya dengan sangat cepat tetapi firebase bukan hanya realtime database, jauh lebih dari itu. Firebase memiliki banyak fitur seperti authentication, database, storage, hosting, pemberitahuan dan lain-lain.

Firebase Realtime Database adalah database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung. Ketika Anda membuat aplikasi lintas-platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah instance Realtime Database dan menerima update data terbaru secara otomatis.[23]



Gambar 2.29 Arsitektur Sistem Firebase

2.11 Software

Software yang akan digunakan dalam penelitian ini seperti Bahasa Pemrograman C, PHP, JAVA seperti berikut.

2.11.1 Hypertext Preprocessor (PHP)

PHP (akronim dari PHP: *Hypertext Preprocessor*) adalah bahasa pemrograman yang berfungsi untuk membuat website dinamis maupun aplikasi web. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan database, file dan folder, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah website. Blog, Toko Online, CMS, Forum, dan Website Social Networking adalah contoh aplikasi web yang bisa dibuat oleh PHP. PHP adalah bahasa scripting, bukan bahasa tag-based seperti HTML. PHP termasuk bahasa yang cross-platform, ini artinya PHP bisa berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam file plain text (teks biasa) dan mempunyai akhiran ".php". Untuk dapat berjalan, PHP membutuhkan web server, yang bertugas untuk memproses file-file php dan mengirimkan hasil pemrosesan untuk ditampilkan di browser client. Oleh karena itu, PHP termasuk server-side scripting (script yang diproses di sisi server).

```
<?php
    Echo "Hello World";
?>
```

Gambar 2.30 Contoh Bahasa PHP

2.11.2 Bahasa Pemrograman C

C secara umum merupakan bahasa pemrograman terstruktur. Instruksi - instruksinya terdiri dari istilah ekspresi aljabar, ditambah dengan kata kunci bahasa Inggris tertentu seperti *if*, *else*, *for*, *do* dan *while*. Dalam hal ini C menyerupai bahasa tingkat tinggi lainnya seperti pascal dan fortran. Bahasa pemrograman C juga mengandung fitur tambahan tertentu yang memungkinkannya digunakan pada level yang lebih rendah, lebih dekat ke bahasa mesin komputer. C pertama kali dikembangkan oleh Dennis Ritchie dan Brian Kerighan di Bell Telephone Laboratories Inc (sekarang AT&T Bel Laboratories) pada 1970-an. C sebagian besar dikembangkan pada Bell Labs hingga 1978, ketika Brian Kerighan dan Dennis Ritchie menerbitkan deskripsi definitif bahasa, "*The C Programming Language, Prentice-Hall*" tahun 1978. Versi C yang sesuai dengan standar ANSI dikenal sebagai ANSI C. Tetapi kebanyakan kompiler C yang kompatibel dengan ANSI juga memiliki fitur-fitur khusus yang ditambahkan, berbeda dengan yang tidak kompatibel dengan ANSI. C memiliki set instruksi yang relatif kecil, 32 kata kunci dalam versi bahasa yang paling umum, tetapi hal ini bisa menjadi komentar dengan fungsi *library* yang luas yang dapat dibangun oleh programmer. Salah satu fitur menarik dari bahasa C adalah bahwa C dapat dikompilasi untuk menghasilkan kode yang dapat dieksekusi dengan sangat cepat dan saling terhubung. Hal ini memungkinkannya digunakan untuk memprogram mikrokontroller, yang biasanya memiliki sedikit memori.[24]

```c
#include <stdio.h>
int main() {
  // printf() displays the string inside quotation
  printf("Hello, World!");
  return 0;
}
```

Gambar 2.31 Contoh Bahasa Pemrograman C

2.11.3 Java

Menurut Budi Raharjo, Imam Heryanto, Arif haryono (Mudah Belajar Java 2010) java adalah bahasa pemrograman yang dapat dijalankan di berbagai

komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan Java mampu berjalan di beberapa platform sistem operasi yang berbeda, Java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun".[25]

```
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Gambar 2.32 Contoh Java

2.11.4 Java Script Object Nation (JSON)

Java Script Object Notation (JSON) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi 3 Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dan lain-lain. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data.

JSON terbuat dari dua struktur yaitu:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*),

kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.

2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*) .

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data JSON.[26]

```
myObj = {
  "name":"John",
  "age":30,
  "cars": {
    "car1":"Ford",
    "car2":"BMW",
    "car3":"Fiat"
  }
 }
```

Gambar 2.33 Contoh JSON

2.12 Metode Pengujian

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean[25]. Sejumlah aturan yang berfungsi sebagai sasaran pengujian pada perangkat lunak adalah :

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
2. Test case yang baik adalah test case yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya .

Karakteristik umum dari pengujian perangkat lunak adalah sebagai berikut :

1. Pengujian dimulai pada level modul dan bekerja keluar kearah integrasi pada sistem berbasiskan komputer.
2. Teknik pengujian yang berbeda sesuai dengan poin-poin yang berbeda pada waktunya.
3. Pengujian diadakan oleh software developer dan untuk proyek yang besar oleh group testing yang independent.
4. Testing dan Debugging adalah aktivitas yang berbeda tetapi debugging harus diakomodasikan pada setiap strategi testing .[27]

Metode pengujian perangkat lunak ada 3 jenis, yaitu :

1. *White Box/Glass Box* - pengujian operasi.
2. *Black Box* - untuk menguji sistem.
3. *Use case* - untuk membuat input dalam perancangan black box dan pengujian *statebased* .[27]

2.12.1 White Box Testing

Pengujian *white box* adalah pengujian yang meramalkan cara kerja perangkat lunak secara rinci, karenanya logikal *path* (jalur logika) perangkat lunak akan di-test dengan menyediakan test case yang akan mengerjakan kumpulan kondisi atau pengulangan secara spesifik. Secara sekilas dapat diambil kesimpulan *white box* testing merupakan petunjuk untuk mendapatkan program yang benar secara 100%. Menurut Pressman [28], pengujian *White box* atau *Glass box* adalah metode *test case* desain yang menggunakan struktur kontrol desain *procedural* untuk memperoleh *test-case*. Dengan menggunakan metode pengujian *white box,* analis sistem akan dapat memperoleh *test case* yang:

a. Memberikankan jaminan bahwa semua jalur *independent* pada suatu modul telah digunakan paling tidak satu kali.
b. Menggunakan semua keputusan logis dari sisi *true* dan *false.*
c. Mengeksekusi semua batas fungsi *loops* dan batas operasionalnya.
d. Menggunakan struktur *internal* untuk menjamin validitasnya.

Ujicoba basis *path* adalah teknik uji coba *white box* yang diusulkan Tom Mc Cabe. Metode ini memungkinkan perancang *test case* mendapatkan ukuran kekompleksan logical dari perancangan prosedural dan menggunkan ukuran ini

sebagai petunjuk untuk mendefinisikan basis set dari jalur pengerjaan. *Test case* yang didapat digunakan untuk mengerjakan basis set yang menjamin pengerjaan setiap perintah minimal satu kali selama uji coba.

Terdapat beberapa proses yang harus di lakukan dalam uji coba basis *path* yaitu diantaranya :

1. Notasi Diagram Alir

   Sebelum metode basis *path* diperkenalkan, terlebih dahulu akan dijelaskan mengenai notasi sederhana dalam bentuk diagram alir (grafik alir). Diagram alir menggambarkan aliran kontrol logika yang menggunakan notasi.

2. Kompleksitas Siklomatis

   Kompleksitas siklomatis adalah metriks perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program. Bila metriks ini digunakan dalam konteks metode pengujian basis *path*, maka nilai yang terhitung untuk kompleksitas siklomatis menentukan jumlah jalur independen dalam basis set suatu pemrograman memberi batas atas bagi jumlah pengujian yang harus dilakukan untuk memastikan bahwa semua statemen telah dieksekusi sedikitnya satu kali. Jalur independen adalah jalur yang memlalui progarm yang mengintroduksi sedikitnya satu rangkaian statemen proses baru atau suatu kondisi baru. Bila dinyatakan dengan terminologi grafik alir, jalur independen harus bergerak sepanjang paling tidak satu *edge* yang tidak dilewatkan sebelum jalur tersebut ditentukan.

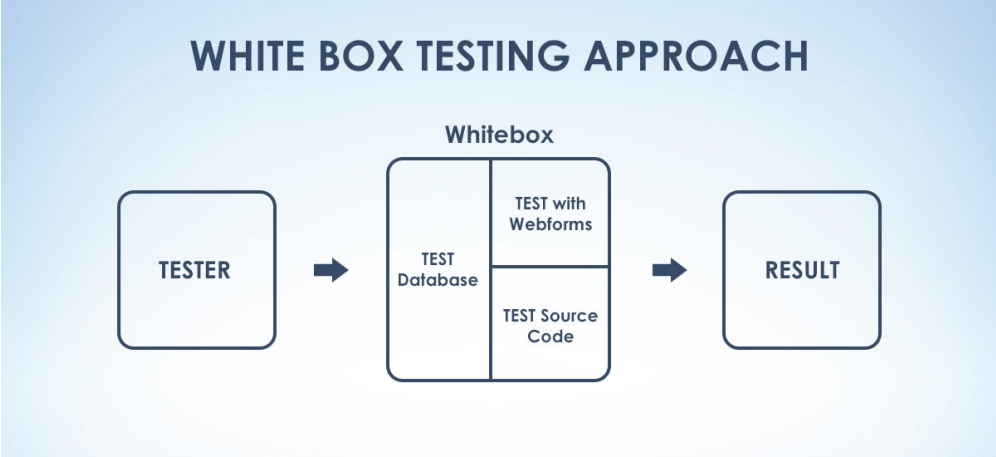
3. Melakukan Test Case

   Metode uji coba basis *path* juga dapat diterapkan pada perancangan prosedural rinci atau program sumber. Pada bagian ini akan dijelaskan langkah-langkah uji coba basis *path*.

4. Matriks Grafis

   Prosedur untuk mendaptakan grafik alir dan menentukan serangkaian basis *path*, cocok dengan mekanisasi. Untuk mengembangkan peranti

perangkat lunak yang membantu pengujian basis *path*, struktur data yang disebut matriks grafis dapat sangat berguna.



Gambar 2.34 White Box Testing

2.12.2 Black Box Testing

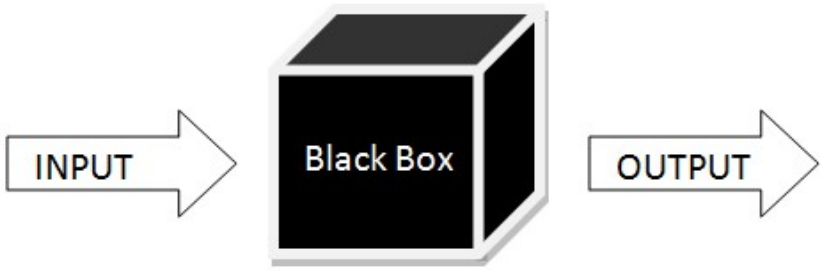
Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian *black box* atau bisa disebut pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan menggunakan perangkat lunak, serta dilihat apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian *black box testing* harus dibuat dengan kasus benar dan kasus salah. *Black box* testing juga disebut pengujian tingkah laku, memusat pada kebutuhan fungsional perangkat lunak. Teknik pengujian *black box* memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data (pengaksesan basis data), kesalahan performasi, kesalahan inisialisasi dan akhir program. *Equivalence Partitioning* merupakan metode *black box* testing yang membagi domain masukan dari program kedalam kelas-kelas sehingga *test case* dapat diperoleh. *Equivalence Partitioning* berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji

yang harus dibuat. Kasus uji yang didesain untuk *equivalence partitioning* berdasarkan pada evaluasi dari kelas ekuivalensi untuk kondisi masukan yang menggambarkan kumpulan keadaan yang valid atau tidak. Kondisi masukan dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi boolean .[27]

Pengujian *black box* berusaha menemukan kesalahan dalam kategori :

1. Fungsi – fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.[27]



Gambar 2.35 Black Box Testing