

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

Landasan teori merupakan definisi, konsep yang telah disusun secara sistematis dan dasar yang kuat dalam sebuah penelitian. Landasan teori yang digunakan dalam penyusunan Pembangunan *early warning security system* menggunakan Arduino berbasis Android pada mobil meliputi Aplikasi Mobile, Aplikasi Mobile, Pengertian Mobile, Karakteristik Perangkat Mobile, Android, Basis Data, SQL, Pemrograman Berorientasi Objek, Konsep Dasar Berorientasi Objek, Arduino, Sensor PIR, PHP, Mysql, *Unified Modeling Language* (UML), Web Service.

##### **2.1.1 Early Warning Security System**

*Early Warning Security System* jika diterjemahkan ke dalam bahasa Indonesia yaitu sistem keamanan peringatan dini. Sistem keamanan peringatan dini terbagi menjadi 4 suku kata yaitu sistem, kemanan, peringatan, dan dini.

Menurut James Havery [12] Sistem adalah prosedur logis dan rasional untuk merancang suatu rangkaian komponen yang berhubungan satu dengan yang lainnya dengan maksud untuk berfungsi sebagai suatu kesatuan dalam usaha mencapai suatu tujuan yang telah ditentukan.

Menurut Peter Salim [13] Keamanan berasal dari kata pokok "aman" yang berarti bebas, terlindung dari bahaya, selamat, tidak membahayakan, yakin, dapat dipercaya, dapat diandalkan. Sedangkan "keamanan" memiliki arti "suasana aman" ketenteraman, ketenangan.

Menurut Kamus Besar Bahasa Indonesia[14] peringatan dapat diartikan sebagai suatu teguran untuk mengingatkan berita yang mengingatkan akan adanya sesuatu yang akan terjadi.

Menurut Kamus Besar Bahasa Indonesia[15] dini dapat diartikan sebelum waktunya.

Maka *Early Warning Security System* dapat diartikan suatu mekanisme yang dirancang untuk memberikan peringatan keamanan kepada user dengan memberikan peringatan sedini mungkin.

### **2.1.2 Aplikasi Mobile**

Menurut Buyens [16] aplikasi mobile berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat diartikan sebagai perpindahan dari suatu tempat ke tempat yang lain.

Maka aplikasi *mobile* dapat diartikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah – pindah dari satu tempat ke tempat yang lain serta mempunyai ukuran yang kecil. Aplikasi *mobile* ini dapat diakses melalui perangkat nirkabel, pager, PDA, telepon seluler, *smartphone*, dan perangkat sejenisnya.

### **2.1.3 Karakteristik Perangkat Mobile**

Perangkat mobile memiliki banyak jenis dalam hal ukuran, desain dan layout, tetapi mereka memiliki karakteristik yang sangat berbeda dari sistem desktop. Berikut karakteristik perangkat mobile, diantaranya :

#### **a. Ukuran yang kecil**

Perangkat mobile memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.

#### **b. Memory yang terbatas**

Perangkat mobile juga memiliki memory yang kecil, yaitu primary (RAM) dan secondary (disk). Pembatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan jumlah dari memory, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.

c. Daya proses yang terbatas

Sistem mobile tidaklah setangguh rekan mereka yaitu desktop. Ukuran, teknologi dan biaya adalah beberapa faktor yang mempengaruhi status dari sumber daya ini. Seperti harddisk dan RAM, Anda dapat menemukan mereka dalam ukuran yang pas dengan sebuah kemasan kecil.

d. Mengkonsumsi daya yang rendah

Perangkat mobile menghabiskan sedikit daya dibandingkan dengan mesin desktop. Perangkat ini harus menghemat daya karena mereka berjalan pada keadaan dimana daya yang disediakan dibatasi oleh baterai-baterai.

e. Kuat dan dapat diandalkan

Karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

f. Konektivitas yang terbatas

Perangkat mobile memiliki bandwidth rendah, beberapa dari mereka bahkan tidak tersambung. Kebanyakan dari mereka menggunakan koneksi wireless.

g. Masa hidup yang pendek

Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala. Coba ambil kasus sebuah handphone, mereka booting dalam hitungan detik dan kebanyakan orang tidak mematikan handphone mereka bahkan ketika malam hari. PDA akan menyala jika anda menekan tombol power mereka.

#### **2.1.4 Android**

Android merupakan sistem operasi mobile yang tumbuh di tengah sistem operasi lainnya yang berkembang dewasa ini. Sistem operasi lainnya seperti Windows Mobile, i-Phone OS, Symbian dan masih banyak lagi juga menawarkan kekayaan isi dan keoptimalan berjalan di atas perangkat hardware yang ada. Akan

tetapi, sistem operasi yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

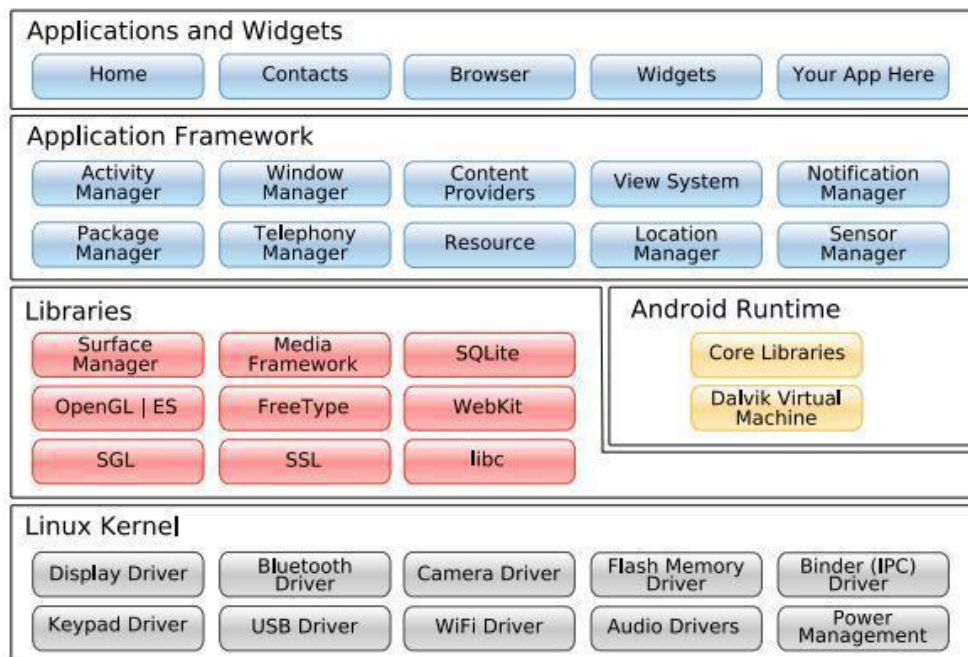
Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. Application Programming Interface (API) yang disediakan menawarkan akses ke hardware, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga.[17]

#### **2.1.4.1        *Arsitektur Android***

Arsitektur Android dapat digambarkan seperti pada gambar 1 dan secara garis besar Arsitektur Android dapat dijelaskan sebagai berikut:

##### **1.        Application dan Widgets**

Application dan Widgets ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Hampir semua aplikasi ditulis menggunakan bahasa pemrograman Java.



**Gambar 2.1 Arsitektur Android**

*Sumber Gambar : N. Safaat [17]*

## 2. Application Framework

Android adalah “Open Development Platform” yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resource, menjalankan service background, mengatur alarm, dan menambah status notifications, dan sebagainya. Pengembang memiliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (reuse). Sehingga bisa kita simpulkan Application Frameworks ini adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti content providers yang berupa sms dan panggilan telepon. Komponen-komponen yang termasuk di dalam Application Frameworks adalah sebagai berikut :

- 1) Views
- 2) Content Provider
- 3) Resource Manager
- 4) Notification Manager
- 5) Activity Manager [17]

### 3. Libraries

Libraries ini adalah layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan di atas Kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc SSL, serta :

- 1) Libraries media untuk pemutaran media audio dan video
  - 2) Libraries untuk manajemen tampilan
  - 3) Libraries Graphics mencakup 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 mencakup modern web browser dengan engine embedded webview
  - 7) Libraries 3D yang mencakup implementasi OpenGL ES1.0 API's.
- [17]

### 4. Android Run Time

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. Dalvik Virtual Machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android Run Time dibagi menjadi dua bagian yaitu:

- 1) Core Libraries: Aplikasi Android dibangun dalam bahasa Java, sementara Dalvik sebagai virtual mesinnya bukan Virtual Machine 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, dimana merupakan pengembangan yang mampu membuat Linux Kernel untuk melakukan threading dan manajemen tingkat rendah.  
[1]

## 5. Linux Kernel

Linux Kernel adalah layer dimana inti dari system operasi Android itu berada. Berisi file-file system yang mengatur sistem processing, memory, resource, drivers, dan sistem-sistem operasi Android lainnya. Linux Kernel yang digunakan Android adalah Linux Kernel release 2.6.

### 2.1.5 Definisi Mikrokontroler

Menurut Setiawan [18] Mikrokontroller adalah suatu IC dengan kepadatan yang sangat tinggi, dimana semua bagian yang diperlukan untuk suatu kontroler sudah dikemas dalam satu keping, biasanya terdiri dari CPU (*Central Processing Unit*), RAM (*Random Access Memory*), EEPROM/EPROM/PROM/ROM, I/O, Serial & Parallel, *Timer*, *Interrupt Controller*.

Menurut Ardi Winoto[19] “Mikrokontroler adalah Sebuah sistem microprocessor dimana didalamnya sudah terdapat CPU, ROM, RAM, I/O, clock dan peralatan internal lainnya yang sudah terhubung dan terorganisasi dengan baik oleh pabrik pembuatannya dan dikemas dalam satu chip yang siap pakai, sehingga kita tinggal memprogram isi ROM sesuai dengan aturan penggunaan oleh pabrik pembuatannya”.

Berdasarkan definisi yang dikemukakan diatas dapat disimpulkan bahwa mikrokontroller adalah suatu IC yang didesain atau dibentuk dengan kepadatan yang sangat tinggi, dimana semua bagian yang diperlukan suatu kontroler sudah dikemas dalam satu keping, biasanya terdiri dari CPU (*Central Processing Unit*), RAM (*Random Access Memory*), EEPROM/EPROM/PROM/ROM, I/O, Serial & Parallel, *Timer*, *Interrupt*

*Controller* dan berfungsi sebagai pengontrol rangkaian elektronik serta umumnya dapat menyimpan program didalamnya.

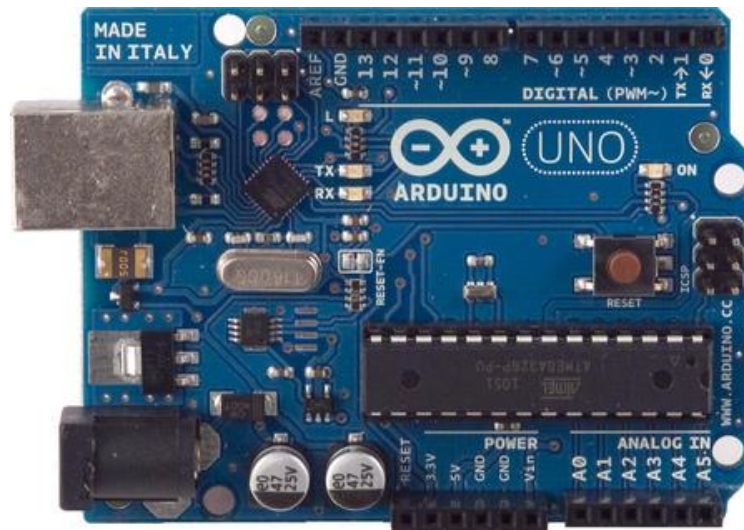
#### **2.1.6 Definisi Arduino**

Arduino merupakan rangkaian elektronik yang bersifat *open source*, serta memiliki perangkat keras dan lunak yang mudah untuk digunakan. Arduino dapat mengenali lingkungan sekitarnya melalui berbagai jenis sensor dan dapat mengendalikan lampu, motor, dan berbagai jenis aktuator lainnya. Arduino mempunyai banyak jenis, di antaranya Arduino Uno, Arduino Mega 2560, Arduino Fio, dan lainnya.

##### **2.1.6.1 *Arduino UNO***

Arduino adalah sebuah board mikrokontroller yang berbasis ATmega328. Arduino memiliki 14 pin input/output yang mana 6 pin dapat digunakan sebagai output PWM, 6 analog input, crystal osilator 16 MHz, koneksi USB, jack power, kepala ICSP, dan tombol reset. Arduino mampu *support* mikrokontroller; dapat dikoneksikan dengan komputer menggunakan kabel USB. [20]





**Gambar 2.2 Board Arduino UNO**

*Sumber Gambar : [www.arduino.cc](http://www.arduino.cc) [20]*

Menurut Feri Djuandi [21] Arduino adalah merupakan sebuah board minimum system mikrokontroler yang bersifat open source. Didalam rangkaian board arduino terdapat mikrokontroler AVR seri ATmega 328 yang merupakan produk dari Atmel.

Arduino memiliki kelebihan tersendiri disbanding board mikrokontroler yang lain selain bersifat open source, arduino juga mempunyai bahasa pemrogramanya sendiri yang berupa bahasa C. Selain itu dalam *board* arduino sendiri sudah terdapat *loader* yang berupa USB sehingga memudahkan kita ketika kita memprogram mikrokontroler didalam arduino. Sedangkan pada kebanyakan *board* mikrokontroler yang lain yang masih membutuhkan rangkaian *loader* terpisah untuk memasukkan program ketika kita memprogram mikrokontroler. Port USB tersebut selain untuk *loader* ketika memprogram, bisa juga difungsikan sebagai port komunikasi serial.

Arduino menyediakan 20 pin I/O, yang terdiri dari 6 pin input analog dan 14 pin digital input/output. Untuk 6 pin analog sendiri bisa juga difungsikan sebagai output digital jika diperlukan output digital tambahan selain 14 pin yang sudah tersedia. Untuk mengubah pin analog menjadi digital cukup mengubah konfigurasi pin pada program. Dalam *board* kita bisa lihat pin digital

diberi keterangan 0-13, jadi untuk menggunakan pin analog menjadi output digital, pin analog yang pada keterangan board 0-5 kita ubah menjadi pin 14-19. dengan kata lain pin analog 0-5 berfungsi juga sebagai pin output digital 14-16.

Sifat *open source* arduino juga banyak memberikan keuntungan tersendiri untuk kita dalam menggunakan *board* ini, karena dengan sifat *open source* komponen yang kita pakai tidak hanya tergantung pada satu merek, namun memungkinkan kita bisa memakai semua komponen yang ada dipasaran.

Bahasa pemrograman arduino merupakan bahasa C yang sudah disederhanakan syntax bahasa pemrogramannya sehingga mempermudah kita dalam mempelajari dan mendalami mikrokontroler.

Mikrokontroler	ATmega 328
Tegangan Pengoperasian	5 V
Tegangan Input yang disarankan	7 – 12 V
Batas Tegangan Input	6 – 20 V
Jumlah pin I/O digital	14 pin digital (6 diantaranya menyediakan keluaran PWM)
Jumlah pin input Analog	6 pin
Arus DC tiap pin I/O	40mA
Arus DC untuk pin 3,3 V	50mA
Memori Flash	32 KB (ATmega 328) sekitar 0,5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega 328)
EPROM	1 KB (ATmega 328)
Clock Speed	16 MHz

**Gambar 2.3 Deskripsi Arduino UNO**

*Sumber Gambar : [www.arduino.cc](http://www.arduino.cc) [20]*

#### **2.1.6.2 Power**

Arduino dapat diberikan *power* melalui koneksi USB atau *power supply*. *Power*nya diselekt secara otomatis. *Power supply* dapat menggunakan adaptor DC atau baterai. Adaptor dapat dikoneksikan dengan mencolok *jack* adaptor pada koneksi port input *supply*.

*Board* Arduino dapat dioperasikan menggunakan *supply* dari luar sebesar 6 - 20 volt. Jika *supply* kurang dari 7V, kadangkala pin 5V akan menyuplai kurang dari 5 volt dan *board* bisa menjadi tidak stabil. Jika menggunakan lebih dari 12 V, tegangan di regulator bisa menjadi sangat panas dan menyebabkan kerusakan pada *board*. Rekomendasi tegangan ada pada 7 sampai 12 volt.

Penjelasan pada pin power adalah sebagai berikut :

- 1) **Vin**  
Tegangan input ke board arduino ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 volt dari koneksi USB atau tegangan yang diregulasikan). Pengguna dapat memberikan tegangan melalui pin ini, atau jika tegangan suplai menggunakan *power jack*, aksesnya menggunakan pin ini.
- 2) **5V**  
Regulasi *power supply* digunakan untuk power mikrokontroller dan komponen lainnya pada board. 5V dapat melalui Vin menggunakan regulator pada board, atau supply oleh USB atau *supply* regulasi 5V lainnya.
- 3) **3V3**  
Suplai 3.3 volt didapat oleh FTDI chip yang ada di board. Arus maximumnya adalah 50mA
- 4) **Pin Ground**  
berfungsi sebagai jalur ground pada arduino
- 5) **Memori**  
ATmega328 memiliki 32 KB flash memori untuk menyimpan kode, juga 2 KB yang digunakan untuk bootloader. ATmega328 memiliki 2 KB untuk SRAM dan 1 KB untuk EEPROM.

### **2.1.6.3      *Input dan Output***

Setiap 14 pin digital pada arduino dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Input/output dioperasikan pada 5 volt. Setiap pin dapat menghasilkan atau

menerima maximum 40 mA dan memiliki internal pull-up resistor (disconnected oleh default) 20-50K Ohm.

Beberapa pin memiliki fungsi sebagai berikut :

- 1) Serial : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) TTL data serial. Pin ini terhubung pada pin yang koresponding dari USB ke TTL chip serial.
- 2) Interrupt eksternal : 2 dan 3. Pin ini dapat dikonfigurasi untuk trigger sebuah interap pada low value, rising atau falling edge, atau perubahan nilai.
- 3) PWM : 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit output PWM dengan fungsi `analogWrite()`.
- 4) SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mensupport komunikasi SPI, yang mana masih mendukung hardware, yang tidak termasuk pada bahasa arduino.
- 5) LED : 13. Ini adalah dibuat untuk koneksi LED ke digital pin 13. Ketika pin bernilai HIGH, LED hidup, ketika pin LOW, LED mati.

#### **2.1.6.4      *Komunikasi***

Uno Arduino memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). *Firmware* Arduino menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file. Ini diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board* Arduino. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer.

#### **2.1.6.5      *Software Arduino***

Arduino Uno dapat diprogram dengan perangkat lunak Arduino. Pada ATmega328 di Arduino terdapat *bootloader* yang memungkinkan Anda

untuk meng-*upload* kode baru untuk itu tanpa menggunakan *programmer hardware eksternal*.

IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
2. *Compiler*, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
3. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memory didalam papan Arduino.
4. *Sketch*, Sebuah kode program Arduino umumnya disebut dengan istilah *sketch*. Kata “*sketch*” digunakan secara bergantian dengan “kode program” dimana keduanya memiliki arti yang sama. [20]



**Gambar 2.4 Tampilan IDE Arduino Dengan Sebuah Sketch**

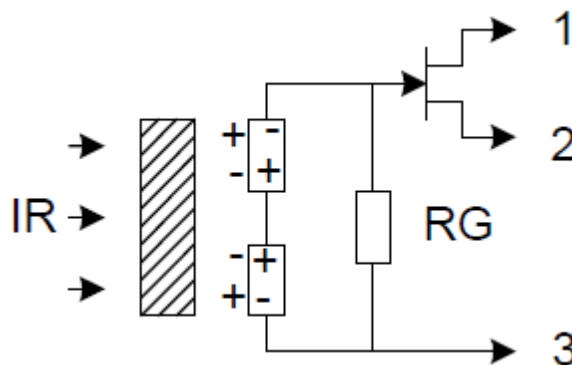
*Sumber Gambar: [www.arduino.cc](http://www.arduino.cc) [20]*

### 2.1.7 Sensor PIR (Passive Infra Red)

Sensor adalah komponen yang mengubah besaran fisis menjadi besaran listrik Franky chandra dan Deni Arifianto[22]. Sensor yang digunakan pada sistem ini adalah Sensor PIR.

PIR merupakan sebuah sensor berbasis infrared. Akan tetapi, tidak seperti sensor infrared kebanyakan yang terdiri dari IR LED dan fototransistor. PIR tidak memancarkan apapun seperti IR LED. Sesuai namanya “Passive”, sensor ini hanya merespon energi dari pancaran sinar inframerah pasif yang dimiliki oleh setiap benda yang terdeteksi olehnya. Benda yang dapat dideteksi oleh sensor ini biasanya adalah tubuh manusia (Dian Renita Rahmalia et al., 2012).

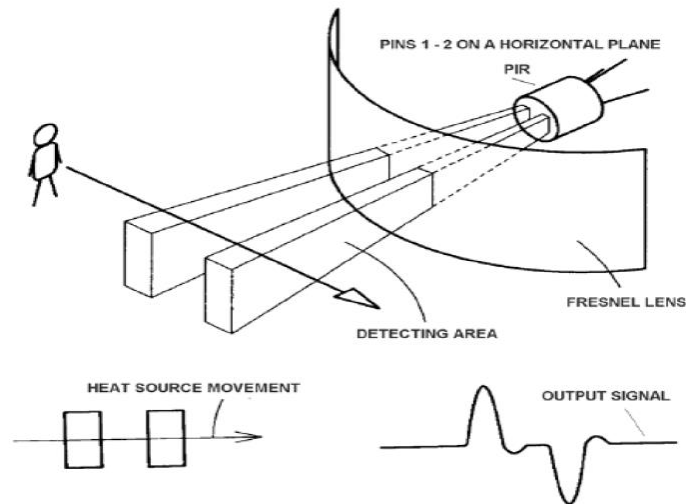
Menurut Mohd. Syaryadhi et al.,( 2007) PIR sensor mempunyai dua elemen sensing yang terhubung dengan masukan dengan susunan seperti yang terdapat dalam Gambar berikut:



**Gambar 2.5 Diagram Internal Sensor PIR**

*Sumber Gambar: [www.arduino.cc](http://www.arduino.cc) [20]*

Jika ada sumber panas yang lewat di depan sensor tersebut, maka sensor akan mengaktifkan sel pertama dan sel kedua sehingga akan menghasilkan bentuk gelombang seperti ditunjukkan dalam Gambar 2.2 Sinyal yang dihasilkan sensor PIR mempunyai frekuensi yang rendah yaitu 0,2 – 5 Hz..



**Gambar 2.6 Arah Jangkauan Gelombang Sensor PIR (Passive Infra Red)**

*Sumber Gambar: [www.arduino.cc](http://www.arduino.cc) [20]*

Benda yang dapat memancarkan panas berarti memancarkan radiasi infra merah. Benda – benda ini termasuk makhluk hidup seperti binatang dan tubuh manusia. Tubuh manusia dan binatang dapat memancarkan radiasi infra merah terkuat yaitu pada panjang gelombang  $9,4 \mu\text{m}$ . Radiasi infra merah yang dipancarkan inilah yang menjadi sumber pendeteksian bagi detektor panas yang memanfaatkan radiasi infra merah. (Mohd. Syaryadhi et al., 2007).

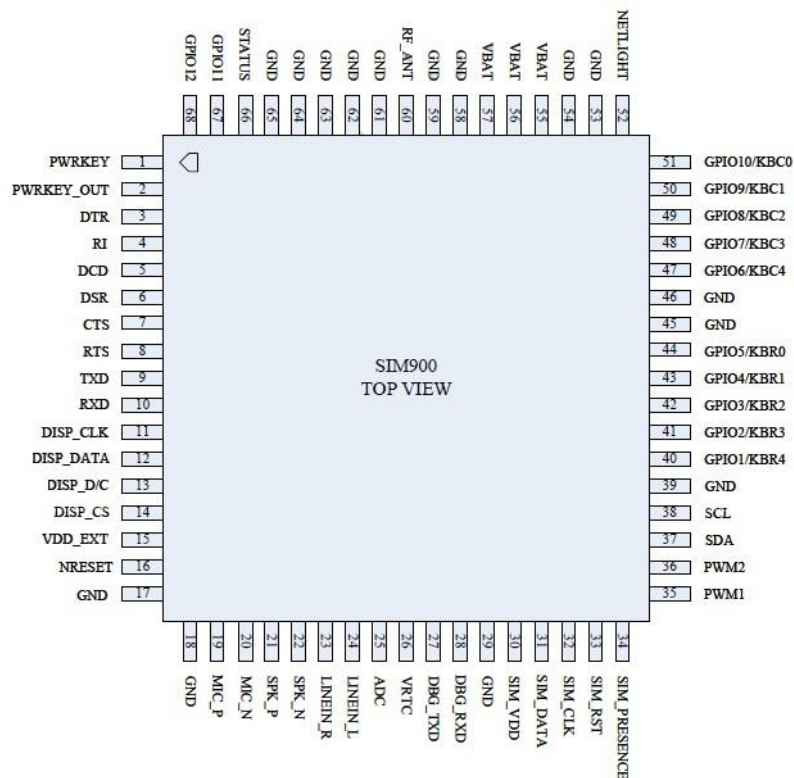
Deni Arifianto, (2011) menyebutkan modul sensor PIR memiliki karakterisasi sebagai berikut :

- 1) Tegangan Catu Daya : 4.7 – 12 VDC
- 2) Jangkauan Deteksi Sensor : 5 meter pada sudut 0 derajat
- 3) Output sensor tegangan High : 5 VDC
- 4) Output lebar pulsa : 0.5 s

### 2.1.8 Modul SIM 900A

SIM900A adalah modul SIM yang digunakan pada penelitian ini. Modul SIM900 GSM/GPRS adalah bagian yang berfungsi untuk komunikasi antara mikrokontroler Arduino dengan Web Service.

Modul komunikasi GSM/GPRS menggunakan core IC SIM900A. Modul ini mendukung komunikasi *dual band* pada frekuensi 900 / 1800 MHz (GSM900 dan GSM1800) sehingga fleksibel untuk digunakan bersama kartu SIM dari berbagai operator telepon seluler di Indonesia. Operator GSM yang beroperasi di frekuensi *dual band* 900 MHz dan 1800 MHz sekaligus: Telkomsel, Indosat, dan XL. Operator yang hanya beroperasi pada band 1800 MHz: Axis dan Three.



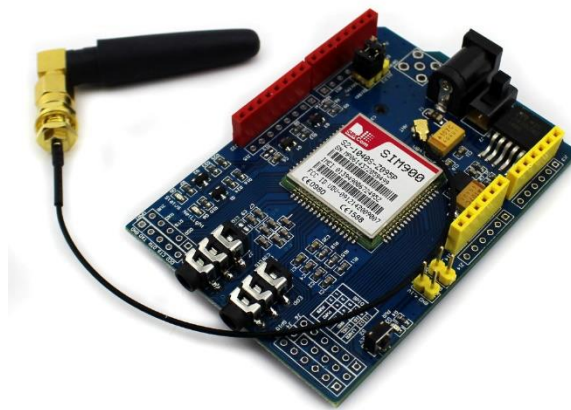
**Gambar 2.7 Layout dan Pin-pin dari Modul SIM900**

Sumber Gambar : <http://www.open-electronics.org/>

Pada gambar 2.7 merupakan tampilan dari konfigurasi pin GSM SIM900. Modul ini sudah terpasang pada *breakout-board* (modul inti dikemas dalam SMD/ *Surface Mounted Device packaging*) dengan *pin header* standar 0,1"



(2,54 mm) sehingga memudahkan penggunaan, bahkan bagi penggemar elektronika pemula sekalipun. Modul GSM SIM900 ini juga disertakan antena GSM yang kompatibel dengan produk ini. Pada gambar 2.9 dapat dilihat tampilan dari modul GSM SIM900 yang dilengkapi dengan antena.



**Gambar 2.8 Modul SIM900**

*Sumber Gambar : <http://www.open-electronics.org/>*

Spesifikasi modul GSM SIM900A :

1. GPRS multi-slot class 10/8, kecepatan transmisi hingga 85.6 kbps (*downlink*), mendukung PBCCH, PPP *stack*, skema penyandian CS 1,2,3,4
2. GPRS mobile station class B
3. Memenuhi standar GSM 2/2 +
4. Class 4 (2 W @ 900 MHz)
5. Class 1 (1 W @ 1800MHz)
6. SMS (Short Messaging Service): point-to-point MO & MT, SMS cell *broadcast*, mendukung format teks dan PDU (*Protocol Data Unit*)
7. Dapat digunakan untuk mengirim pesan MMS (*Multimedia Messaging Service*)
8. Mendukung transmisi faksimili (*fax group 3 class 1*)
9. *Handsfree mode* dengan sirkit reduksi gema (*echo suppression circuit*)
10. Dimensi: 24 x 24 x 3 mm
11. Pengendalian lewat perintah AT (GSM 07.07, 07.05 & SIMCOM Enhanced

AT Command Set)

12. Rentang catu daya antara 7 Volt hingga 12 Volt DC
13. SIM Application Toolkit
14. Hemat daya, hanya mengkonsumsi arus sebesar 1 mA pada moda tidur (*sleep mode*)
15. Rentang suhu operasional: -40 °C hingga +85 °C

#### **2.1.8.1 Cara kerja Modul SIM 900A**

Modul GSM SIM900A dapat bekerja dengan diberi perintah “AT Command”, (AT = Attention). AT Command adalah perintah-perintah standar yang digunakan untuk melakukan komunikasi antara komputer dengan ponsel melalui serial port. Melalui AT Command, data-data yang ada di dalam ponsel dapat diketahui, mulai dari vendor ponsel, kekuatan sinyal, membaca pesan, mengirim pesan, dan lain-lain.

Berikut ini beberapa perintah “AT Command” yang biasa digunakan pada modul GSM SIM900A :

AT+CPBF : cari no telpon

AT+CPBR : membaca buku telpon

AT+CPBW : menulis no telp di buku telpon

AT+CMGF : menyetting mode SMS text atau PDU

AT+CMGL : melihat semua daftar sms yg ada.

AT+CMGR : membaca sms.

AT+CMGS : mengirim sms.

AT+CMGD : menghapus sms.

AT+CMNS : menyetting lokasi penyimpanan ME(hp) atau SM(SIM Card)

AT+CGMI : untuk mengetahui nama atau jenis ponsel

AT+CGMM : untuk mengetahui kelas ponsel

AT+COPS? : untuk mengetahui nama provider kartu GSM

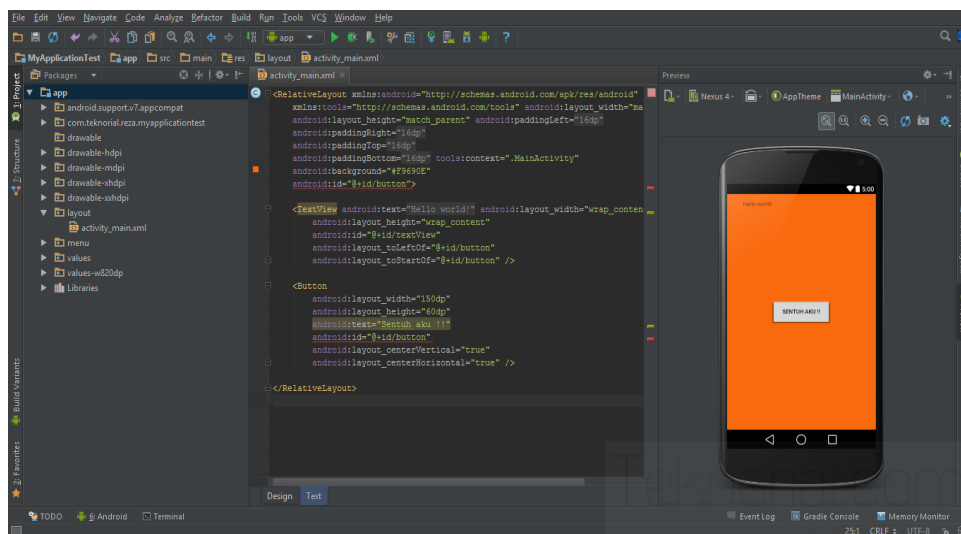
AT+CBC : untuk mengetahui level baterai

AT+CSCA : untuk mengetahui alamat SMS Center

### 2.1.9 Android Studio

Android studio adalah lingkungan pengembangan terpadu (*IDE*) resmi untuk pengembangan platform android, hal itu disampaikan pada tanggal 16 mei 2013 Google I/O, android studio sudah tersedia secara bebas dibawah lisensi Apache 2.0.

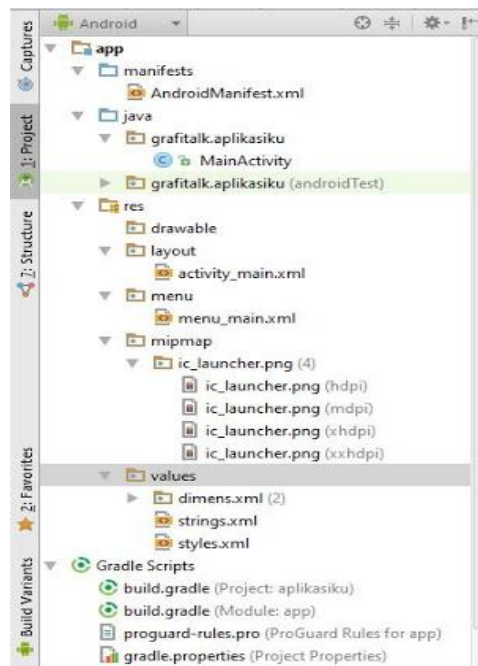
Android studio pada awalnya tahap preview versi 0.1 yang dipakai pada tanggal 1 mei 2013 dan memasuki tahap beta pada bulan juni 2014 dan mulai stabil dirilis pada Desember 2014 dengan versi 1.0 ,berdasarkan JetBrains 'IDEA IntelliJ Software, android studio dirancang khusus untuk pengembangan android yang tersedia untuk Windows, Mac OS X dan Linux sebagai pengganti Eclipse.[23]



**Gambar 2.9 Layout Android Studio**

*Sumber Gambar : <http://www.visualmicro.com> [23]*

### 2.1.9.1 Struktur Android Studio



**Gambar 2.10 Struktur Android Studio**

Sumber Gambar : <http://www.visualmicro.com> [23]

Berikut rincian dari struktur Android Studio :

- 1) App : berfungsi untuk menyimpan file pendukung aplikasi yang akan diolah.
  1. *Manifest*  
tempat dimana konfigurasi konfigurasi pendukung project serta izin (permission) untuk mengakses hardware.
  2. *Java*  
tempat berdiamnya code/scripts java untuk menjalankan aplikasi yang akan dibuat
  3. *Res* :
    - 1) *Drawable* : tempat untuk menyimpan gambar atau icon pendukung aplikasi.
    - 2) *Layout* : tempat berkumpulnya design tampilan yang akan tampak pada tampilan aplikasi anda.

- 3) *Menu* : berisi code berupa file xml yang didalamnya berisi array tentang menu yang akan tampil dalam project anda, misalkan file, new,open,close,dsb.
  - 4) *Mipmap* : tempat berkumpulnya icon launcher (icon utama aplikasi) yang ukurannya menyesuaikan device yang digunakan.
  - 5) *Values* : untuk menyimpan file .xml yang menggambarkan sumber dari berbagai jenis. Sebagai contoh folder berisi file strings.xml dengan definisi sumber string. Seperti yang akan anda lihat dari menambahkan bagian color , folder tata letak juga dapat berisi, misalnya, deskripsi warna. Di dalam file dimens.xml, terdapat string-string untuk konfigurasi dimensi aplikasi. Tidak jauh berbeda dengan dimens.xml maupun strings.xml, pada file styles.xml terdapat deklarasi variable untuk theme.
- 2) Gradle scripts : berfungsi sebagai konfigurasi aplikasi anda dengan device, serta menghubungkan library pendukung aplikasi yang anda buat.

#### **2.1.10 Pemrograman Berbasis Objek**

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek. [24]

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini beragam

dengan platform yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut. Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas  
Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (reusable).
2. Kecepatan pengembangan  
Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengodean.
3. Kemudahan pemeliharaan  
Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Adanya konsistensi  
Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak  
Karena pendekatan pengembangan lebih dekat dengan dunai nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

#### **2.1.11 Konsel Dasar Berorientasi Objek**

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang

sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dimiliki oleh objek akan dienkapsulasi (dibungkus) menjadi suatu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancangan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu. [24]

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya, dan dapat berinteraksi satu sama lain.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (*Class*)

Kelas adalah sekumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (metode/operasi), hubungan (relationship) dan arti. Suatu kelas dapat diturunkan dari kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara teknis kelas adalah sebuah struktur dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (*object*)

Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata benda, manusia, suatu organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu

menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan. Secara teknis, sebuah kelas saat program dieksekusi akan dibuat sebuah objek. Objek dilihat dari segi teknis adalah elemen pada saat runtime yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehingga sebuah objek hanya ada saat sebuah program dieksekusi. Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat runtime (saat sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.

3. Metode (*method*)

Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau operasi dapat berasal dari event, aktifitas atau aksi keadaan, fungsi, atau kelakuan dunia nyata. Contoh metode atau operasi misalnya *Read*, *Write*, *Move*, *Copy*, dan sebagainya.

4. Atribut (*attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dimiliki secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*abstraction*)

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)



Pembungkusa atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja-nya.

7. Pewarisan (*inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*interface*)

Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain.

9. *Reusability*

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antarobjek

Komunikasi antarobjek dilakukan lewat pesan (message) yang dikirim dari satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

Kemampuan suatu objek digunakan di banyak tujuan yang berbeda dengan nama yang sehingga menghemat baris program.

13. *Package*

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam package yang berbeda.

### 2.1.12 Unified Modeling Language

Analisis dan desain berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Pengertian berorientasi objek berarti bahwa cara mengorganisasikan perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data perilakunya.

Konsep OOAD (*Object Oriented Analysis Design*) mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu:

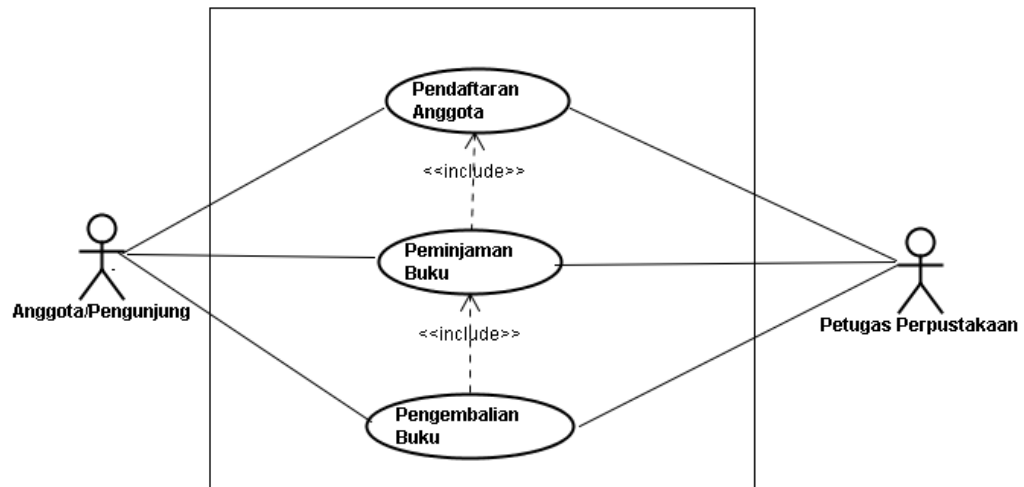
#### 1. Analisis berorientasi objek (OOA)

Metode analisis yang memeriksa requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan. Unified Modeling Language (UML) adalah bahasa untuk mengspesifikasi, memvisualisasi, membangun dan mendokumentasi artefacts (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak), seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. UML merupakan bahasa standar untuk penulisan blueprint software yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Berikut ini merupakan beberapa bagian dari UML adalah sebagai berikut :

#### 1) Use Case Diagram

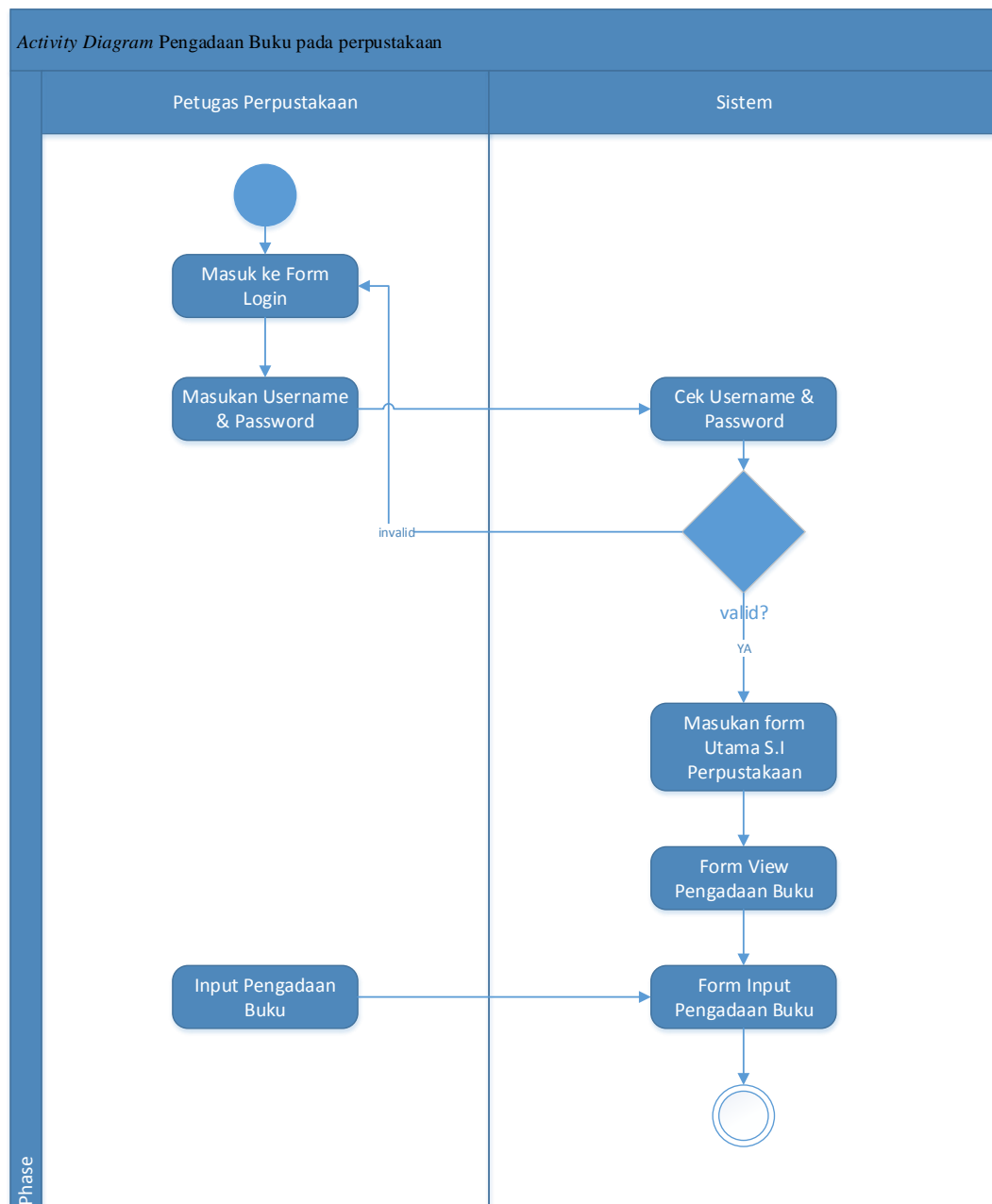
*Use case diagram* menggambarkan sejumlah eksternal aktor dan hubungannya ke use case yang diberikan oleh sistem.



**Gambar 2.11 Contoh Use Case Diagram**

2) *Activity Diagram*

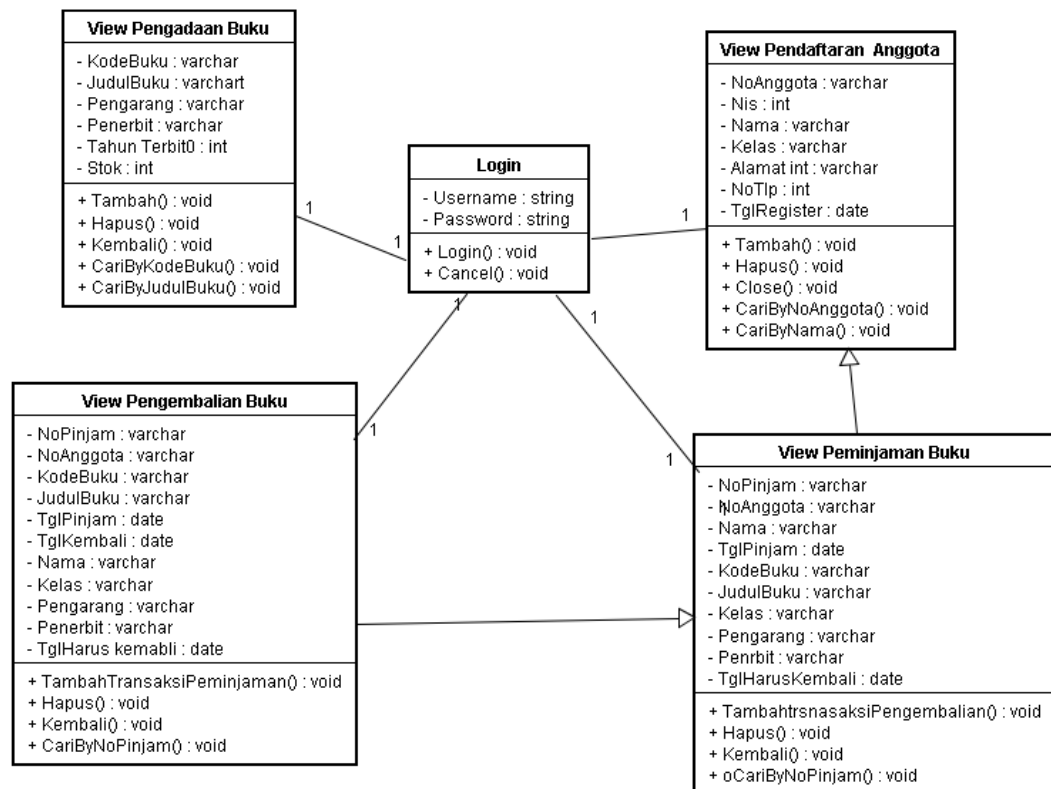
*Activity diagram* menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lain seperti *use case*.



**Gambar 2.12 Contoh Activity Diagram**

3) *Class Diagram*

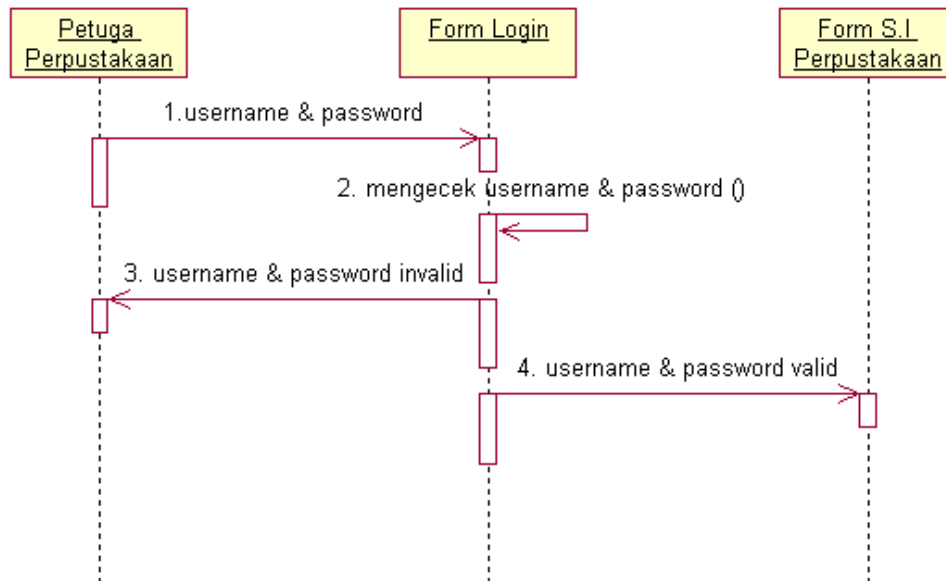
*Class diagram* menggambarkan struktur dan deskripsi class, package (paket) dan objek beserta hubungan satu sama lain seperti containment (penahanan), pewarisan, asosiasi dan lain-lain.



**Gambar 2.13 Contoh Class Diagram**

#### 4) *Sequence Diagram*

*Sequence Diagram* menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.



**Gambar 2.14 Contoh Sequence Diagram**

### 2.1.13 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. [24]

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apa pun bentuknya, entah berupa file teks ataupun Database Management System (DBMS). Kebutuhan basis data dalam sistem informasi meliputi :

1. Masukkan, menyimpan, dan mengambil data
2. Membuat laporan berdasarkan data yang telah disimpan

#### 2.1.14 SQL

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal query yang dimasukkan sama. [24].

#### 2.1.15 Database MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang multithread, dan multi-user. MySQL adalah implementasi dari system manajemen basisdata relasional (RDBMS). MySQL dibuat oleh TcX dan telah dipercaya mengelola system dengan 40 buah database berisi 10.000 tabel dan 500 di antaranya memiliki 7 juta baris. MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL. Pendiri

MySQL AB adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael “Monty”. Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL (General Public License) namun tidak boleh menjadikan produk turunan yang bersifat komersial.

Pada saat ini MySQL merupakan database server yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses database yaitu SQL. SQL (*Structured Query Language*) pertama kali diterapkan pada sebuah proyek riset pada laboratorium riset San Jose, IBM yang bernama system R. Kemudian SQL juga dikembangkan oleh Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses pengaksesan database lebih user-friendly dibandingkan dengan yang lain, misalnya dBase atau Clipper karena mereka masih menggunakan perintah-perintah pemrograman murni. [25]

SQL dapat digunakan secara berdiri sendiri maupun di lekatkan pada bahasa pemrograman seperti C, dan Delphi.

Elemen dari SQL yang paling dasar antara lain pernyataan, nama, tipe data, ekspresi, konstanta dan fungsi bawaan.

Dalam MYSQL ada beberapa operasi dasar yang lebih dikenal dengan operasi CRUD yaitu:

1. *Create*

Create adalah operasi penambahan data baru kedalam table. Terdapat 2 Query untuk menambah data, yang pertama adalah penambahan yang tidak menspesifikasikan nama kolom yang akan ditambahkan dan hanya memberikan isi dari tabelnya. Querynya adalah sebagai berikut:

```
INSERT INTO table_name VALUES (value1,value2,value3,...);
```

Untuk yang kedua adalah menginputkan data dengan mencantumkan nama kolom yang akan diisikan dengan isi di dalamnya. Querynya adalah sebagai berikut:

```
INSERT INTO table_name (column1,column2,column3,...)
VALUES (value1,value2,value3,...);
```

2. *Read*

Read adalah operasi untuk menampilkan semua atau sebagian data yang berada di dalam database. Querynya adalah sebagai berikut:

```
SELECT column_name,column_name
FROM table_name;
```

3. *Update*

Update adalah operasi untuk mengubah data yang ada di dalam database. Querynya adalah sebagai berikut:

```
UPDATE table_name SET column1=value1,column2=value2,...
WHERE some_column=some_value;
```

4. *Delete*

Delete adalah operasi untuk menghapus data yang ada di dalam table. Querynya adalah sebagai berikut:

```
DELETE FROM table_name
WHERE some_column=some_value;
```



Sistem database menyediakan bahasa untuk mendefinisikan database Data Definition Language (DDL) serta bahasa untuk memanipulasi database Data Manipulation Language (DML) untuk melakukan operasi-operasi tertentu pada database dan juga Data Control Language (DCL).

1) Data Definition Language (DDL)

Kumpulan perintah SQL yang digunakan untuk membangun database, bisa juga memiliki arti lain perintah untuk membuat struktur dasar database dan tabel. Perintah-perintah yang termasuk DDL adalah statment connect, create, show, describe, drop, rename, dan alter.

2) Data Manipulation Language (DML)

Kumpulan statment yang digunakan sebagai perintah untuk mengelola data dalam sebuah tabel. Perintah DML digunakan setelah menggunakan DDL anatar lain seperti statment insert, update, delete, dan select.

3) Data Control Language (DCL)

Perintah yang digunakan untuk keperluan keamanan (security) database dengan membuat hak akses tertentu bagi setiap user. Statment pada DCL antara lain grant, revoke, set, dan lock table.

### **2.1.16 PHP**

PHP merupakan akronim dari PHP: Hypertext Preprocessor adalah bahasa pemrograman script berbasis web yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram 30 situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

Untuk menjalankan sistem PHP dibutuhkan tiga komponen:

1. Web server, karena PHP termasuk bahasa pemrograman server side.
2. Program PHP, program yang memproses script PHP.
3. Database server, yang berfungsi untuk mengelola database.

Kelebihan PHP dari bahasa pemrogaraman lain adalah:

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web server yang mendukung PHP dapat ditemukan dimana-mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.

3. Dalam isi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
4. Dalam isi pemahaman, PHP adalah bahasa scripting yang paling mudah karena referensi yang banyak.
5. PHP adalah bahasa open source yang dapat digunakan diberbagai mesin (linux, unix, windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah sistem. [26]

#### **2.1.17 Java Script**

Menurut Sunyoto [2] JavaScript adalah bahasa scripting yang populer di internet dan dapat bekerja di sebagian besar browser populer seperti Internet Explorer (IE), Mozilla FireFox, Netscape, dan Opera. Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag SCRIPT.

Keunggulan javascript yang juga dikenal dengan nama ECMAScript yaitu dapat berjalan di semua platform dengan browser yang mendukung JavaScript, dan hampir semua platform dan browser saat ini mendukung JavaScript. Contoh dari aplikasi yang dibangun dengan JavaScript adalah Google Maps yang dapat berjalan di atas Linux, Windows, dan Mac OS. JavaScript juga semakin dan terus berkembang, seperti pertumbuhan pustaka (library) yang memudahkan untuk menavigasi dokumen, memilih elemen DOM, membuat animasi, menangani event dan mengembangkan aplikasi Ajax. JavaScript adalah bahasa pemrograman client-side yang cross-platform (berjalan di banyak platform) dan bersifat bebas (untuk dimodifikasi dan gratis) juga diadopsi secara universal.

#### **2.1.18 JSON**

Berdasarkan website resminya [www.json.org](http://www.json.org), JSON merupakan bagian dari bahasa pemrograman JavaScript (Standard ECMA-262 3rd Edition – December 1999). JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga-C, termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan sebagainya. Kelebihan inilah yang membuat JSON menjadi sebuah bahasa data-interchange yang ideal.

JSON dibangun dalam dua struktur:

- 1 Beberapa pasangan dari nama/nilai. Dalam beberapa bahasa pemrograman biasa disebut dengan istilah object, record, struct, tabel hash, key list atau associative array.
- 2 Nilai-nilai yang terusun secara ordered list. Biasa disebut dengan array, vector, list atau daftar dalam bahasa pemrograman.

JSON adalah struktur data yang universal, dalam artian bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON.

#### **2.1.19 Jaringan Internet**

Jaringan Internet adalah gabungan jaringan-jaringan komputer di seluruh dunia dan merupakan rangkaian komputer terbesar di dunia yang memungkinkan komunikasi data antar komputer yang terhubung ke jaringan tersebut.

Internet awalnya merupakan suatu rencana dari Departemen Pertahanan Amerika Serikat (US Department of Defense) pada sekitar tahun 1960. Dimulai dari suatu proyek yang dinamakan ARPANET atau Advanced Research Project Agency Network. Beberapa universitas di Amerika Serikat diantaranya UCLA, Stanford, UC Santa Barbara dan University of Utah, diminta bantuan dalam mengerjakan proyek ini dan awalnya telah berhasil menghubungkan empat computer di lokasi universitas berbeda tersebut.

Tahun 1985, Internet Modern memperoleh dukungan ketika National Science foundation membentuk NSFNET, menghubungkan lima pusat superkomputer di Princeton University, Pittsburgh, University of California di San Diego, University of Illinois di Urbana-Champaign, dan Universitas Cornell., beberapa jaringan regional segera dikembangkan; Akhirnya, pemerintah menindahkan bagian-bagian ARPANET untuk NSFNET.

### **2.1.20 Definisi IP Address**

IP Address merupakan deretan bilangan biner diantara 32 bit hingga 128 bit yang dipakai sebagai media untuk mengidentifikasi untuk setiap perangkat komputer yang terhubung pada jaringan komputer (intranet / internet). Bilangan biner 32 bit dipakai untuk setiap IP Address versi IPv4, sedangkan bilangan biner 128 bit digunakan untuk setiap versi IP Address IPv6.

IP Address akan berguna sebagai data identifikasi setiap device (komputer dan perangkat lainnya) yang terhubung ke jaringan komputer yang memanfaatkan internet protocol sebagai media penghubungnya. Adapun fungsi dari IP address yaitu :

1. **Alat Identifikasi Host Atau Antar Muka Pada Jaringan Komputer.**  
Fungsi IP Address yang pertama adalah sebagai alat identifikasi host ataupun antar muka jaringan komputer. Jika diilustrasikan seperti kehidupan nyata, maka IP Address berfungsi sebagai nama ataupun identitas seseorang. Dalam hal ini, seperti halnya nama, setiap komputer memiliki IP Address yang unik dan berbeda antara satu dengan yang lainnya (yang terkoneksi pada satu jaringan komputer).
2. **Alamat Lokasi Jaringan.**  
Fungsi IP Address yang kedua adalah sebagai penunjuk alamat lokasi jaringan. Jika kita ilustasikan kembali dalam kehidupan nyata, maka IP address dapat diilustrasikan sebagai penunjukkan alamat rumah tempat tinggal seseorang. IP Address akan menunjukkan lokasi keberadaan sebuah komputer, berasal dari daerah mana, ataupun negara mana. Dalam hal ini, seperti halnya dalam kehidupan nyata, ada rute / jalan yang harus ditempuh agar data yang diinginkan bisa sampai ke komputer yang ingin dituju.

#### **2.1.20.1 Kelas Pada IP Address**

IP Address dikelompokkan dalam lima kelas, yaitu kelas A, B, C, D, dan E. Perbedaannya terletak pada ukuran dan jumlah. Adapun kelas IP address yaitu sebagai berikut :

1. Kelas A

IP Address kelas A digunakan untuk sedikit jaringan dengan jumlah host yang sangat banyak. IP Address kelas ini biasanya digunakan untuk jaringan-jaringan komputer yang tidak terlalu padat lalu lintas trafictnya.

2. Kelas B

IP Address kelas B digunakan pada jaringan yang berukuran sedikit lebih besar / sedang dari IP Address kelas A. Network IP kelas B biasanya mampu menampung sekitar 65.000 an host.

3. Kelas C

IP Address kelas C memiliki kemampuan yang paling besar dibandingkan dengan dua kelas yang sebelumnya. IP Address kelas ini mampu dibentuk oleh lebih dari 2 juta network.

4. Kelas D

IP kelas D digunakan untuk multicasting, yaitu penggunaan aplikasi secara bersama-sama oleh beberapa komputer, dan IP yang bisa digunakan adalah 224.0.0.0 – 239.255.255.255.

5. Kelas E

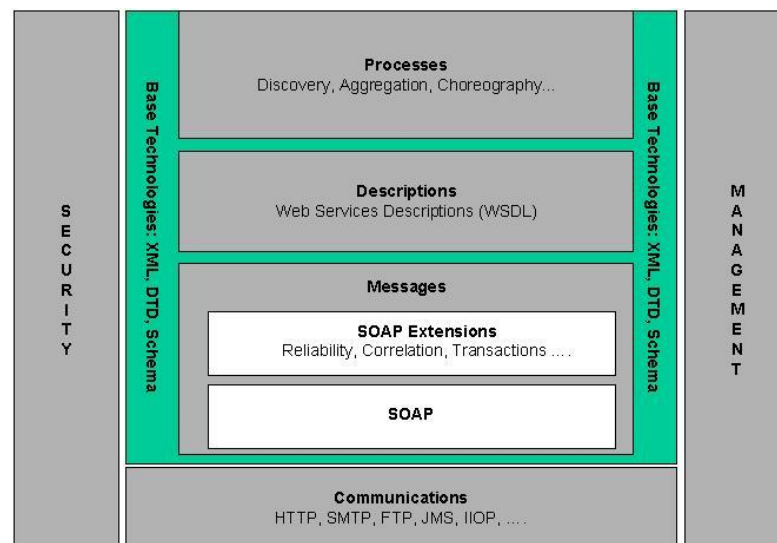
Memiliki range dari 240.0.0.0 – 254.255.255.255, IP ini digunakan untuk eksperimen yang dipersiapkan untuk penggunaan IP address di masa yang akan datang.

### **2.1.21 Web Service**

Web service adalah sebuah entitas komputasi yang dapat diakses melalui jaringan internet maupun intranet dengan standar protokol tertentu dalam platform dan antarmuka bahasa pemograman yang independen. Tujuan pengembangannya adalah untuk menjembatani komunikasi antar program, sehingga aplikasi yang satu dan aplikasi yang lain yang terdapat pada suatu jaringan yang sama atau pada jaringan berbeda dapat saling berkomunikasi asalkan menggunakan standart protokol yang ditetapkan oleh web service. Hal ini bisa terjadi, karena standart protokol itu tidaklah terikat pada suatu platform atau bahasa pemograman. Protokol itu sendiri dibangun oleh Extensible Markup Language

(XML) yang memang kenyataannya telah didukung oleh banyak platform, bahasa pemrograman, dan oleh developer di seluruh dunia.

Arsitektur web service dibangun oleh beberapa layer dan teknologi yang saling berhubungan. Banyak cara untuk memvisualisasikan service, sama banyaknya dengan cara untuk membangun dan menggunakan web service gambar berikut hanyalah salah satu cara untuk menggambarkan arsitekturnya.



**Gambar 2.15 Arsitektur Web Service**

Web service disusun oleh tiga komponen standart, yaitu :

1. Simple Object Access Protocol (SOAP), yaitu protokol yang bertanggungjawab dalam pertukaran informasi dalam lingkungan jaringan terdistribusi.
2. Web service definition Language(WDSL), dokumen standart yang dituliskan pada format XML, dan mendefinisikan kehadiran web service dalam suatu jaringan.
3. Universal description, Discovery and Inttegration(UDDI), yaitu suatu lokasi direktori yang beriskan service (layanan) dan bersifat bebas platform (platform independent), dituliskan berbasis XML dan dapat diakses oleh entittas yang berada di dalam dan luar jaringan.

Adanya standart tersebut membuat web service mudah di akses melalui berbagai antarmuka dan juga memberi peluang dimungkinkannya berbagai sistem

yang dibangun pada platform yang berbeda dan bahasa yang berbeda untuk berkolaborasi pada dalam suatu pekerjaan. [27]

### 2.1.22 *Google Cloud Messaging(GCM)*

GCM (*Google Cloud Messaging*) adalah sebuah layanan pada *Android* yang berfungsi untuk menjalankan sebuah *Push Notification*. Di *Android* layanan GCM dijalankan oleh sebuah aplikasi yang berjalan di latar belakang dan bernama Layanan Google Play. Layanan ini akan membangunkan aplikasi yang bersangkutan saat menerima notifikasi baru dan akan menampilkan notifikasi tersebut pada panel notifikasi *Android*.

Aplikasi *Android* yang menggunakan GCM tidak perlu berjalan di belakang layar untuk menerima sebuah notifikasi. GCM (dalam hal ini Layanan Google Play) akan membangunkan aplikasi tersebut dengan mekanisme Broadcast Intent (Event yang terjadi saat ada notifikasi baru) ketika ada pesan tiba selama aplikasi tersebut diatur untuk diizinkan menerima Pesan Siaran. Namun tidak sedikit pula aplikasi berjalan di latar belakang yang menggunakan GCM Service.

GCM ini “100 persen” menggunakan server Google dan *Android* akan memeriksa ke server tersebut setiap beberapa menit (tergantung pengembang rom). Setiap pesan tersebut mempunyai batasan 1024 byte atau hanya 1 kb dan Google membatasi jumlah pesan secara keseluruhan dan jumlah pesan ke perangkat tertentu. [22]

Pada system ini GCM digunakan untuk memberikan notification ke pengguna pada saat mikrokontroler mendeteksi pergerakan.