BAB 2
LANDASAN TEORI

2.1 Profil Perusahaan

PT. Dirgantara Indonesia(Persero) adalah salah satu perusahaan di Asia yang bergerak dalam dunia penerbangan. PT. Dirgantara Indonesia(Persero) mempunyai kompetensi utama dalam bidang pengembangan dan perancangan pesawat, pembuatan struktur pesawat, perakitan, serta perawatan untuk pesawat sipil dan militer ukuran kecil sampai menengah. Sejak awal berdirinya di Bandung pada tahun 1976, PT. Dirgantara Indonesia(Persero) telah berhasil menjadi industri penerbangan. Dalam dunia penerbangan, PT. Dirgantara Indonesia(Persero) telah memproduksi berbagai macam pesawat terbang. Salah satunya pesawat CN235 yang digunakan sebagai alat transportasi untuk sipil dan militer, patroli maritim dan penjaga pantai.

Dengan persetujuan kerjasama antara PT. Dirgantara Indonesia(Persero) dan Airbus Defence and Space, PT. Dirgantara Indonesia(Persero) memproduksi pesawat NC212i (pengembangan pesawat C212-400), juga sebagai pusat pengiriman dan perakitan akhir serta penyedia komponen pesawat CN295.

Disamping pesawat dengan jenis *fixed wing*, PT. Dirgantara Indonesia(Persero) juga memproduksi beberapa tipe pesawat jenis *rotary wing* atau helikopter seperti NAS-332, AS 725-Cougar, AS365N3-Dauphin dibawah lisensi Airbus Helikopter dan BELL-412 EP dibawah lisensi Bell Textron. Selain itu, PT. Dirgantara Indonesia(Persero) juga memproduksi komponen, *tools* dan bagian dari pesawat – pesawat Airbus tipe A320/321/330/340/350/380, MK2 dan EC725 milik Airbus Helicopters, serta pesawat CN235 dan C295 milik Airbus Defence and Space. [11]

2.1.1 Sejarah Perusahaan

Aktivitas kedirgantaraan di Indonesia dimulai tahun 1946 dengan dibentuknya Biro Rencana dan Konstruksi Pesawat di lingkungan Tentara Republik Indonesia Angkatan Udara di Madiun, yang kemudian dipusatkan di

Andir, Bandung. Tahun 1953, kegiatan tersebut mendapat wadah baru dengan nama Seksi Percobaan yang pada tahun 1957 berubah menjadi Sub Depot Penyelidikan, Percobaan dan Pembuatan Pesawat Terbang. Tahun 1960, Sub Depot ini ditingkatkan menjadi Lembaga Persiapan Industri Penerbangan (LAPIP) yang kemudian berubah menjadi Komando Pelaksanaan Industri Pesawat Terbang (KOPELAPIP) yang pada tahun 1966 digabung dengan PN Industri Pesawat Terbang Berdikari menjadi Lembaga Industri Penerbangan Nurtanio (LIPNUR).

Pada tahun 1975, PT. Pertamina membentuk Divisi Advanced Technology dan Teknologi Penerbangan (ATTP) yang bertujuan menyiapkan infrastruktur bagi industri kedirgantaraan di Indonesia. Berdasarkan Akta Notaris No. 15, tanggal 24 April 1976, didirikan PT. Industri Pesawat Terbang Nurtanio, dipimpin oleh Prof. Dr. Ing. B.J.Habibie. Perusahaan ini merupakan penggabungan antara LIPNUR dan ATTP. Kemudian pada bulan April 1986, melalui Keputusan Presiden (KEPRES) N0. 15/1986 dan Rapat Umum Pemegang Saham Perusahaan, nama perusahaan diganti menjadi PT. Industri Pesawat Terbang Nusantara (IPTN) dan tanggal 24 Agustus 2000, nama perusahaan secara resmi diubah oleh Presiden Republik Indonesia saat itu menjadi PT. Dirgantara Indonesia (PT. DI).

Pada tahun 1998, berdasarkan Peraturan Pemerintah Nomor 35 Tahun 1998 tentang Penyertaan Modal Negara Republik Indonesia Untuk Pendirian Perusahaan Perseroan (Persero) Di Bidang Industri, saham negara pada PT. IPTN (Persero) dialihkan menjadi penyertaan pada PT. Bahana Pakarya Industri Strategis (Persero) (PT. BPIS), dengan demikian status PT. IPTN berubah menjadi anak perusahaan PT. BPIS.

Kemudian pada tahun 2002, berdasarkan Peraturan Pemerintah Nomor 52 Tahun 2002 tentang Penyertaan Modal Negara Republik Indonesia Ke Dalam Modal Saham PT. Dirgantara Indonesia, PT. Industri Kereta Api, PT. Industri Telekomunikasi Indonesia Dan PT. LEN Industri Dan Pembubaran Perusahaan Perseroan (Persero) PT. Bahana Pakarya Industri Strategis, PT. Dirgantara Indonesia berubah menjadi badan hukum persero.

2.1.2 Visi dan Misi Perusahaan

Visi PT. Dirgantara Indonesia(Persero) adalah menjadi perusahaan kelas dunia dalam industri berbasis pada penguasaan teknologi tinggi dan mampu bersaing dalam pasar global dengan mengandalkan keunggulan biaya.

Misi PT. Dirgantara Indonesia(Persero) adalah sebagai pusat keunggulan di bidang industri dirgantara terutama dalam rekayasa, rancang bangun, manufaktur, produksi dan pemeliharaan untuk kepentingan komersial dan militer dan juga aplikasi di luar industri dirgantara. Menjalankan usaha dengan selalu berorientasi pada aspek bisnis dan komersial dan dapat menghasilkan produk jasa yang memiliki keunggulan biaya.

2.1.3 Logo Perusahaan

Adapun logo perusahaan PT. Dirgantara Indonesia (Persero) ditunjukan pada Gambar 1.1 Logo Perusahaan.



Sumber Gambar : http://www.indonesian-aerospace.com/kiosk-web/gallery.html

Gambar 1.1 Logo Perusahaan

2.2 Monitoring

Monitoring adalah proses pengumpulan dan analisis informasi berdasarkan indikator yang ditetapkan secara sistematis dan kontinu tentang kegiatan/program sehingga dapat dilakukan tindakan koreksi untuk penyempurnaan program/kegiatan itu selanjutnya. *Monitoring* adalah pemantauan yang dapat dijelaskan sebagai kesadaran(*awareness*) tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan akan dapat membuat pengukuran melalui waktu yang menunjukan pergerakan kearah tujuan atau menjauh dari itu. *Monitoring* akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap

proses berikut objek. Proses *Monitoring* adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program. Memantau perubahan yang fokus pada proses dan keluaran. [12]

2.3 Polusi Udara

Polusi udara atau Pencemaran udara adalah masuknya atau dimasukkannya zat, energi, dan atau komponen lain ke dalam udara oleh kegiatan manusia, sehingga melampaui baku mutu udara yang telah ditetapkan. [13] Sumber pencemaran udara dapat dibagi menjadi tiga kategori besar, yaitu sumber perkotaan dan industri, sumber pedesaan/pertanian dan lainnya, emisi alami. Umumnya sumber pencemar udara utama dari industri merupakan sumber titik, walaupun di dalam kawasan industri besar akan ditemui pula sumber garis, misalnya jalan penghubung di dalam kawasan tersebut, maupun sumber area. Sumber titik di industri juga dapat diamati sebagai :

1. Emisi normal, yaitu emisi yang berasal dari sumber – sumber yang terkontrol dan disalurkan melalui cerobong sehingga dapat diukur atau dipantau besarannya,
2. Emisi abnormal, yaitu emisi yang berasal dari sumber – sumber titik kecil. Sumber ini lebih sulit dikontrol dan diukur,
3. Emisi sementara/aksidental yang berasal dari kebocoran dan tumpahan kecil, ledakan dan kebakaran.

Emisi normal dan abnormal masih dapat dikelola walaupun pada emisi abnormal tindakannya lebih sulit, sedangkan emisi aksidental diatasi dengan tindakan tanggap darurat. Emisi abnormal berasal dari sumber kecil sebuah proses berupa kebocoran gas atau uap dari sambungan pipa. Emisi ini sulit untuk diperhitungkan satu persatu tetapi dapat menjadi beban emisi yang cukup besar di dalam suatu kompleks industri. Emisi dapat diestimasi berdasarkan pengukuran pada titik – titik yang merepresentasikan jenis, sumber dan pelaporan yang dilakukan dengan cara digabungkan sebagai sumber area.

2.4 IoT (*Internet of Things*)

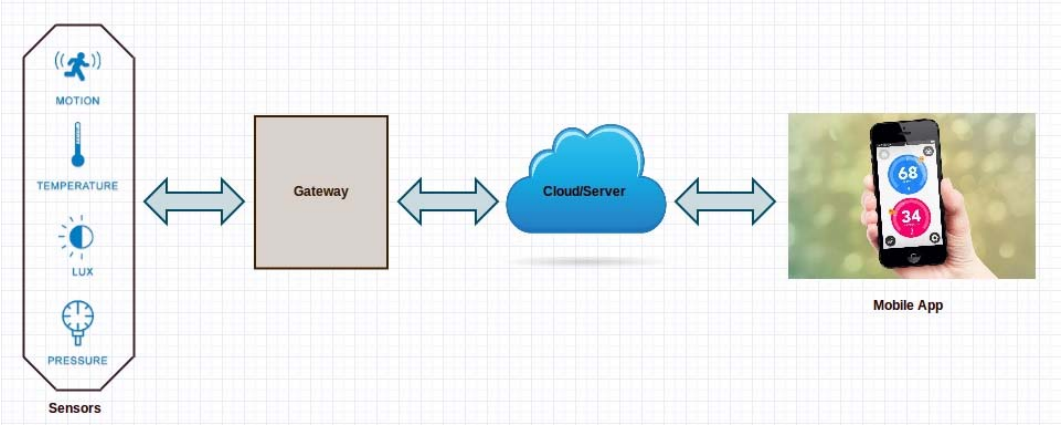
Internet of Things(IoT) pertama kali dikenalkan oleh Kevin Ashton pada tahun 1999. Meski telah diperkenalkan sejak 15 tahun yang lalu, hingga kini belum ada sebuah konsensus global mengenai definisi IoT. Namun secara umum konsep IoT diartikan sebagai sebuah kemampuan untuk menghubungkan objek – objek cerdas dan memungkinkannya untuk berinteraksi dengan objek lain, lingkungan maupun dengan peralatan komputasi cerdas lainnya melalui jaringan internet. [14] IoT dalam berbagai bentuknya telah mulai diaplikasikan pada banyak aspek kehidupan manusia. CISCO bahkan telah menargetkan bahwa pada tahun 2020, akan ada 50 miliar objek yang terhubung dengan internet. [15]

Meluasnya adopsi berbagai teknologi IoT, membuat kehidupan manusia menjadi lebih nyaman. Dari sisi pengguna perorangan, IoT sangat terasa pengaruhyna dalam bidang domestik seperti pada aplikasi rumah dan mobil cerdas. Dari sisi pengguna bisnis, IoT sangat berpengaruh dalam meningkatkan jumlah produksi serta kualitas produksi, mengawasi distribusi barang, mencegah pemalsuan, mempersingkat waktu ketidaktersediaan barang pada pasar retail dan supply chain management.

IoT yang dipakai pada aplikasi *smart city* seperti monitor kualitas udara dengan perangkat yang dipasang di area yang membutuhkan pengawasan, akan memudahkan setiap pengguna untuk menerima data kualitas udara dari area tersebut. Dengan demikian, pengawasan kualitas udara akan lebih mudah dilakukan. [16]

2.4.1 Arsitektur IoT

Untuk berjalannya sebuah sistem IoT, maka diperlukan beberapa komponen pembangun, seperti yang ditunjukan pada Gambar 1.2 Blok Sistem IoT.



Sumber Gambar : https://www.embitel.com/blog/embedded-blog/how-iot-works-an-overview-of-the-technology-architecture-2

Gambar 1.2 Blok Sistem IoT

Meski telah mulai diaplikasikan pada banyak bidang kehidupan sehari – hari, namun belum ada satu definisi yang baku dari IoT. Secara sederhana konsep IoT dapat digambarkan dengan bentuk arsitektur seperti ditunjukan pada Gambar 1.2 Blok Sistem IoT. Pada tingkat pertama adalah perangkat keras yang dapat mengenali dirinya dan mengindera lingkungannya. Seperti membaca lokasi, kondisi cuaca, gerakan mesin atau kondisi kesehatan. Perangkat yang digunakan pada lapisan ini adalah RFID, sensor, kontrol dan aktuator.

Pada lapisan atau tingkat kedua adalah gateway yang merupakan jembatan penghubung antara jaringan internal sensor yang mengumpulkan data, dengan jaringan luar internet melalui berbagai media komunikasi nirkabel seperti WiFi, bluetooth, selular satelit, Zigbee. *Gateway* juga merupakan tempat pengolah data tahap pertama, pengalamatan dan pengaturan *routing*. Data yang ditransmisikan melalui *gatewar* kemudian disimpan, diolah dan kemudian digunakan untuk melakukan hal – hal cerdas sesusai tujuan IoT.

Pada sisi pengguna, layanan IoT dimanfaatkan melalui aplikasi bergerak pada perangkat cerdas. Aplikasi bergerak yang intuitif ini yang membantu pengguna untuk mengatur dan memonitor perangkatnya dari jarak jauh. [14]

2.5 API (*Application Programming Interface*)

API (*Application Programming Interface*) adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal.

Pada konteks web, API merupakan pemanggilan fungsi lewat *Hyper Text Transfer Protocol*(HTTP) dan mendapatkan respon berupa *Extensible Markup Language*(XML) atau *Javascript Object Notation*(JSON). Pemanggilan fungsi ke suatu situs tertentu akan menghasilkan respon yang berbeda. [17]

Format pemanggilan fungsi ini diatur pada masing – masing penyedia API. Beberapa contoh situs yang menyediakan API yaitu situs Facebook, Twitter, Picasa, Flickr dan Photobucket. Berikut ini beberapa penjelasan dari API yang telah ada.

2.5.1 *Facebook API*

Graph API merupakan inti dari Facebook platform yang memungkinkan pengembang untuk membaca dan menulis data pada Facebook. Graph API merupakan perkembangan dari *Old Rest* API. Paradigma membaca dan menulis data dari yang berorientasi pada fungsi berubah menjadi berorientasi objek. Objek Facebook contohnya seperti Users, Photos, Albums, dan koneksi diantaranya. Pendekatan ini membuat Facebook API lebih sederhanda dan konsisten saat bekerja dengan objek. [18]

Contoh Syntax :

```
FB.login(function(response) {
  // handle the response
}, {scope: 'public_profile,email'});
```

Gambar 1.3 Facebook API Syntax

2.5.2 *Flickr API*

Flickr API terdiri dari fungsi atau method yang dapat dipanggil. Pemanggilan dilakukan dengan mengirim permintaan pada fungsi tertentu dengan beberapa argumen. Pemanggilan fungsi akan menghasilkan respon dengan format

tertentu. Semua pemanggilan fungsi memiliki beberapa parameter. Daftar penjelasan argumen dan respon dengan format tertentu. Daftar penjelasan argumen dan respon tiap fungsi dapat dilihat pada daftar fungsi di http://www.flickr.com/services/api. [19]

Contoh Syntax :

```
https://api.flickr.com/services/rest/?method=f
lickr.galleries.getPhotos&api_key=APIKEY&galle
ry_id=66911286-
72157647277042064&format=json&nojsoncallback=1
```

Gambar 1.4 Flickr API Syntax

2.5.3 Picasa *API*

Pengaksesan API Picasa yaitu dengan menggunakan Google API. Google API menyediakan penggunaan OAuth 2.0 untuk mengakses data – data terlindungi yang tersimpan di dalam Google API. Alur proses pengaksesan data berbeda untuk setiap aplikasi *client*. Dalam alur proses ini aplikasi *client* mengirim permintaan untuk mendapatkan akses token yang nantinya digunakan untuk berhubungan dengan pemilik data terlindungi yang akan diakses. Akses token memiliki batasan pengaksesan berdasarkan ruang lingkup yang telah didefinisikan sebelumnya. *Library* untuk google OAuth 2.0 yaitu google-oauth-java-client yang dapat diunduh di http://code.google.com/p/google-oauth-javaclient/wiki/OAuth2. [17]

```
GET
https://picasaweb.google.com/data/feed/api/user/user
ID
```

Gambar 1.5 Picasa API Syntax

2.6 Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer yang dibangun pada sebuah keping (*chip*) tunggal. Jadi, hanya dengan sebuah keping IC saja dapat dibuat sebuah sistem komputer yang dapat dipergunakan untuk mengontrol alat.

Saat ini sebagian besar peralatan elektronika dikontrol dengan mikrokontroler, misalnya mesin fax, mesin foto – copy, mesin cuci otomatis, sampai handphone. Peralatan tersebut tidak akan dapat dibuat dengan ukuran yang cukup kecil jika tidak menggunakan kontrol dengan menggunakan mikrokontroler.

2.6.1 Komponen Penyusun Mikrokontroler

Mikrokontroler disusun oleh beberapa komponen, yaitu CPU (*Central Processing Unit*), ROM (*Read Only Memory*), RAM(*Random Access Memory*), dan I/O(*Input/Output*). Keempat komponen ini secara bersama – sama membentuk sistem komputer dasar. Beberapa mikrokontroler memiliki tambahan komponen lain, misalnya ADC (*Analog to Digital Converter*), Timer/Counter. [20]

2.6.2 CPU (*Central Processing Unit*)

CPU terdiri atas dua bagian, yaitu unit pengendali (*control unit*) serta unit aritmatika dan logika (ALU). Fungsi utama unit pengendali adalah mengambil, mengkodekan, dan melaksanakan urutan instruksi sebuah program yang tersimpan dalam memori.

Unit pengendali menghasilkan dan mengatur sinyal pengendali yang diperlukan untuk menyerempakkan operasi, aliran, dan instruksi program. Unit aritmatika dan logika berfungsi untuk melakukan proses perhitungan yang diperlukan selama program dijalankan serta mempertimbangkan suatu kondisi dan mengambil keputusan yang diperlukan untuk instruksi – instruksi berikutnya.

2.6.3 Bus Alamat

Bus alamat berfungsi sebagai sejumlah lintasan saluran pengalamatan antara alat dengan komputer. Pengalamatan ini harus ditentukan terlebih dahulu untuk menghindari terjadinya kesalahan pengiriman sebuah instruksi dan terjadinya bentrok antara dua buah alat yang bekerja secara bersamaan.

2.6.4 Bus Data

Bus data merupakan sejumlah lintasan saluran keluar – masuknya data dalam suatu mikrokontroler. Pada umumnya saluran data yang masuk sama dengan saluran data yang keluar.

2.6.5 Bus Kontrol

Bus kontrol atau bus pengendali ini berfungsi untuk menyerempakkan operasi mikrokontroler dengan operasi rangkaian luar.

2.6.6 Memori

Di dalam sebuah mikrokontroler terdapat suatu memori yang berfungsi untuk menyimpan data atau program. Ada beberapa jenis memori, di antaranya adalah RAM dan ROM. Ada beberapa tingkatan memori, di antaranya adalah register internal, memori utama dan memori massal. Register internal adalah memori di dalam ALU. Waktu akses register sangat cepat, umumnya kurang dari 100ns.

Memori utama adalah memori yang ada pada suatu sistem. Waktu aksesnya lebih lambat dibandingkan register internal, yaitu antara 200 sampai 1.000ns. Memori masal dipakai untuk penyipanan berkapasitas tinggi, biasanya berbentuk disket, pita magnetik, atau kaset. [21]

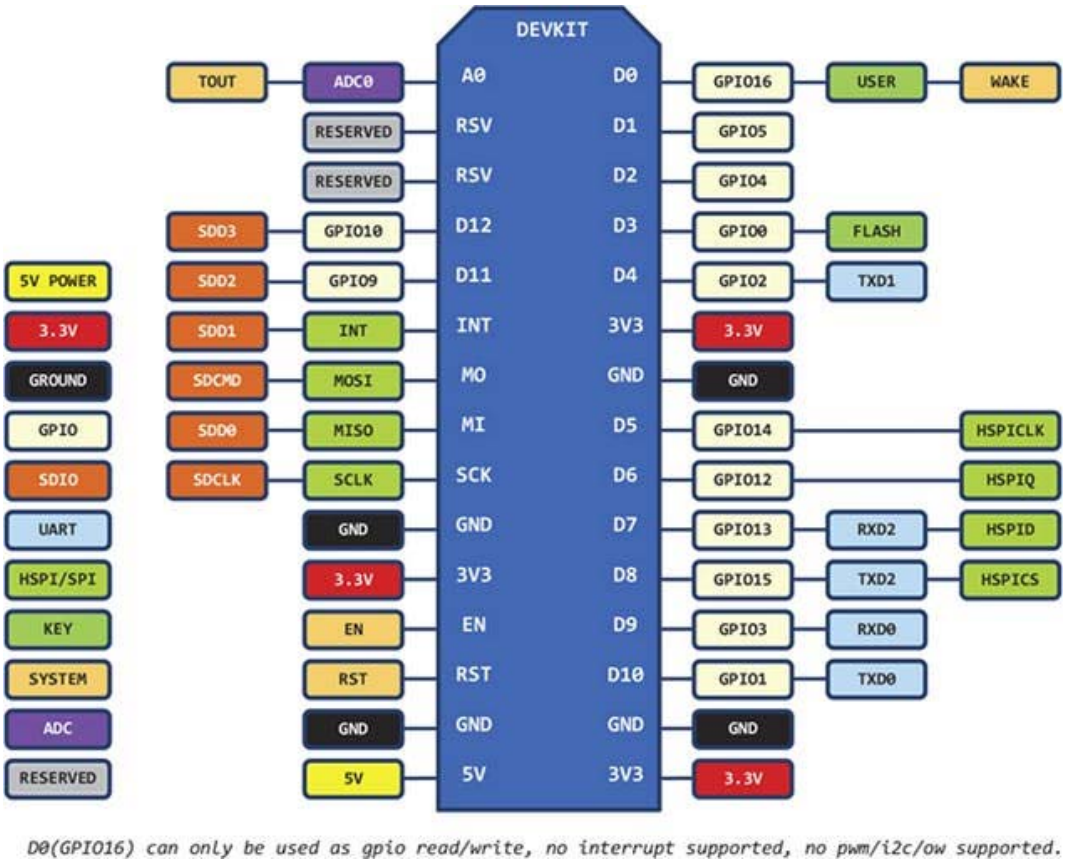
2.7 NodeMCU ESP8266

NodeMCU adalah sebuah perangkat IoT yang bersifat open – source. Didalamnya telah terdapat firmware yang berjalan diatas ESP8266 Wi-Fi SoC dari Sistem Espress if, dan perangkat keras yang berbasis modul ESP – 12. Nama NodeMCU sendiri mengacu pada firmware yang ada didalamnya. *Firmware* perangkat ini menggunakan bahasa pemrograman Lua. Perangkat ini berbasis pada proyek eLua, dan dibangun diatas Espress if Non – OS SDK untuk ESP8266. ESP8266 adalah Wi – Fi SoC yang telah diintegrasikan dengan inti Tensilica Xtensa LX106, yang banyak digunakan untuk aplikasi – aplikasi IoT. Proyek NodeMCU dimulai sejak tanggal 13 Oktober 2014, ketika Hong melakukan *commit* untuk file nodemcu-firmware pertama ke GitHub. Dua bulan kemudian, proyek ini

diperluas dengan menambahkan sebuah platform open – hardware yang dinamakan ESP8266, dan setelahnya diberi nama devkit v.0.9. Setelah itu, dilakukan porting library klien MQTT dari Contiki ke dalam *platform* ESP8266 SoC, yang kemudian ditetapkan untuk proyek NodeMCU. Dengan itu, NodeMCU mempunyai dukungan untuk protokol MQTT IoT, menggunakan Lua untuk mengakses MQTT broker. Perkembangan lainnya yang tidak kalah penting dilakukan pada tanggal 30 Januari 2015, ketika Devsaurus melakukan porting u8glib untuk proyek NodeMCU, memberikan kemudahan pada NodeMCU untuk menggunakan sebuah LCD, OLED, monitor. [8]

2.7.1 Konfigurasi PIN

Konfigurasi PIN dari perangkat NodeMCU dapat dilihat pada



Sumber Gambar : https://www.hackster.io/Aritro/getting-started-with-esp-nodemcu-using-arduinoide-aa7267

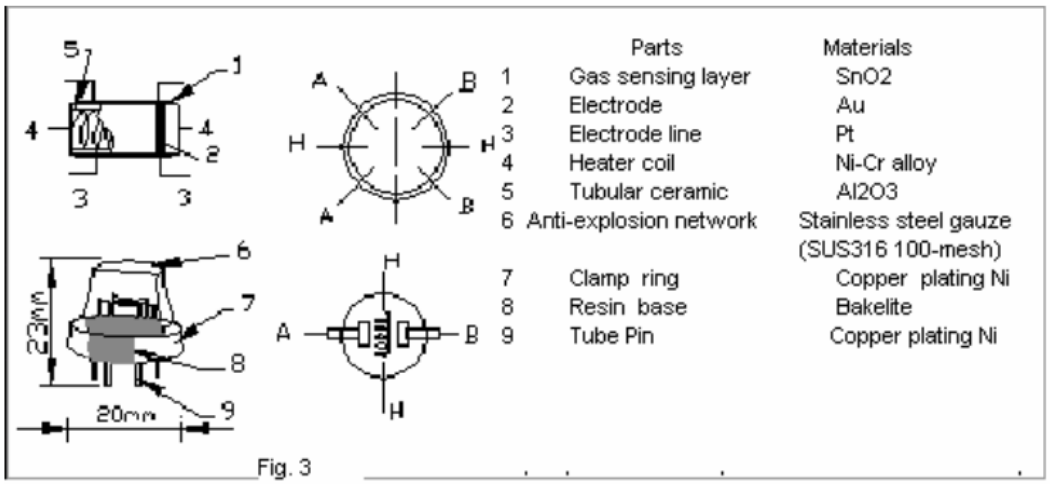
Gambar 1.6 NodeMCU *pinout*

2.7.1.1 GPIO (General Purpose Input/Output)

Merupakan sebuah *interface* yang banyak terdapat pada perangkat – perangkat mikrokontroler saat ini. Secara umum, dalam sebuah mikrokontroler ada lebih dari satu GPIO. Pin ini dapat diprogram sebagai input, dimana data dari sumber luar dimasukan kedalam sistem untuk dapat dimanipulasi. Pin GPIO juga dapat diprogram sebagai keluaran atau *output* dari mikrokontroler.

2.8 MQ – 135

Tujuan utama dari sensor MQ – 135 adalah untuk mengukur kualitas udara untuk di dalam dan di luar ruangan. Khususnya mendeteksi nilai ppm (*particles per million*) untuk kandungan NH_3, NO_x, alkohol, bensin, asap, CO_2. Sensor gas MQ-135 memiliki kepekaan yang relatif tinggi, low cost dan relatif lebih tahan lama untuk digunakan sebagai alternatif dalam pengukuran polusi udara. [22]



Sumber Gambar : Technical Data MQ – 135 Gas Sensor

Gambar 1.7 Struktur dan Konfigurasi sensor MQ-135

2.9 Smartphone

Menurut Gary B, Thomas J dan Misty E,(2007:139), *Smartphone* adalah sebuah telepon genggam yang memiliki konektivitas internet dan menyediakan fungsi Personal Digital Assistant(PDA), seperti fungsi kalender, buku agenda, buku alamat, kalkulator, dan catatan. Menurut David Wood, Wakil Presiden Eksekutif PT Symbian OS, “Ponsel dan *Smartphone* dapat dibedakan dengan dua cara fundamental : bagaimana mereka dibuat dan apa yang mereka bisa lakukan.” [23]

Schmidt(2009) juga mengemukakan bahwa istilah *smartphone* merupakan istilah yang digunakan untuk mendeskripsikan perangkat *mobile* yang menggabungkan fungsi telepon genggam, PDA, audio player, digital camera, camcoder, penerima GPS(*Global Positioning System*), dan *Personal Computer*(PC).

2.10 Android

Android adalah sebuah sistem operasi berbasis kernel linux yang dikembangkan oleh Google dan digunakan untuk perangkat - perangkat *mobile*. Android memungkinkan pengembang untuk membuat koding dengan menggunakan bahasa Java, juga dapat mengontrol perangkat melalui library Java yang telah dikembangkan oleh Google.



Sumber Gambar : https://developer.android.com/distribute/marketing-tools/brand-guidelines.html

Gambar 1.8 Logo Android

Sistem operasi Android bersiftat Open Source. Android adalah sebuah subset perangkat lunak yang dapat diunduh secara bebas dan didalamnya telah terdapat sistem operasi, *middleware* dan *key* aplikasi berbasis Linux dan Java. Google telah membeli pengembang android pada tahun 2005, dan Android dikenalkan pada tahun 2007. Google telah merilis kode android secara open-source dibawah lisensi Apache. Sampai saat ini sudah banyak sekali pengembang aplikasi Android diseluruh dunia. Langkah pertama dalam pengembangan aplikasi Android,

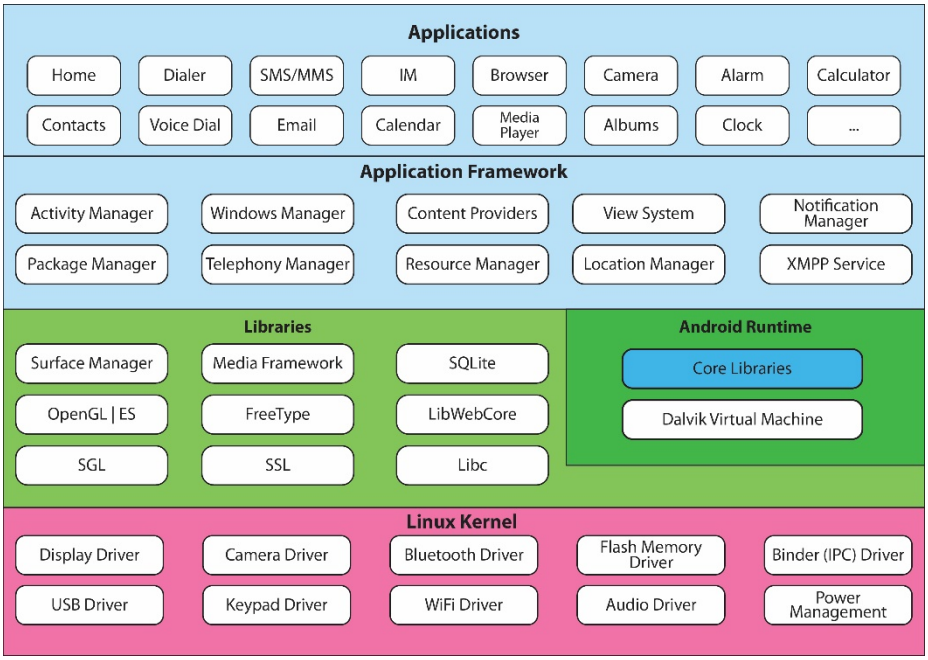
pengembang membuat aplikasi menggunakan bahasa Java, kemudian aplikasi – aplikasi tersebut diunduh melalui situs pihak ketiga atau toko aplikasi online. Pada bulan Februari 2012, sebanyak 450.000 aplikasi telah tersedia untuk Android. Tetapi perkiraan jumlah aplikasi yang telah diunduh sejak bulan Desember 2011 telah lebih dari 10 miliar. Ada lebih dari 300 juta Android digunakan dan lebih dari 850.000 perangkat aktif setiap hari. Android adalah sistem operasi yang paling banyak digunakan dengan saham sebesar 48% dan lebih dari 400.000 aplikasi tersedia di Google Play Store. Aplikasi – aplikasi Android telah terinstal lebih dari 10 miliar dan mencakup berbagai macam kategori – kategori dari hiburan dan permainan sampai finansial dan layanan bisnis. Pengembangan perangkat lunak android dan pasar Google Play relatif terbuka dan tidak terbatas. Dengan aturan seperti itu, memberikan fleksibilitas dan kebebasan pada pengembang dan pengguna, juga membuat tantangan keamanan yang lebih signifikan. [24]

2.10.1 Arsitektur Android

Jika dibandingkan dengan sistem operasi *mobile* lainnya seperti Windows Phone atau iOS, Android sangat berbeda. Aplikasi android dibangun dengan bahasa pemrograman Java dan dijalankan dalam sebuah Virtual Machine yang bernama Dalvik VM(*Virtual Machine*). VM ini adalah sebuah komponen utama pada android, karena semua aplikasi – aplikasi android dan aplikasi framework dieksekusi oleh VM ini. Seperti halnya platform lain, aplikasi – aplikasi bisa didapatkan dari tempat khusus yang sekarang dikenal dengan Google Play.

Dengan Android 5.0(Lollipop), Google membuat perubahan internal pada platform, yaitu dengan mengganti Dalvik Android Runtime untuk pebaikan performansi aplikasi.

Secara prinsip, Android adalah sistem operasi seperti UNIX yang berbasis kernel linux seri 2.6. Tapi tentu saja dengan sedikit pengembangan yaitu dengan menambahkan elemen – elemen yang dibutuhkan untuk menyediakan fungsi – fungsi dasar sebuah perangkat *mobile* termasuk menambahkan konektivitas jaringan seperti sistem selular GSM dan UMTS.



Sumber Gambar :

https://www.tutorialspoint.com/android/android_architecture.htm

Gambar 1.9 Arsitektur Android

Sistem Arsitektur Android ditunjukan pada Gambar 1.9 Arsitektur Android terdiri dari lima layer, yaitu Linux kernel, Libraries, Android runtime, application framework dan Applications.

Pada lapisan Linux Kernel, Kernel tersebut adalah sebuah kernel Linux seri 2.6 yang telah dimodifikasi dengan tujuan utama untuk mengelola memori dan daya. Sistem operasi Android berinteraksi dengan setiap komponen pada perangkat melalui lapisan ini. Lapisan ini juga mempunyai banyak driver komponen penting.

Lapisan selanjutnya terdiri dari library – library asli yang diprogram menggunakan bahasa pemrogaman C atau C++. Dengan keadaan tersebut, dapat disimpulkan bahwa android telah didesain untuk menjalankan CPU dengan daya rendah dan perangkat GPU dengan memori terbatas, library utama seperti *libc* atau *libm* telah dikembangkan untuk memastikan konsumsi memory yang rendah. Pada layer ini juga terdapat sebuah Surface Manager yang bertanggung jawab untuk akses layar, sebuah Media Framework telah dioptimalkan untuk menangani audio dan video codecs, seperti MP3, MPEG-4(*Moving Picture Experts Group*), H.264,

sebuah basis data SQL(*Structured Query Language*) dan engine untuk web browser bawaan(WebKit).

Didalam sebuah Android Runtime terdapat Dalvik Virtual Machine dan Java Core Libraries. VM ini adalah sebuah penerjemah untuk kode – kode byte yang telah di konversi dari file standar Java *jar* menjadi file *dex*, yang mana lebih kompak dan efisien daripada file – file class mengingat dengan adanya keterbatasan memori dan daya baterai dari sebuah perangkat mobile Android.

Bagian paling penting dari lapisan Application Framework adalah Activity Manager, yang bertanggung jawab untuk mengontrol siklus hidup aplikasi – aplikasi Android. Aplikasi Android apapun, berjalan masing masing dalam bagiannya masing – masing pada Dalvik VM dan bisa terdiri dari beberapa komponen, contohnya layanan, aktivitas, penyedia konten, yang mana setiap bagian tersebut dapat berinteraksi dengan satu atau banyak aplikasi yang berbeda pada satu waktu. [25]

Lapisan Applications adalah lapisan paling atas dalam arsitektur Android. Beberapa aplikasi telah terpasang sebelumnya pada setiap perangkat Android, seperti: SMS Clientapp, Dialer, Web browser dan Contact Manager. Seorang developer atau pengembang dapat membuat aplikasi sendiri dan dapat mengganti aplikasi yang telah ada.

2.10.2 Sejarah Perkembangan Versi Android

Gambar 1.10 Versi Android menunjukan versi Android yang telah dirilis hingga bulan November 2014.

Versi	Nama	Tanggal Rilis
1.5	Cupcake	April 2009
1.6	Donut	September 2009
2.0 - 2.1	Eclair	Oktober 2009
2.2 - 2.2.3	Froyo	Mei 2010
2.3 - 2.3.7	Gingerbread	Desember 2010
3.0 - 3.2.6	Honeycomb	Februari 2011
4.0 - 4.0.4	Ice Cream Sandwich	Oktober 2011
4.1 - 4.3.1	Jelly Bean	Juni 2012
4.4 - 4.4.4	Kitkat	Oktober 2013
5.0 - 5.0.2	Lollipop	November 2014

Sumber Gambar : E-Journal Android OS : A Review(2015) **[25]**

Gambar 1.10 Versi Android

2.10.2.1 Cupcake 1.5

Versi ini merupakan versi pertama yang dirilis pada tanggal 30 April 2009. Fitur dari cupcake 1.5 adalah adanya virtual keyboard pada perangkat Android Sendiri dan dukungan untuk keyboard pihak ketiga. Video recording dan playback juga mengalami kemajuan pesat pada versi 1.5. dukungan tambahan pada versi ini adalah sistem multi-task yang lebih friendly ketika mengunggah ke youtube dan picasa. [26]

2.10.2.2 Donut 1.6

Versi kedua yang dirilis adalah Donut 1.6 pada tanggal 15 September 2009, untuk versi ini menggunakan kernel linux seri 2.6.29. Versi ini memperkenalkan fitur Top paid, Top free, dan kategori – kategori aplikasi yang baru dirilis serta kemampuan untuk melakukan screenshot. Fungsi pencarian pada versi ini juga mengalami perbaikan, baik pencarian melalui suara ataupun teks. Sistem pencarian pun telah mengalami perkembangan dengan adanya bookmarks, history, contacts, apps, folder. Tampilan untuk kamera dan galeri telah mengalami perkembanga dan penggunaan kamera menjadi lebih cepat dan lebih responsif. Penambahan VPN dan keamanan WiFi juga dikembangkan.

2.10.2.3 Éclair 2.0

Versi ketiga dirilis pada 26 Oktober 2009, dengan kernel versi 2.6.32. Sistem perpesanan diperbarui pada versi ini dengan menambahkan fungsi pencarian yang memungkinkan penggunanya untuk melakukan pencarian pada pesan SMS dan MMS. Juga menambahkan fungsi auto-delete yang akan menghapus pesan lama pada perangkat setelah kotak masuk mencapai batas maksimal.

2.10.2.4 Froyo 2.2

Versi keempat dengan codename Froyo dirilis pada tanggal 20 Mei 2010, kernel yang digunakan pada versi ini yaitu seri 2.6.35. Rilis ini, membawa banyak pembaruan dari versi – versi sebelumnya. Pada versi ini lebih banyak RAM yang dikosongkan untuk meningkatkan stabilitas dan kecepatan

penggunaan aplikasi – aplikasi begitu juga dengan optimisasi memori dan performansi sistem operasi. Untuk pertama kalinya, Froyo dibangun dengan JIT compilation untuk mendapatkan performansi sistem yang lebih cepat dari sebelumnya. Kemampuan display mendukung untuk layar dengan kepadatan piksel sebesar 320 ppi. Kemajuan utama pada versi ini yaitu penambahan USB dan WIFI hotspot tethering, yang memungkinkan penggunanya untuk mengkoneksikan internet secara langsung melalui telepon, perubahan opsi tambahan untuk menggunakan kata sandi dalam bentuk PIN.

2.10.2.5 Gingerbread 2.3

Android versi kelima yang dirilis pada tanggal 6 Desember 2010. Dari semua aspek, versi ini lebih user friendly daripada versi sebelumnya. Fitur seperti kecepatan, copy/paste, manajemen daya dan UI banyak berkembang dan diperbarui. Perbedaan yang sangat menonjol pada versi ini adalah adanya fitur NFC (*Near Field Communication*) yang digunakan sebagai identitas komunikasi jarak pendek antar perangkat.

2.10.2.6 Honeycomb 3.0.X

Versi ini dirilis pada tanggal 22 Februari 2011. Pada dasarnya versi ini dibangun untuk perangkat – perangkat mobil dengan layar berukuran besar seperti tablet, serta menghadirkan banyak elemen antarmuka baru. Contohnya seperti System bar yang berada dibagian bawah layar untuk menggantikan Status bar. Bar Notifikasi yang lebih elegan dan Action Bar disematkan di bagian home screen untuk menjalankan aplikasi secara bersamaan, hal ini memberikan kenyamanan lebih pada penggunanya. Honeycomb merupakan versi pertama yang didesain untuk berjalan diatas prosesor dengan arsitektur single atau multi-core.

2.10.2.7 Ice Cream Sandwich 4.0

Android Versi ketujuh yang dirilis pada Desember 2011, dengan basis kernel linux versi 3.0.1. versi ini memperkenalkan beberapa fitur baru termasuk tampilan home screen yang baru, web browser yang telah diperbarui, contact

manager yang baru dengan integrasi media sosial, kemampuan untuk mengakses kamera dan kontrol playback musik dari layar yang terkunci, dukungan visual voicemail, face recognition untuk sistem membuka kunci menggunakan wajah, dan kemampuan untuk mengawasi dan membatasi penggunaan data internet, dan beberapa pengembangan internal.

2.10.2.8 Jellybean 4.1

Dirilis pada tanggal 9 juli 2012 dan menggunakan kernel versi 3.0.31. Jelly Bean membawa fitur akun multi – user, notifikasi yang dapat diberi aksi, widget pada saat layar terkunci, quick-setting pada bar notifikasi. Beberapa fitur yang ditingkatkan dari versi sebelumnya seperti Equalizer untuk musik, mematikan notifikasi, mengatur ikon aplikasi pada home screen dan mengakses info aplikasi.

2.10.2.9 Kitkat 4.4

Dirilis pada tanggal 3 September 2013 yang membawa beberapa peningkatan. Fitur – fitur penting pada versi ini didesain untuk berjalan pada perangkat dengan RAM minimum 512MB, dukungan untuk penyimpanan cloud seperti Google Drive telah ditambahkan dalam sistem operasi versi ini, serta sistem pencetakan(*print*) dokumen yang dapat dilakukan melalui WiFi atau Bluetooth.

Perubahan signifikan pada setiap versi yaitu :

1. Cupcake : Perbaikan UI (*User Interface*) dari seluruh element utama, aplikasi – aplikasi rotasi berbasis akselerometer, on-screen soft keyboard, video recording dan playback, dukungan stereo Bluetooth.
2. Donut : kolom pencarian cepat, VPN(*Virtual Private Network*) dan dukungan 802.1x, indikator penggunaan baterai.
3. Éclair : dukungan Bluetooth 2.1, tambahan mode – mode kamera, dukungan e – mail dan akun lebih dari satu.
4. Froyo : kemampuan untuk tethering dan hotspot WiFi, JIT (*Just In Time*) compiler, dukungan Adobe Flash.

5. Gingerbread : NFC (*Near Field Communications*), tambahan dukungan sensor(gyroscope, putaran vektor, akselerasi linear, gravitasi, barometer), dukungan kamera lebih dari satu, dukungan resolusi layar besar(tablet), Google Talk.
6. Honeycomb : tablet-only Android update, aksesoris – aksesoris konektivitas USB(*Universal Serial Bus*), kunci WiFi high-performance.
7. Ice Cream Sandwich : Wifi Direct, membuka kunci menggunakan wajah, beberapa perbaikan (stabilitas, optimisasi, rotasi layar, grafis).
8. Jelly Bean : Google Wallet, Google Now, USB audio, Photo Sphere panorama photos, akun user lebih dari satu, dukungan Miracast wireless display.
9. KitKat – NFC host card emulation, new experimental runtime virtual machine(ART), dukungan Bluetooth MAP(*Message Access Profile*).
10. Lollipop : Penggantian Dalvik oleh ART dengan AOT(*Ahead-Of-Time*) compilation, dukungan untuk CPU 64-bit, dukungan OpenGL ES 3.1. aktivitas terbaru dengan tugas dari aplikasi, proyek Volta(perbaikan penggunaan baterai), masukan dan keluaran audio melalui perangkat USB. [25]

2.11 *Framework*

Framework dapat diartikan sebagai koleksi atau kumpulan potongan – potongan program yang disusun dan diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal. Beberapa keuntungan yang dapat diperoleh dari penggunaan *framework* adalah :

1. Waktu pembuatan aplikasi menjadi jauh lebih singkat.
2. Kode aplikasi menjadi lebih mudah dibaca, karena sedikit dan sifatnya pokok, detailnya adalah kode dari *framework.*
3. Aplikasi menjasi lebih mudah diperbaiki, karena tidak perlu fokus ke semua komponen aplikasi, terutama kode sistem *framework.*

4. Pikiran pengembang menjadi lebih terfokus pada kode alur permasalahan aplikasi, apa yang ditampilkan dan layanan apa saja yang diberikan oleh aplikasi.
5. Jika dikerjakan secara *team work*, maka akan lebih terarah karena sistem *framework,* mengharuskan adanya keteraturan peletakan kode. [27]

2.12 Ionic

Ionic adalah sebuah *framework* dapat digunakanuntuk membangun sebuah aplikasi *mobile* dengan menggunakan teknologi weh seperti HTML5, CSS dan JavaScript. Tetapi bukan hanya itu yang dapat dilakukan oleh Ionic. Ionic menyediakan komponen – komponen yang dapat digunakan oleh pengembang untuk mengakses fitur – fitur aplikasi *mobile* layaknya aplikasi *mobile* sebenarnya (*native*). Beberapa fitur – fitur seperti *gesture, popups, modals* semuanya disediakan ionic untuk digunakan oleh pengembang untuk melengkapi kebutuhan aplikasi yang diperlukan.

Framework ionic dibangun menggunakan AngularJS, yang merupakan sebuah *framework* JavaScript yang telah banyak digunakan. Sebelumnya, membangun arsitektur aplikasi – aplikasi hybrid sangat sulit untuk dilakukan, tetapi dengan menggunakan Angulaer, kita dapat membuat aplikasi – aplikasi *mobile* menggunakan sebuah teknik yang dinamakan SPA (*Single Page Application*). Angular juga menjadikan aplikasi – aplikasi yang dibangun lebih mudah dikelola untuk pengembangan dan bekerja secara *team work* karena ionic memungkinkan pengembang untuk dapat dengan mudah menambah, mengkostumisasi fitur – fitur atau library – library aplikasi. [28]

2.12.1 Fitur – fitur Ionic

Ionic menyediakan beberapa fitur yang dapat membantu pengembang untuk membuat aplikasi hybrid dengan fungsi dan tampilan yang menarik dalam waktu singkat. Fitur – fitur dari ionic ada dalam 3 kategori, seperti berikut :

2.12.1.1 Fitur CSS

Ionic dihadirkan dengan CSS library yang menyediakan pengembang untuk membuat *User Interface* yang menarik. CSS Style pada ionic di *generate* menggunakan SASS, sebuah *preprocessor* CSS yang digunakan untuk manipulasi yang lebih *advance*.

Beberapa fitur CSS yang telah dimasukkan kedalam sebuah *framework* ionic yaitu :

1. Buttons
2. Cards
3. Header dan Footer
4. Lists
5. Forms Elements
6. Grid System

Semua fitur – fitur tersebut telah disediakan untuk pengembang dan dapat dikostumisasi dengan mudah. Fitur – fitur tersebut sudah memiliki tampilan yang sama seperti layaknya komponen pada aplikasi *mobile* sebenarnya.

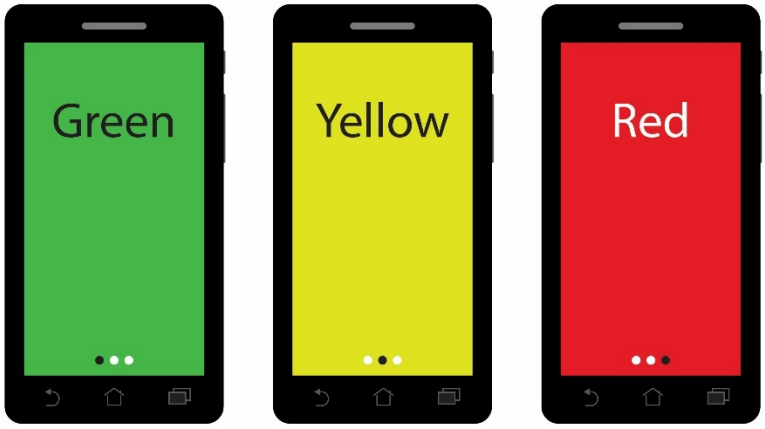
2.12.1.2 Fitur JavaScript

Fitur JavaScript merupakan hal yang yang paling penting untuk membangun aplikasi menggunakan *Framework* Ionic. Fitur – fitur ini memungkinkan pengembang untuk mengkostumisasi aplikasi atau bahkan menyediakan fungsi – fungsi pembantu yang dapat digunakan untuk memberikan kemudahan dalam pembuatan aplikasi.

Beberapa fitur JavaScript pada Ionic yaitu :

1. Modal
2. Slide Box

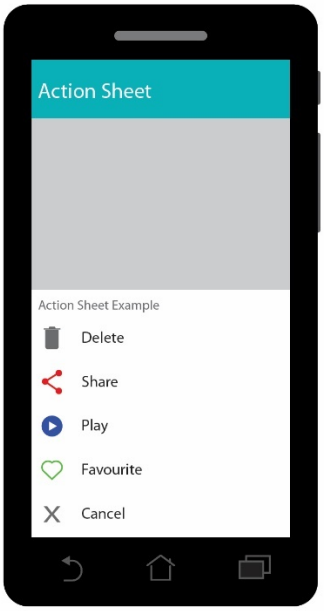
   Contoh Slide Box pada Ionic dapat dilihat pada Gambar 1.11 Ionic Slide Box



Gambar 1.11 Ionic Slide Box

3. Action Sheet

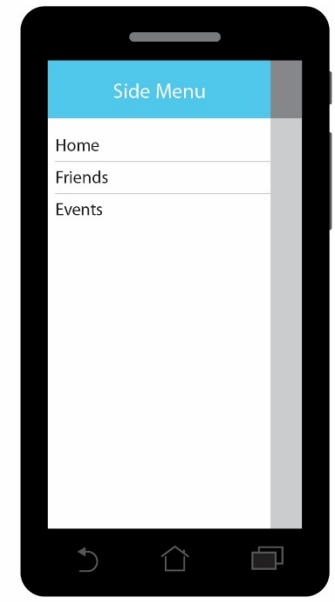
Contoh Action Sheet dari sebuah *framework* Ionic dapat dilihat pada Gambar 1.12 Ionic Action Sheet



Gambar 1.12 Ionic Action Sheet

4. Side Menu

   Contoh Side Menu Ionic dapat dilihat pada Gambar 1.13 Ionic Side Menu



Gambar 1.13 Ionic Side Menu

5. Tabs

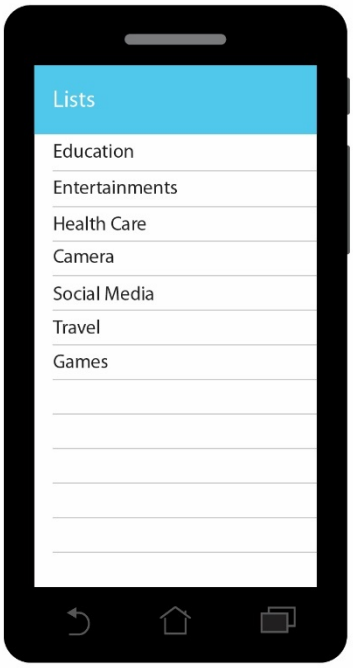
   Contoh Tabs dari *framework* Ionic dapat dilihat pada Gambar 1.14 Ionic Tabs



Gambar 1.14 Ionic Tabs

6. Lists

   Contoh Lists pada *framework* Ionic dapat dilihat pada Gambar 1.15 Ionic Lists



Gambar 1.15 Ionic Lists

7. Collection repeat

2.12.1.3 Ionic CLI

Ionic CLI adalah alat yang sangat penting dalam *framework* Ionic yang memungkinkan pengembang atau programmer untuk menggunakan perintah – perintah Ionic dengan menggunakan *command line/terminal*. Dengan alat ini juga kita dapat mengakses fitur – fitur Ionic yang membuat pengembangan aplikasi kita menjadi lebih singkat. Bisa dibilang bahwa bagian ini merupakan bagian paling penting dari Ionic dan juga merupakan fitur yang akan paling banyak digunakan dalam pengembangan aplikasi.

Fitur – Fitur pada Ionic CLI memungkinan pengguna untuk melakukan beberapa aksi sebagai berikut :

1. Membuat proyek Ionic
2. Memberikan perintah – perintah Cordova
3. Pengetesan dan Pengembangan Aplikasi

4. Ionic splash/generator ikon
5. Ionic labs
6. SASS
7. Mengakses tools Ionic.IO

2.13 Database (Basis Data)

Database/basis data terdiri dari 2 susku kata, yaitu data dan base/basis. Data dapat diartikan sebagai representasi fakta dunia nyata yang mewakili suatu objek, misalnya manusia, hewan, barang, peristiwa, konsep, dan lain sebagainya yang direkam dalam bentuk huruf, teks, simbol, angka, suara, gambar, dan lainnya. Sedangkan basis/base dapat diartikan sebagai tempat berkumpul, sarang atau gudang untuk menyimpan sesuatu. Dengan demikian, basis data/database dapat diartikan sebagai tempat berkumpul/menyimpan data – data suatu benda atau kejadian yang saling berhubungan. [29]

Database juga dapat diartikan sebagai kumpulan file, arsip yang saling berhubungan yang disimpan di dalam media elektronis. Data yang disimpan tersebut tidak dibiarkan begitu saja, namun dikelola dan diorganisasi yang dikenal dengan Database Management System (DBMS). Dengan demikian, data yang tersimpan di dalam database dapat disusun dengan rapi dan terstruktur sehingga memudahkan dalam mendapatkan informasi yang bersangkutan dengan database.

2.13.1 Manfaat Database

Database sangat bermanfaat untuk mengatasi berbagai masalah yang sering terjadi dalam penyusunan data. Masalah dalam penyusunan data dapat berupa redudansi dan inkonsistensi data atau kesulitan pengaksesan data. Lebih jelas akan dipaparkan manfaat database sebagai berikut :

1. Redudansi dan Inkonsistensi Data

   Redudansi adalah penyimpanan data yang sama dalam dua tempat (tabel) yang berbeda sehingga dapat menyebabkan inkonsistensi data. Inkonsistensi data, yaitu data yang sama dan disimpan pada dua tabel dapat memiliki nilai yang berbeda. Sebagai contoh, dalam database perpustakaan, di mana data

nama anggota disimpan di tabel anggota dan tabel transaksi peminjaman (redudansi). Apabila nama anggota pada tabel anggota diubah, maka nama pada tabel transaksi tidak berubah (inkonsistensi data) padahal data tersebut menjelaskan objek yang sama.

2. Kesulitan Pengaksesan data

Kesulitan pengaksesan data pada database dengan program yang dapat muncul dapat diatasi dengan DBMS. DBMS mampu mengambil data secara langsung dengan bahasa yang sudah familiar dan mudah digunakan, misalnya SQL. Sebagai contoh, apabila pada suatu waktu akan dibutuhkan laporan tentang adventori buku yang terbit tahun 2009, sedangkan program untuk itu belum dibuat, maka dengan mudah DBMS menyelesaikannya.

3. Isolasi Data untuk Standarisasi

Isolasi data untuk standarisasi berkaitan dengan format penyimpanan data tersebut. Sebagai contoh, data disimpan pada file dengan bahasa tertentu, misalnya Java, C/C++,Visual Basic, Pascal, dan lainnya, tentunya hal ini menyulitkan programmer untuk mengakses data tersebut. Untuk itu, data dalam suatu database harus disimpan dalam format yang sama. Sebagai contoh, dalam format JavaDB/Derby, MySQL, MsQL.

4. Multiple User atau Banyak Pemakai

Database dirancang agar data diakses oleh banyak pemakai(*user*) dan banyak program yang sama dalam waktu bersamaan maupun berbeda. Hal ini mungkin terjadi dengan database, dikarenakan dalam rangka mempercepat daya guna sistem dan mendapatkan responsi waktu yang cepat, beberapa sistem mengizinkan banyak pemakai untuk meng-*update* data secara simultan. Salah satu alasan mengapa basis data dibangun adalah karena nantinya data tersebut akan digunakan oleh banyak orang dalam waktu yang berbeda, diakses oleh program yang sama, tetapi berbeda orang dan waktu. Semua ini mungkin akan terjadi karena data yang diolah dan tidak tergantung dan menyatu dalam program, tetapi terlepas dalam satu kelompok data.

5. Masalah Keamanan

Walaupun database dibuat untuk diakses oleh banyak pemakai, database tetap menjaga keamanan data. Keamanan data dilakukan dengan memberikan hak akses tertentu bagi para user tersebut dan tidak semua user diizinkan untuk mengakses database. Sebagai misal data transaksi peminjaman buku tidak boleh diakses oleh anggota. Keamanan dapat diatur melalui program yang dirancang oleh programmer atau melalui fasilitas keamanan sistem operasi.

6. Masalah Kesatuan Data

Database memiliki tabel – tabel yang saling berkaitan/berhubungan. Hubungan antartabel secara teknis dikaitkan/dihubungkan dengan field kunci, sehingga dikenal dengan kunci primer (*primary key*) dan kunci tamu (*foreign key*) dalam suatu tabel. Sebagai contoh tabel buku dan tabel transaksi peminjaman buku, secara teknis buku mempunyai field kunci, sebagai misal field *id* dan tabel peminjaman juga mempuyai field *id* di mana mempunyai struktur yang sama dengan field *id* pada tabel buku. Maka field *id* pada tabel buku disebut kunci primer dan field *id* pada tabel peminjaman disebut kunci tamu. Kedua field yang disebut kunci inilah yang menghubungkan antar tabel, sehingga dalam database mempunyai kesatuan data yang saling berkaitan.

7. Masalah Kebebasan Data

Kebebasan data yang dimaksud adalah kemudahan pembacaan data dalam database dapat diakses dengan mudah dalam menampilkan informasi yang dibutuhkan. Sebagai contoh, ketidakbebasan data adalah aplikasi program database yang telah dibuat secara khusus, di mana apabila suatu struktur database diubah strukturnya, maka progammer harus mengubah program aplikasi tersebut untuk mengakses data pada struktur database yang baru. Hal ini berbeda dengan database DBMS yang bersifat bebas, di mana data dengan mudah ditampilkan, dibaca, diubah, dan sebagainya dengan perintah – perintah tertentu.

8. Kemandirian Program Data

Program aplikasi berbasis database terpisah dengna data yang digunakannya, di mana data disimpan dalam DBMS dengan format tertentu. Dengan demikian, program aplikasi dan database berdiri sendiri – sendiri, namun antara program aplikasi dan database mempunyai hubungan yang erat. Hubungan tersebut adalah program aplikasi dapat mengakses data yang disimpan dalam DBMS. Hal ini dapat meningkatkan pengembangan aplikasi dan produktivitas perawatan.

9. Meningkatkan Kualitas Data

Untuk meningkatkan kualitas data dapat dilakukan dengan aturan validasi data melalui program aplikasi. Validasi data adalah penyaringan data – data yang hanya boleh disimpan dalam database, misal field *id* hanya boleh menyimpan data dengan format A11 (satu karakter diikuti 2 angka), atau hanya menerima input data berupa angka.

10. Akses Data Lebih Baik atau Responsif

Dengan menggunakan bahasa Structured Query Language (SQL), pengaksesan data jauh lebih baik dan cepat. Hal ini dikarenakan DBMS menggunakan bahasa SQL yang mempunyai bahasa yang simpel dan mudah digunakan serta telah menjadi standar bahasa pembuatan database.

2.13.2 Database Management System (DBMS)

Database Management System atau disingkat DBMS adalah perangkat lunak (*software*) yang berfungsi untuk mengelola database. Mulai dari membuat database itu sendiri, sampai dengan proses yang berlaku dalam database tersebut, baik berupa entry, edit, hapus, query terhadap data, membuat laporan dan lain sebagainya secara efektif dan efisien. Salah satu jenis DBMS yang sangat terkenal saat ini adalah Relational DBMS (RDBMS). RDBMS merepresentasikan data dalam bentuk tabel – tabel yang saling berhubungan. Sebuah tabel disusun dalam bentuk baris (*record*) dan kolom (*field*). [30]

Ada tiga kelompok perintah yang digunakan dalam mengelola dan mengorganisasikan data dalam RDBMS, yaitu :

1. Data Definition Language

   Merupakan perintah yang digunakan oleh seorang Database Administrator untuk mendefinisikan struktur database, baik membuat tabel baru, menentukan struktur penyimpanan tabel, model relasi antar tabel dan validasi data.

2. Data Manipulation Language (DML)

   Perintah – perintah yang digunakan untuk memanipulasi dan mengambil data pada suatu database. Manipulasi yang dapat dilakukan terhadap data adalah :

   a. Penambahan data
   b. Penyisipan data
   c. Penghapusan data
   d. Pengubahan data

   DML merupakan bahasa yang memudahkan pengguna dalam mengakses database. Ada dua jenis DML, yaitu :

   a. Prosedural, mengharuskan pengguna menentukan spesifikasi data apa yang dibutuhkan dan bagaimana cara mendapatkannya. Contoh paket bahasanya adalah dBase III, FoxBase, FoxPro.
   b. Non Prosedural, pengguna hanya menentukan data apa yang dibutuhkan tanpa harus tahu cara mendapatkannya. Contoh paket bahasanya diberi nama Structural Query Language (SQL).

3. Data Control Language

   Bagian ini berkenaan dengan cara mengendalikan data, seperti siapa saja yang bisa melihat isi data, bagaimana data bisa digunakan oleh banyak user. Lebih mengarah ke segi sekuritas data.

2.14 MySQL

MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius. MySQL database server adalah RDBMS (*Relational Database Management System*) yang dapat menangani data yang bervolume besar. Meskipun begitu, tidak menuntut *resource* yang besar. MySQL adalah database yang paling populer diantara database – database yang lain.



Sumber Gambar : https://www.mysql.com/about/legal/logos.html

Gambar 1.16 Logo MySQL

MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan multi user. MySQL memiliki dua bentuk lisensi yaitu free software dan shareware. Untuk penelitian ini akan menggunakan MySQL yang free software karena bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*general public license*), yang dapat didapatkan pada *website* resminya di http://www.mysql.com [31]. MySQL sudah cukup lama dikembangkan beberapa fase penting dalam pengembangan MySQL adalah sebagai berikut :

1. MySQL dirilis pertama kali secara internal pada 23 Mei 1995
2. Versi Windows dirilis pada 8 Januari 1998 untuk Windows 95 dan Windows NT.
3. Versi 3.23 : beta dari Juni 2000, dan dirilis pada Januari 2001.
4. Versi 4.0 : beta dari Agustus 2002, dan dirilis pada Maret 2003 (unions).
5. Versi 4.1 : beta dari bulan Juni 2004, dirilis pada bulan Oktober 2004 (R-trees dan B-trees, subqueries, prepared statements).

6. Versi 5.0 : beta dari bulan Maret 2005, dirilis pada Oktober 2005.
7. Sun Microsystems Membeli MySQL AB pada tanggal 26 Februari 2008.
8. Versi 5.1 : dirilis 27 November 2008 (event scheduler, partitioning, plug – in API, row – based replication, server log table).

2.14.1 Kelebihan dan Keuntungan Menggunakan MySQL

Ketika menggunakan sebuah software DBMS perlu diperhitungkan apa kebutuhan yang paling penting. Apakah performa, support, fitur – fitur SQL, kondisi keamanan dalam license atau harga. Dengan pertimbangan – pertimbangan tersebut, database MySQL memiliki beberapa kelebihan dan keuntungan dibandingkan dengan database lain. Beberapa diantaranya adalah :

1. MySQL merupakan Software DBMS yang bersifat OpenSource sehingga bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar pada pembuatnya.
2. MySQL dapat diakses melalui protokol ODBS (*Open Database Connectivity*) buatan Microsoft. Ini menjadikan MySQL dapat diakses oleh banyak software.
3. Semua klien dapat mengakses server dalam satu waktu, tanpa harus menunggu yang lain untuk mengakses database.
4. Database MySQL mengerti bahasa SQL (Structured Query Language).
5. Database MySQL dapat diakses darimana saja melalui internet dengan hak akses tertentu.
6. MySQL dapat berjalan di berbagai Sistem Operasi seperti Linux, Windows dan Solaris.

2.14.2 SQL

SQL adalah singkatan dari *Structured Query Language.* SQL sering dibaca dengan “Sequel” adalah bahasa terstruktur yang digunakan untuk query, meng – update dan memanipulasi database. Bahasa ini digunakan untuk memuat, mengurutkan dan menyaring suatu data, sehingga dihasilkan data yang spesifik dari sebuah database.

2.14.2.1 SQL dalam MySQL

SQL dalam MySQL dapat dikelompokkan menjadi beberapa kelompok perintah, meskipun sebetulnya pengelompokan ini hanyalah untuk membedakan fungsi – fungsi utamanya saja. Kelompok SQL tersebut adalah seperti berikut ini :

1. Data Definition Statements

   Adalah kelompok SQL untuk melakukan perintah dalam tingkat definisi database, tabel dan incex. Terdiri dari perintah – perintah Alter database, Alter Table, Create Database, Create Index, Create Table, Drop Database, Drop Index, Drop Table, dan Rename Table.

   Contoh syntax :

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
    [create_specification] ...
create_specification:
    [DEFAULT] CHARACTER SET [=] charset_name
  | [DEFAULT] COLLATE [=] collation_name
```

Gambar 1.17 Create Database Syntax

2. Data Manipulation Statements

   Adalah kelompok SQL untuk melakukan perintah manipulasi data. Terdiri dari perintah – perintah Delete, Do, Handler, Insert, Load Data InFile, Select, Sub Query, Truncate, dan Update.

   Contoh Syntax :

```
INSERT INTO tbl_name () VALUES();
```

Gambar 1.18 Insert Data Syntax

3. MySQL Utility Statements

Adalah kelompok SQL bagian utility. Terdiri dari perintah – perintah Describe, Use, Explain dan Help.

Contoh Syntax :

```
HELP 'search_string'
```

Gambar 1.19 Help Syntax

4. MySQL Transactional and Locking Statements

Adalah kelompok SQL untuk melakukan perintah transaksi dan locking (penguncian). Terdiri dari perintah- perintah Start Transaction, Commit, Rollback., SavePoint, Rollback to SavePoint, Lock Tables, Unlock Tables, dan Set Transaction.

Contoh Syntax :

```
START TRANSACTION;
SELECT  @A:=SUM(salary)  FROM  table1  WHERE
type=1;
UPDATE table2 SET summary=@A WHERE type=1;
COMMIT;
```

Gambar 1.20 Transaction Syntax

2.15 Peramalan (*Forecasting*)

Peramalan adalah suatu proses memperkirakan secara sistematis tentang apa yang mungkin terjadi dimasa yang akan datang berdasarkan informasi masa lalu dan sekarang yang dimiliki agar kesalahannya dapat diperkecil. Peramalan tidak memberikan jawaban pasti tentang apa yang akan terjadi, melainkan berusaha mencari pendekatan tentang apa yang akan terjadi sehingga dapat memberikan kontribusi dalam menentukan keputusan terbaik. [32]

2.15.1 Jenis – jenis Peramalan

Peramalan dapat dilakukan secara kuantitatif maupun kualitatif. Pengukuran secara kuantitatif menggunakan metode statistik, sedangkan pengukuran secara kualitatif berdasarkan pendapat (*judgment*) dari yang melakukan peramalan. Berkaitan dengan itu, dalam peramalan dikenal dengan istilah prakiraan dan prediksi. [33]

Prakiraan didefinisikan sebagai proses peramalan suatu variabel (kejadian) di masa datang dengan berdasarkan data variabel itu pada masa sebelumnya. Data masa lampau itu secara sistematik digabugnkan dengan menggunakan suatu metode tertentu dan diolah untuk memperoleh prakiraan keadaan pada masa datang. Sementara, prediksi adalah proses peramalan suatu variabel di masa datang dengan lebih mendasarkan pada pertimbangan intuisi daripada data masa lampau. Meskipun lebih menekankan pada intuisi, dalam prediksi juga sering digunakan data kuantitatif sebagai pelengkap informasi dalam melakukan peramalan. Dalam prediksi, peramalan yang baik sangat tergantung pada kemampuan, pengalaman dan kepekaan dari si pembuat ramalan.

Perbedaan antara penggunaan prakiraan dan prediksi dapat dijelaskan sebagai berikut. Suatu perusahaan ingin meramalkan berapa permintaan pasar atas produknya pada periode akan datang. Perusahaan itu dapat melakukan prakiraan dengan menggunakan data penjualan periode sebelumnya untuk mengetahui taksiran permintaan pasar pada masa datang. Namun, jika akan mengeluarkan produk baru, perusahaan itu melakukan prediksi untuk mengetahui berapa jumlah yang dapat diserap pasar karena belum mempunyai data penjualan masa lampau. Dalam hal ini, perusahaan dapat menggunakan data kuantitatif seperti data penjualan produk sejenis dari perusahaan lain sebagai masukan dalam melakukan prediksi.

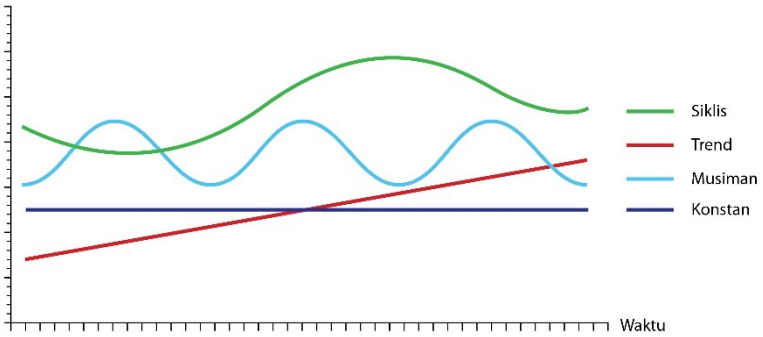
Berdasarkan horison waktu, peramalan dapat dikelompokkan dalam tiga bagian, yaitu peramalan jangka panjang, peramalan jangka menengah, dan peramalan jangka pendek. Peramalan jangka panjang banyak menggunakan pendekatan kualitatif sedangkan peramalan jangka menengah dan pendek biasanya menggunakan pendekatan kuantitatif.

Metode kuantitatif yang digunakan dalam prakiraan, pada dasarnya dapat dikelompokkan dalam dua jenis, yaitu metode serial waktu dan metode eksplanatori. Metode serial waktu (deret berkala, *time series*) adalah metode yang digunakan untuk menganalisis serangkaian data yang merupakan fungsi dari waktu. Metode ini mengasumsikan bahwa beberapa pola atau kombinasi pola selalu berulang sepanjang waktu, dan pola dasar dapat diidentifikasi semata – mata atas dasar data historis dari serial itu. Tujuan analisis ialah untuk menemukan pola deret variabel yang bersangkutan berdasarkan atas nilai variabel pada masa sebelumnya, dan mengekstrapolasikan pola itu untuk membuat peramalan nilai variabel itu pada masa datang.

Metode eksplanatori mengasumsikan bahwa nilai suatu variabel merupakan fungsi dari satu atau beberapa variabel lain. Misalnya, jumlah penjualan suatu komoditi dapat diprediksi dari nilai harga komoditi itu, pendapatan konsumen, jumlah konsumen, dan harga produk substitusi/komplementer. Dengan kata lain, permintaan produk merupakan fungsi dari variabel – variabel tersebut. Kegunaan metode eksplanatori ialah untuk menemukan bentuk hubungan antara suatu variabel dengan variabel – variabel lain, dan menggunakannya untuk meramalkan nilai variabel tak bebas (yang diramalkan, dependen) terhadap perubahan dari variabel bebasnya.

2.15.2 Metode Serial Waktu

Analisis serial waktu dimulai dengan memplot data pada suatu skala waktu (membuat diagram pencari *scatter diagram*) kemudian dengan mempelajari plot tersebut, dan akhirnya mencari suatu bentuk atau pola yang konsisten atas data. Pola ini serangkaian data dalam serial waktu yang dapat dikelompokkan kedalam pola dasar seperti yang dapat dilihat pada



Sumber Gambar : E-Book Manajemen Operasi **[33]**

Gambar 1.21 Pola Dasar dalam Serial Waktu

a. Horizontal (konstan), yaitu apabila data berfluktuasi di sekitar rata – rata secara stabil. Polanya berupa garis lurus mendata. Pola seperti ini biasanya terdapat dalam jangka pendek atau menengah. Jarang sekali suatu variabel memiliki pola konstan dalam jangka panjang.
b. Kecenderungan (*trend*), yaitu apabila data mempunya kecenderungan, baik yang arahnya meningkat atau menurun dari waktu ke waktu. Pola ini disebabkan antara lain oleh bertambahnya populasi, perubahan pendapatan, dan pengaruh budaya.
c. Musiman (*seasonal*) yaitu apabila polanya merupakan gerakan yang berulang – ulang secara teratur dalam setiap periode tertentu, misalnya tahunan, triwulanan, bulanan atau mingguan. Pola ini biasanya berhubungan dengan faktor iklim/cuaca atau faktor yang dibuat oleh manusia seperti liburan dan hari besar.
d. Siklus (*cyclical*), yaitu apabila data dipengaruhi oleh fluktuasi ekonomi jangka panjang, seperti daur hidup bisnis. Perbedaan utama antara pola musiman dengan siklus adalah pola musiman mempunya panjang gelombang yang tetap dan terjadi pada jarak waktu (durasi) yang tetap, sedangkan pola siklus memiliki jarak waktu yang lebih panjang dan bervariasi dari satu siklus ke siklus lainnya.
e. Residu atau variasi acak, yaitu apabila data tidak teratur sama sekali. Data yang bersifat residu tidak dapat digambarkan.

Pengolahan data kuantitatif dari serial waktu dapat dilakukan dengan beberapa metode dasar, sebagai berikut :

1. Rata – rata bergerak (*Moving Average*)
2. Pemulusan Eksponensial (*Exponential Smoothing*)
3. Dekomposisi

Sejalan dengan perkembangan ilmu pengetahuan dan teknologi, setiap metode dasar telah dikembangkan dan memiliki berbagai derivasi/turunannya. Notasi yang dipergunakan dalam metode serial waktu dapat dilihat pada Gambar 1.22 Notasi Metode Serial Waktu.

Periode/waktu	1	2	...	t-1	t	t+1	...	t+m
Nilai Observasi	X_1	X_2	...	X_{t-1}	X_t			
Nilai Peramalan	F_1	F_2	...	F_{t-1}	F_t	F_{t+1}	...	F_{t+m}
Nilai Kesalahan (X-F)	e_1	e_2	...	e_{t-1}	e_t			

Sumber Gambar : E-Book Manajemen Operasi [33]

Gambar 1.22 Notasi Metode Serial Waktu

t merupakan periode saat ini (periode berjalan). Data periode berjalan biasanya belum dapat diperoleh, sehingga seringkali dipergunakan data proyeksi. Nilai observasi hanya bisa diperoleh sampai Xt, sedangkan nilai prakiraan dapat dilakukan sampai t+m, dimana m menunjukan berapa periode ke depan dari periode berjalan. Nilai kesalahan merupakan besarnya kesalahan dari nilai prakiraan terhadap nilai aktualnya.

2.16 Metode Pemulusan Eksponensial (*Exponential Smoothing*)

Metode peramalan *Exponential Smoothing* merupakan sebuah metode peramalan yang bekerja dengan menggunakan konstanta untuk memberikan bobot pada nilai saat ini dan sebelumnya untuk menentukan sebuah peramalan baru. Nilai konstanta ini mempengaruhi respon peramalan. Oleh karena itu, nilai konstanta ini akan mempengaruhi kesalahan peramalan. Telah banyak upaya difokuskan untuk mencari nilai konstanta yang cocok untuk digunakan. Salah satu pendekatannya adalah menggunakan konstanta *smoothing* yang dapat meminimalisir kesalahan peramalan. [34]

2.16.1 Metode Pemulusan Eksponensial Tunggal

Metode pemulusan eksponensial tunggal (*single exponential smoothing*) menambahkan parameter α dalam model untuk mengurangi faktor kerandoman [33]. Nilai prakiraan dapat dicari dengan menggunakan rumus berikut.

$$F_{t+1} = \alpha.X_t + (1 - \alpha).F_t$$

Sumber Gambar : E-Book Manajemen Operasi [33]

Gambar 1.23 Persamaan Pemulusan Eksponensial Tunggal

Di mana :

X_t = data pada periode t

α = faktor/konstanta pemulusan

F_{t+1} = prakiraan untuk periode t

Berbeda dengan metode rata – rata bergerak (*Moving Average*) yang hanya menggunakan data observasi N periode terakhir dalam melakukan prakiraan. Metode pemulusan eksponensial tunggal mengikutsertakan data dari semua periode. Setiap data pengamatan mempunyai kontribusi dalam penentuan nilai prakiraan periode sesudahnya. Dalam perhitungan prakiraan, harus tersedia data observasi dan hasil prakiraan periode terakhir. Di mana, hasil prakiraan periode terakhir sudah mengandung nilai – nilai observasi dari prakiraan sebelumnya.

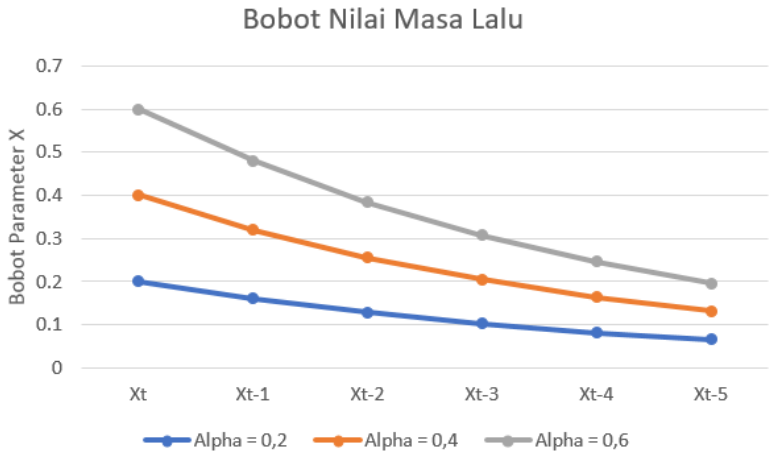
Istilah eksponensial dalam metode ini berasal dari pembobotan (faktor pemulusan) dari periode – periode sebelumnya yang membentuk hubungan eksponensial. Jika persamaan pada Gambar 1.23 Persamaan Pemulusan Eksponensial Tunggal diperluas dengan mengganti F_t dengan komponennya, dan meneruskan proses substitusi F_{t+1} dengan komponennya, dan seterusnya, akan diperoleh hasil sebagai berikut.

$$\begin{aligned} F_{t+1} &= \alpha X_t + (1 - \alpha) F_t \\ &= \alpha X_t + (1 - \alpha) \{ \alpha X_{t-1} + (1 - \alpha) F_{t-1} \} \\ &= \alpha X_t + \alpha (1 - \alpha) X_{t-1} + (1 - \alpha)^2 \{ \alpha X_{t-2} + (1 - \alpha) F_{t-2} \} \\ &= \alpha X_t + \alpha (1 - \alpha) X_{t-1} + \alpha (1 - \alpha)^2 X_{t-2} + \dots \\ &\quad + \alpha (1 - \alpha)^{N-1} X_{t-(N-1)} + (1 - \alpha)^N F_{t-(N-1)} \end{aligned}$$

Dari persamaan tersebut dapat diketahui bahwa koefisien X dari waktu ke waktu membentuk hubungan eksponensial terhadap nilai α.

Misalnya, untuk α = 0,2 bobot parameter X_t, X_{t-1}, X_{t-2}, X_{t-3}, …, $X_{t(N-1)}$ berturut – turut adalah 0,2; 0,2(0,8); 0,2$(0,8)^2$; 0,2$(0,8)^3$; … 0,2$(0,8)^{N-1}$ atau sama dengan 0,2; 0,16; 0,13; 0,10; … 0,03 (untuk N = 10).

Dalam bentuk grafik, hubungan eskponensial itu digambarkan sebagai seperti pada Gambar 1.24 Bobot yang Diberikan pada Nilai Observasi Masa Lalu



Gambar 1.24 Bobot yang Diberikan pada Nilai Observasi Masa Lalu

Semakin besar nilai α garis persamaan semakin membentuk hubungan eskponensial yang nyata, sebaliknya semakin kecil α semakin ke arah linear.

Contoh perhitungan prakiraan dengan menggunakan metode pemulusan eksponensial tunggal ditunjukan pada Tabel 1.1.

Tabel 1.1 Prakiraan dengan Metode Pemulusan Eksponensial Tunggal

Periode (t)	Nilai Pengamatan (X)	Nilai prakiraan (F)	
		α = 0,1	α = 0,2
1	41	*41,0*	*41,0*
2	40	41,0	41,0
3	42	40,9	40,8
4	43	41,0	41,0
5	41	41,2	41,4
6	42	41,2	41,3
7	41	41,3	41,5
8	40	41,2	41,4
9	43	41,1	41,1
10	-	41,3	41,5

Nilai prakiraan pada periode t = 1 berupa nilai inisial (asumsi). Nilai ini bisa diperoleh dengan cara menganggap nilai prakiraan pada periode itu sama dengan nilai sebenarnya (dalam contoh ini = 41), atau rata – rata dari nilai beberapa periode. Untuk konstanta pemulusan dapat menggunakan setiap nilai diantara 0 sampai 1. Nilai konstanta pemulusan terbaik adalah yang dapat memberikan ketelitian prakiraan tertinggi.

2.16.2 Metode Pemulusan Eksponensial Linear

Metode pemulusan eksponensial tunggal hanya akan efektif apabila serial data yang diamati memiliki pola horizontal (stasioner). Jika metode itu digunakan untuk serial data yang memiliki unsur trend (kecenderungan) yang konsisten, nilai – nilai parakiraan akan selalu berada di belakang nilai aktualnya (terjadi *lagging* yang terus menerus). Metode yang tepat untuk melakukan prakiraan serial data yang memiliki unsur kecenderungan adalah metode pemulusan eksponensial linear (*linear exponential smoothing*). Salah satu metode pemulusan eksponensial linear, yang diperkenalkan oleh Holt, menggunakan persamaan sebagai berikut.

$$S_t = \alpha.X_t + (1 - \alpha)(S_{t-1} + T_{t-1})$$
$$T_t = \beta.(S_t - S_{t-1}) + (1 - \beta).T_{t-1}$$
$$F_{t+m} = S_t + T_t.m$$

Gambar 1.25 Persamaan Pemulusan Eksponensial Linear

Pemulusan eksponensial linear Holt menambahkan persamaan T_t untuk memperoleh pemulusan trend. Selanjutnya, dengan menggabungkan persamaan trend T_t dan persamaan pemulusan standar S_t dapat dihasilkan persamaan F_t. metode Holt menggunakan dua parameter, α dan β, yang masing – masing nilainya dapat dipilih dari setiap angka antara 0 sampai 1. Kedua parameter itu dapat mempunyai nilai yang sama atau berbeda besarnya. Contoh penggunaan metode ini dalam suatu serial data yang memiliki unsur trend ditunjukan pada Tabel 1.2.

Tabel 1.2 Prakiraan dengan Metode Pemulusan Eksponensial Linear Holt

Periode (t)	Nilai Pengamatan (X)	Nilai Prakiraan		
		S α = 0,2	T β = 0,3	F
1	500	*500,0*	*7,00*	-
2	524	510,4	8,02	507,0
3	521	518,9	8,17	518,4
4	530	527,7	8,35	527,1
5	540	536,8	8,59	536,0
6	542	544,7	8,38	545,4
7	555	553,5	8,49	553,1
8	550	559,6	7,78	562,0
9	-	-	-	567,4 (m = 1)
10	-	-	-	575,1 (m = 2)

Proses inisialisasi untuk pemulusan eksponensial linear dari Holt memerlukan dua taksiran, yaitu untuk nilai S_1 dan T_1. Nilai S_1 dapat disamakan dengan nilai aktualnya (observasi) atau rata – rata dari beberapa nilai observasi pada periode awal, sedangkan nilai T_1 dapat menggunakan taksiran kemiringan dari serial data itu atau dengan menggunakan rata – rata kenaikan dari beberapa periode, misalkan :

$$T_t = \frac{(X_2 - X_1) + (X_3 - X_2) + (X_4 - X_3)}{3}$$

Gambar 1.26 Rata – Rata tiga Periode

Tidak ada aturan baku dalam memilih nilai inisial S dan T. Namun, perlu diketahui bahwa inisial tersebut memiliki pengaruh pada nilai parakiraan periode berikutnya.

2.16.3 Metode Pemulusan Eksponensial Musiman

Sebagaimana halnya dengan persamaan pemulusan eksponensial linear yang dapat digunakan untuk memprakirakan serial data yang memiliki pola trend, bentuk persamaan lain dapat digunakan apabila pola dasar serial data berbentuk musiman. Salah satu metode prakiraan yang khusus untuk data yang berpola musiman adalah Metode Pemulusan Eksponensial Linear dan Musiman dari

Winter. Metode ini didasarkan atas tiga persamaan, yaitu masing – masing untuk unsur stasioner, trend dan musiman, yang dirumuskan sebagai berikut :

$$S_t = \alpha \left(\frac{X_t}{I_{t-L}}\right) + (1 - \alpha)(S_{t-1} + T_{t-1})$$
$$T_t = \beta(S_t - S_{t-1}) + (1 - \beta)T_{t-1}$$
$$I_t = \gamma \left(\frac{X_t}{S_t}\right) + (1 - \gamma)I_{t-L}$$
$$F_{t+m} = (S_t + T_t.m)I_{t-L+m}$$

Gambar 1.27 Persamaan Pemulusan Eksponensial Musiman

Di mana:

L = Jumlah periode dalam satu siklus musim

I = Faktor penyesuaian musim (indeks musiman)

Tabel 1.3 Prakiraan dengan Metode Pemulusan Eksponensial Linear dan Musiman dari Winter memberikan contoh perhitungan prakiraan dari suatu serial data yang memiliki unsur musiman. Dalam contoh ini, panjang musiman (L) empat triwulan atau satu tahun, sedangkan nilai parameter α = 0.6, β = 0.2 dan γ = 0.5.

Sebagaimana pada metode eksponensial yang dibahas sebelumnya, nilai S_t, T_t dan I_t yang pertama berupa nilai inisial (diasumsikan). Nilai inisial S_t dapat disamakan dengan nilai aktualnya atau berupa rata – rata dari beberapa nilai pada musim yang sama. Dalam contoh Tabel 1.3 Prakiraan dengan Metode Pemulusan Eksponensial Linear dan Musiman dari Winter, nilai inisial untuk S_4 disamakan dengan nilai aktualnya (X_4) yaitu 441.

Nilai inisial T dapat dicari dengan menggunakan rumus sebagai berikut :

$$T_L = \frac{1}{L}\left\{\frac{(X_{L+1} - X_1)}{L} + \frac{(X_{L+2} - X_2)}{L} + \cdots + \frac{(X_{L+L} - X_l)}{L}\right\}$$

Gambar 1.28 Pencarian Nilai Inisial T

Berikut contoh peramalan dengan menggunakan metode pemulusan eksponensial linear dan musiman dari Winter.

Tabel 1.3 Prakiraan dengan Metode Pemulusan Eksponensial Linear dan Musiman dari Winter

Tahun	Kuartal	Periode	Nilai Observasi X	Nilai Prakiraan			
				S (α = 0,6)	T (β = 0,2)	I (γ = 0,5)	F
2001	1	1	460			*0,96*	
	2	2	484			*1,01*	
	3	3	530			*1,11*	
	4	4	441	*441,0*	*10,25*	*0,92*	
2002	1	5	492	487,7	17,55	0,98	
	2	6	509	504,2	17,33	1,01	510,8
	3	7	588	527,3	18,48	1,11	577,4
	4	8	490	537,5	16,82	0,92	502,7
2003	1	9	533	546,5	15,25	0,98	545,9
	2	10	560	557,3	14,37	1,01	567,5
	3	11	532	569,9	14,03	1,11	635,1
	4	12	560	600,2	17,28	0,92	535,2
2004	1	13	604	616,8	17,13	0,98	605,2
	2	14	675	655,5	21,45	1,02	638,7
	3	15	701	649,7	16,00	1,09	751,4
	4	16	607	660,2	14,89	0,92	615,6
2005	1	17	708	703,6	20,61	0,99	661,3
	2	18	787	753,3	26,41	1,03	737,7
	3	19	850	777,8	26,05	1,09	853,3
	4	20	782	830,4	31,35	0,93	741,3
2006	1	21	-				855,7
	2	22	-				921,4
	3	23	-				1011,0
	4	24	-				890,7

Dengan demikian, nilai T_4 diasumsikan sebesar :

$$T_4 = \frac{1}{4}\left\{\frac{492-460}{4} + \frac{509-484}{4} + \frac{588-530}{4} + \frac{490-441}{4}\right\} = 10{,}25$$

Perhitungan nilai inisial I_t pada satu siklus musim pertama (empat periode pertama, tahun 2001) dilakukan dengan membagi setiap data pengamatan (X) dengan rata – rata data pengamatan pada siklus itu.

Rata – rata permintaan tahun 2001 :

$$X_{2001} = (460 + 484 + 530 + 441) / 4 = 478{,}75$$

Sehingga nilai inisial indeks musimannya

I_1 = 460 / 478,75 = 0,96

I_2 = 484 / 478,75 = 1,01

I_3 = 530 / 478,75 = 1,11

I_4 = 441 / 478,75 = 0,92

Setelah nilai – nilai inisial S, T dan I diperoleh, dapat dilakukan perhitungan S_t, T_t dan I_t dan prakiraan F_{t+m} dapat dicari. Nilai prakiraan dihitung berdasarkan data paling akhir. Dalam contoh Tabel 1.3 Prakiraan dengan Metode Pemulusan Eksponensial Linear dan Musiman dari Winter nilai prakiraan sampai periode ke – 12 diperoleh berdasarkan data satu periode sebelumnya (m = 1). Prakiraan untuk periode – periode selanjutnya diperoleh dengan menggunakan data periode ke – 20 yang merupakan periode terakhir yang memiliki data aktual.

Salah satu masalah yang timbul dalam penggunaan model Winter Untuk prakiraan adalah penentual nilai – nilai α, β, γ. Pendekatan yang biasa dipakai ialah dengan uji coba sampai diperoleh nilai – nilai parameter yang meminimalkan kesalahan prakiraan (MAD atau MSE).

$$0 \leq \alpha, \beta, \gamma \leq 1$$

2.16.4 Nilai Ketepatan Prediksi

Ketepatan ramalan adalah suatu hal yang penting untuk peramalan, yaitu bagaimana mengukur kesesuaian antara data yang sudah ada dengan data peramalan. Ada beberapa perhitungan yang biasa digunakan untuk menghitung kesalahan peramalan total. Tiga dari perhitungan yang paling terkenal adalah deviasi mutlak rata – rata (*Mean Absolute Deviation - MAD*), kesalahan kuadrat rata – rata (*Mean Squared Error - MSE*), dan kesalahan persen mutlak rata – rata (*Mean Absolute Percentage Error - MAPE*). [35]

2.16.4.1 *Mean Absolute Deviation* (MAD)

Mean Absolute Deviation (MAD) adalah metode untuk mengevaluasi metode peramalan menggunakan jumlah dari kesalahan – kesalahan yang absolut. *Mean Absolute Deviation* (MAD) mengukur ketepatan ramalan dengan

menghitung rata – rata kesalahan peramalan (nilai absolut masing masing kesalahan). MAD berguna ketika mengukur kesalahan ramalan dalam unit yang sama sebagai deret asli. MAD merupakan ukuran pertama kesalahan peramalan keseluruhan untuk sebuah model. Rumus untuk menghitung MAD adalah sebagai berikut.

$$MAD = \frac{\sum_{t=1}^{n} |X_t - F_t|}{n}$$

Dimana :

X_t = Data aktual pada periode t

F_t = Nilai peramalan pada periode t

n = Jumlah data

2.16.5 *Mean Square Error* (MSE)

Mean Square Error (MSE) adalah metode lain untuk mengevaluasi metode peramalan. Selisih dari masing masing data peramalan dan data aktual dikuadratkan. Pendekatan ini mengatur kesalahan peramalan yang besar karena kesalahan – kesalahan tersebut dikuadratkan. MSE merupakan cara kedua untuk mengukur kesalahan peramalan keseluruhan. MSE merupakan rata – rata selisih kuadrat antara nilai yang diramalkan dan nilai yang diamati. Kekurangan penggunaan MSE adalah bahwa MSE cenderung menonjolkan deviasi yang besar karena adanya pengkuadratan. Rumus untuk menghitung MSE adalah sebagai berikut.

$$MSE = \sum_{t=1}^{n} \frac{(X_t - F_t)^2}{n}$$

2.16.6 *Mean Absolute Percentage Error* (MAPE)

Mean Absolute Percentage Error (MAPE) dihitung dengan menggunakan kesalahan absolute pada tiap periode dibagi dengan nilai observasi yang nyata untuk periode itu. Kemudian, menghitung rata – rata kesalahan persentase absolut tersebut. MAPE merupakan pengukuran kesalahan yang menghitung ukuran persentase penyimpangan antara data aktual dengan data peramalan. Nilai MAPE dapat dihitung dengan persamaan berikut.

$$MAPE = \left(\frac{100\%}{n}\right)\sum_{t=1}^{n}\frac{|X_t - F_t|}{X_t}$$

Kemampuan peramalan dinilai sangat baik jika memiliki nilai MAPE kurang dari 10% dan mempunyai kemampuan peramalan yang baik jika nilai MAPE kurang dari 20%.

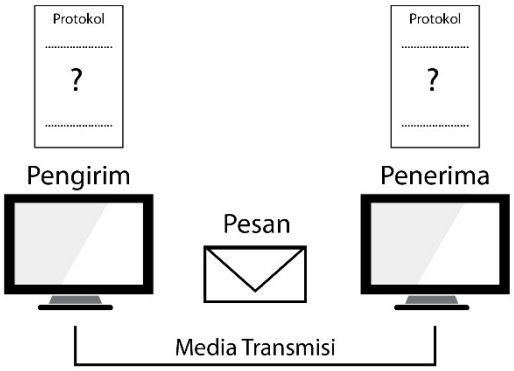
2.17 Komunikasi Data

Komunikasi data adalah hubungan antara dua atau lebih peralatan "data *processing*" (contoh : Komputer) melalui media transmisi untuk melakukan pertukaran informasi. [36] Unsur pokok dalam komunikasi data, adalah :

1. Sumber atau pengirim
2. Media Transmisi
3. Tujuan atau Penerima
4. Informasi atau data yang dikirim

Pendekatan yang lebih realistis dapat mengambil contoh yakni saat seseorang berkomunikasi dengan orang lain, maka akan terjadi saling tukar menukar informasi satu sama lainnya. Tukar menukar informasi ini dapat terjadi secara lokal atau secara remote. Antar dua individu yang berbeda, komunikasi kasi secara lokal terjadi antara *face to face*, sedangkat secara remote dapat terjadi jika terdapat perbedaan jarak, waktu dan tempat.

Komunikasi data adalah proses pertukaran data antara dua perangkat (*device*) komunikasi melalui media transmisi seperti kabel dan wireless. [37]



Gambar 1.29 Komponen Sistem Komunikasi Data

2.17.1 Komponen – komponen Dasar Komunikasi Data

Pada Gambar 1.29 Komponen Sistem Komunikasi Data, terlihat terdapat beberapa komponen dasar pembentuk komunikasi data. Komponen – komponen dasar tersebut meliputi :

1. Pesan (*Message*). Merupakan informasi (data) yang akan ditransmisikan. Pesan tersebut dapat berupa teks, angka, gambar, suara, atau video serta kombinasi dari keseluruhan pesan tersebut.
2. *Sender*. Merupakan perangkat yang berfungsi untuk mengirimkan informasi. Sender dapat berupa komputer, workstation, handset pesawat telepon ataupun kamera video.
3. *Receiver.* Merupakan perangkat yang berfungsi untuk menerima informasi atau pesan dari *sender*. Dapat berupa komputer, workstation, handset pesawat telepon atau TV.
4. Media transmisi. Merupakan media transmisi tempat data atau informasi dilewatkan dari *sender* ke *receiver*. Dapat berupa kabel, fiber optik, laser, gelombang radio.
5. Protokol. Protokol merupakan seperangkat aturan yang digunakan untuk saling berkomunikasi. Dapat dianalogikan seperti bahasa yang digunakan oleh manusia.

2.18 Wifi

Wifi atau Wi-Fi, kependekan dari *Wireless Fidelity,* adalah sekumpulan standar yang digunakan untuk Jaringan Lokal Nirkabel (Wireless Local Area Network - WLAN). Didasari pada spesifikasi IEEE 802.11, yang kemudian berkembang dengan beberapa spesifikasi, antara lain 802.11a, 802.11b, 802.11g, dan 802.11n. Dengan Wifi, kita dapat membangun atau terhubung dengan jaringan area lokal (LAN). Selanjutnya, jika suatu LAN terhubung dengan segmen LAN lain, kita akan juga bisa terhubung dengan perangkat pada segmen LAN tersebut. Demikian juga, jika suatu LAN terhubung dengan jaringan internet, perangkat atau komputer dalam LAN akan bisa mengakses internet. [38]

2.18.1 Frekuensi Wifi

Di Indonesia, standar Wifi yang paling banyak dipakai adalah 802.11b, dan 802.11g yang bekerja pada frekuensi 2.400 MHz sampai dengan 2.483,50 MHz dan terbagi dalam 11 channel (masing – masing 5 MHz), yaitu :

1. Channel 1 – 2.412 MHz
2. Channel 2 – 2.417 MHz
3. Channel 3 – 2.422 MHz
4. Channel 4 – 2.427 MHz
5. Channel 5 – 2.432 MHz
6. Channel 6 – 2.437 MHz
7. Channel 7 – 2.442 MHz
8. Channel 8 – 2.447 MHz
9. Channel 9 – 2.452 MHz
10. Channel 10 – 2.457 MHz
11. Channel 11 – 2.462 MHz

Setiap pemancar Wifi akan menggunakan salah satu channel (frekuensi) dan Wifi lain yang akan mengoneksinya harus menggunakan channel yang sama. Pembagian channel ini juga memperkecil kemungkinan interferensi pada beberapa Wifi yang berlainan jaringan (jika menggunakan channel yang berbeda).

2.18.2 Mode Koneksi Wifi

Koneksi dengan Wifi bisa dilakukan dalam mode Ad – Hoc atau mode Infrastructure.

Mode Ad – Hoc adalah koneksi antara dua komputer, di mana saru komputer berfungsi sebagai server dan komputer lainnya menjadi client. Koneksi semacan ini sering disebut sebagai koneksi peer – to – peer.

Mode Infrastructure adalah koneksi antara dua komputer atau lebih, dengan Access Point (AP) sebagai pengatur lalu lintasnya. Access Point adalah suatu perangkat yang dapat memancarkan sinyal Wifi dalam jangkauan tertentu (sering disebut sebagai hotspot). Melalui sinyal Wifi tersebut, beberapa client bisa terkoneksi ke jaringan dan AP – lah yang akan mengatur lalu lintasnya.

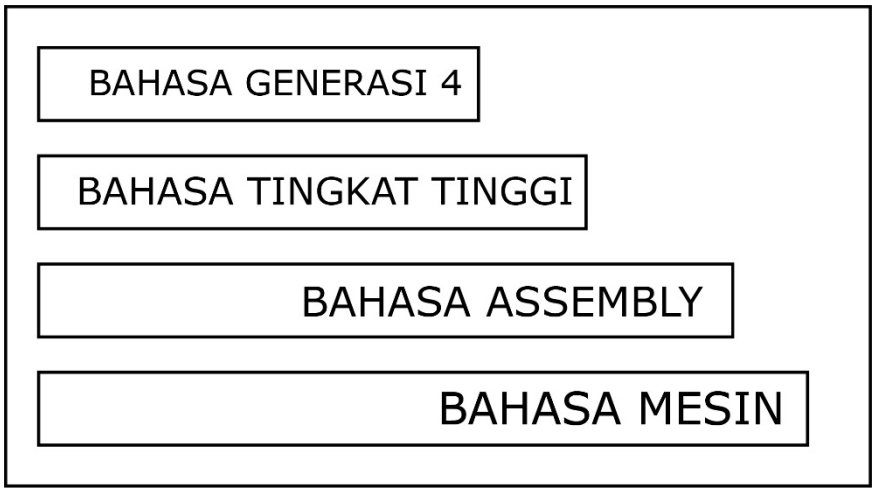
2.19 Bahasa Pemrograman (*programming language*)

Merupakan notasi untuk memberikan instruksi secara tepat pada komputer. Berbeda dengan bahasa manusia yang merupakan bahasa alamial (*natural language*), sintaks atau struktur bahasa pemrograman (komputer) ditentukan secara kaku, sehingga bahasa pemrograman juga disebut sebagai bahasa formal (*formal language*). [39]

Jadi, dalam bahasa pemrograman yang digunakan sebagai alat komunikasi untuk memberikan perintah kepada komputer terdapat aturan – aturan main tertentu yang telah ditetapkan secara ketat.

Generasi bahasa pemrograman :

1. Generasi I : *Machine Language*
2. Generasi II : *Assembly Language* : Assembler.
3. Generasi III : *high – level programming language* : C, PASCAL, Java, Python, HTML.
4. Generasi IV : 4 GL (*fourth – generation language*) : SQL



Sumber Gambar : E-Book Dasar – Dasar Pemrograman Pascal **[39]**

Gambar 1.30 Tingkatan Bahasa Pemrograman

2.19.1 Bahasa Tingkat Rendah (*low – level language*)

Merupakan bahasa *assembly* atau bahasa mesin. Bahasa ini dikatakan tergolong dalam bahasa tingkat rendah dikarenakan lebih dekat ke mesin

(*hardware*), bahasa yang tergolong dalam bahasa tingkat rendah adalah Bahasa Mesin (*Machine Language*) dan Bahasa Rakitan (*Assembly Language*).

2.19.1.1 Bahasa Mesin

Bahasa mesin merupakan representasi tertulis *machine code* (kode mesin), yaitu kode operasi suatu mesin tertentu. Bahasa ini bersifat khusus untuk mesin tertentu dan "dimengerti" langsung oleh mesin, sehingga pelaksanaan proses sangat cepat. Bahasa mesin kelompok komputer tertentu berlainan dengan bahasa mesin kelompok komputer yang lain.

Abstraksi bahasa ini adalah kumpulan kombinasi kode biner "0" dan "1" yang sangat tidak alamiah bagi kebanyakan orang, kecuali untuk para ahli pembuat mesin komputer. Karena tidak alamiah bagi kebanyakan orang, bahasa mesin juga disebut bahasa tingkat rendah

2.19.1.2 Bahasa Assembly

Bahasa rakitan (*assembly language*) merupakan notasi untuk menyajikan bahasa mesin yang lebih mudah dibaca dan dipahami oleh manusia. Bahasa ini sudah menggunakan simbol alphabet yang bermakna (*mnemonic*). Contoh "MOV AX 1111", dipindahkan ke register AX nilai 1111.

Proses data oleh komputer berdasarkan perintah bahasa rakitan adalah cepat. Meski demikian masih merepotkan bahkan bagi kebanyakan pemrogram, karena masih harus mengingat – ingat tempat penyimpanan data. Bahasa rakitan juga bersifat khusus untuk mesin tertentu seperti Assembler.

2.19.1.3 Bahasa Tingkat Tinggi (*high – level language*)

Adalah bahasa pemrograman yang dekat dengan bahasa manusia, kelebihan utama dari bahasa ini adalah mudah untuk dibaca, tulis, maupun diperbaharui, sebelum bisa dijalankan program harus terlebih dahulu di – compile atau diterjemahkan. Contoh bahasa yang tergolong dalam bahasa tingkat tinggi adalah Ada, Algol, Basic, Cobol, C, C++, Fortran, Pascal, Java.

Pada generasi bahasa pemrograman terakhir sekarang ini, kedua cara interpretasi dan kompilasi digabungkan dalam satu lingkungan pengembangan terpadu (IDE = *Integrated Development Environment*).

Cara interpretasi memudahkan dalam pembuatan program secara interaktif dan cara kompilasi menjadikan eksekusi program lebih cepat. Pertama program dikembangkan interaktif, kemudian setelah tidak ada kesalahan keseluruhan program dikompilasi. Contoh bahasa program seperti ini adalah Visual Basic yang berbasis Basic dan Delphi yang berbasis Pascal.

2.19.1.4 Fourth – Generation Language (4GL)

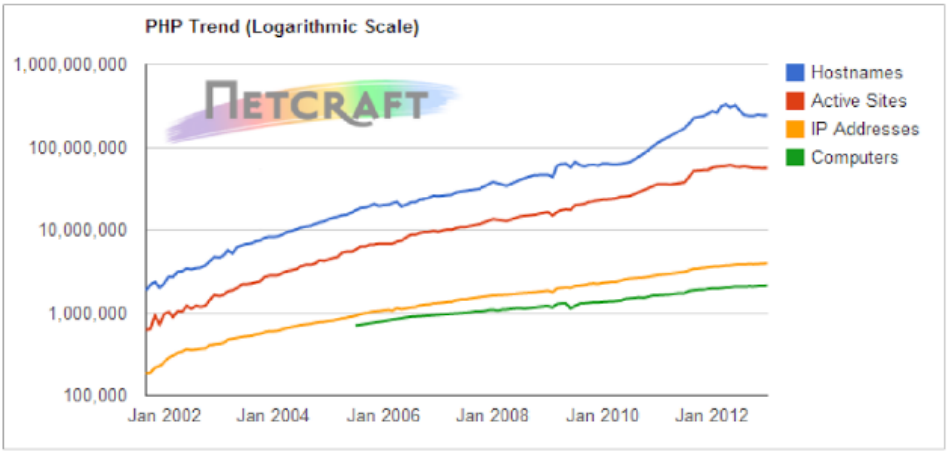
Lebih dekat ke bahasa manusia dibandingkan dengan *high – level programming languages*. Biasanya dipakai untuk mengakses database. Contoh pemrograman tingkat ini adalah *Structured Query Language* (SQL).

2.20 PHP

PHP merupakan salah satu bahasa pemrograman berbasis web yang ditulis oleh dan untuk pengembang web. PHP pertama kali dikembangkan oleh Rasmus Lerdorf, seorang pengembang *software* dan anggota tim Apache, dan dirilis pada akhir tahun 1994. PHP dikembangkan dengan tujuan awal hanya untuk mencatat pengunjung pada website pribadi Rasmus Lerdorf. Pada rilis keduanya, ditambahkan *Form Interpreter*, sebuah ools untuk melakukan penerjemahan perintah SQL. Rilis kedua disebut dengan PHP/FI. Sejak itu, PHP mulai diterima sebagai sebuah bahasa pemrograman baru yang sangat diminati. Terbukti pada pertengahan tahun 1997, tercatat sekitar 50.000 situs di seluruh dunia telah menggunakan PHP. [40]

Dengan bertambah banyaknya pengguna PHP diseluruh dunia, maka PHP tidak memungkinkan lagi untuk dikelola oleh satu orang saja. Sehingga dibentuk sebuah tim pengembang yang mengorganisisr kontributor dari seluruh dunia dengan model pengembangan proyek open source “benevolent junta”. Tim tersebut dipimpin oleh dua orang programmer, Zeev Suraski dan Andi Gutmans, keduanya lalu mendirikan sebuah perusahaan PHP dengan nama Zend (akronim dari nama Zeev Suraski dan Andi Gutsman). Selanjutnya Zend merilis versi PHP3 dan PHP4.

Tahun 1998 terjadi peningkatan penggunaan PHP yang sangat besar, bersamaan dengan naiknya popularitas penggunaan teknologi open source. Pada bulan oktober 1998, PHP sudah digunakan oleh sekitar 100.000 domain di internet. Dan setahun setelahnya, sudah mencapai 1 juta domain. Perkembangan selanjutnya sangatlah luar biasa. Pada tahun 2004 saja tercatat sekitar 15 juta server web terpasang PHP di dalamnya. Berdasarkan survey yang dilakukan Netcraft, pada bulan Januari 2013 situs yang menggunakan PHP sudah mencapai 244 juta situs.



Sumber Gambar : E-Book Pemrograman Web dengan PHP dan MySQL **[40]**

Gambar 1.31 Tren Penggunaan PHP oleh Netcraft

Seiring dengan waktu, PHP terus dikembangkan baik dari sisi fitur, keamanan maupun performanya. Versi terakhir PHP adalah PHP 5.6.14 (Oktober 2015). Sejak PHP5, terjadi penambahan dan peningkatan fitur yang cukup signifikan. PHP5 dibangun dengan konsep berorientasi obyek (OOP) secara penuh seperti halnya bahasa pemrograman seperti Java dan C++. Versi sebelumnya memang PHP belum sepenuhnya mendukung konsep OOP. Mulai PHP5, seluruh fungsi bawaan PHP ditulis ulang menggunakan konsep OOP. Walaupun pada masa transisi PHP masih mendukung gaya klasik yaitu prosedural. Dan sedikit demi sedikit gaya penulisan prosedural akan dihilangkan dari PHP.

Beberapa keunggulan PHP adalah sebagai berikut :

1. Gratis. PHP dapat diunduh dan dipergunakan secara gratis. PHP dikembangkan oleh komunitas open source dan akan selalu didistribusikan secara gratis. Dari, oleh dan untuk pengembang web.

2. PHP berlisensi GNU General Public License (GPL). Hal tersebut menjadi jaminan bahwa semua versi PHP akan selalu didistribusikan secara gratis. Walaupun demikian, untuk setiap program dan aplikasi yang dihasilkan menggunakan PHP, pengembang dapat menentukan lisensinya sendiri (tidak harus berlisensi terbuka). Selain itu, dalam proses distribusinya, tetap diperkenankan menarik biaya dalam jumlah yang wajar, misalnya sebagai biaya *packaging* atau burning ke CD ROM.
3. Performa Handal. PHP sangatlah efisien, dengan sebuah spesifikasi server yang tidak mahal, dapat melayani jutaan akses per hari.
4. Dukungan Basisdata. PHP mendukung hampir semua perangkat basisdata, mulai dari MySQL, Oracle, PostgreSQL, Informix, Interbase, Sybase, MariaDB hingga SQLite. Bahkan beberapa diantaranya sudah terhubung dengan PHP secara native (*default*).
5. Pustaka Bawaan. PHP dibangun secara khusus untuk aplikasi berbasis web. Oleh karena itu, disediakan banyak sekali pustaka (*library*) bawaan berkaitan dengan web yang dapat langsung digunakan. Sebagai contoh, PHP memiliki pustaka untuk menampilkan PDF on-the-fly (langsung di browser), membuat grafik on-the-fly, mengirim email, mengelola file, kompresi, enkripsi, penanganan session dan sebagainya.
6. Cross Platform. PHP dapat dijalankan dengan baik pada hampir semua Sistem Operasi. PHP dapat berjalan di Linux, Unix, Windows, Mac OS, FreeBSD, Sun Solaris dan bahkan saat ini dapat berjalan di sistem operasi Android melalui proyek DroidPHP.
7. Mudah Dipelajari. Perintah – perintah PHP sangat mudah dipelajari karena sebagian besar perintah PHP diadopsi dari bahasa pemrograman populer seperti Bahasa C/C++. Java dan Perl.

```php
<?php
$t = date("H");
if ($t < "20") {
    echo "Have a good day!";
}
?>
```

Gambar 1.32 Contoh Syntax PHP

2.21 HTML, CSS dan Javascript

HTML merupakan singkatan dari *Hypertext Markup Language.* HTML dikembangkan pertama kali oleh tim Berners – Lee bersamaan dengan protokol HTTP (*Hypertext Transfer Protocol*) pada tahun 1989. Tujuan utama pengembangan HTML adalah untuk menghubungkan satu halaman web dengan halaman web lainnya. Tentunya pada awal perkembangannya, halaman web hanya berupa teks, tidak seperti sekarang. [40]

Pada dasarnya, setiap halaman web ditulis dalam bentuk HTML. HTML merupakan bahasa pemrograman web yang memberitahukan peramban web (web browser) bagaimana mentusun dan menyajikan konten di halaman web. Dengan kata lain, HTML adalah pondasi web. HTML disusun dengan bahasa yang sederhana, sehingga sangat mudah diimplementasikan. Saat ini, HTML dapat menampilkan obyek – obyek seperti teks, tabel, tautan, gambar, audio dan video.

HTML sendiri berkembang dari waktu ke waktu, untuk menetapkan standar perintah HTML dan mengembangkannya, didirikan sebuah konsorsium W3C (World Wide Web Consortium) yang diprakarsai oleh tim Berners – Lee. Pada perkembangannya, W3C tidak hanya menetapkan standar HTML, namun beberapa teknologi lainnya seperti XML, XHTML dan SOAP. Beberapa standar versi HTML yang pernah dikeluarkan oleh W3C antara lain HTML 2.0 pada tahun 1995, HTML 3.2 pada Januari 1997, HTML 4.0 pada Desember 1997 dan direvisi menjadi HTML 4.01 pada bulan Desember 1999. Pada bulan Mei 2000, W3C mengumumkan standar baru HTML dengan nama XHTML yang sebenarnya berupa standar HTML 4.01 dengan beberapa penyesuaian. Dan saat ini, standar HTML yang terakhir

dirilis oleh W3C adalah HTML5. Banya fitur (perintah) baru yang dikenalkan pada HTML5 seperti penanganan audio, video dan penyimpanan lokal.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Gambar 1.33 Contoh Syntax HTML

HTML merupakan bahasa dasar web yang berfungsi untuk menampilkan berbagai komponen web. Sementara itu, untuk mempercantik tampilan web, dikembangkanlah CSS atau *Cascading Style Sheet.* CSS pertama kali diusulkan oleh Hakon Wium Lie pda tahun 1994 dan selanjutnya distandarisasi oleh W3c. CSS memberikan cara yang mudah dan efisien bagi pemrogram untuk menentukan tata letak halaman web dan mempercantik halamandengan elemen desain seperti warna, sudut bulat, gradien, dan animasi.

Seperti halnya HTML, versi CSS juga berkembang. Versi pertama yang ditetapkan oleh W3C adalah CSS 1 (dibaca CSS level 1) pada bulan Desember 1996. Selanjutnya berkembang menjadi CSS 2 (ditetapkan bulan Mei 1998). CSS 2.1 (tahun 2011) dan CSS 3 yang diusulkan pada bulan 2012. Versi yang sedang populer saat ini adalah CSS 3 yang memiliki banyak kelebihan dibanding versi sebelumnya seperti penambahan fitur media – queries dan penanganan font web.

```css
body {
    background-color: lightblue;
}

h1 {
    color: white;
    text-align: center;
}

p {
    font-family: verdana;
    font-size: 20px;
}
```

Gambar 1.34 Contoh Syntax CSS

Selain HTML dan CSS, sebuah aplikasi berbasis web tidak dapat dilepaskan dari teknologi Javascript. Pertama kali Javascript dikembangkan oleh Netscape dengan nama awal LiveScript. Fungsi utama dari Javascript adalah untuk menambah fungsionalitas dan kenyamanan halaman web. Javascript lebih fokus pada proses pengolahan data di sisi client serta menyajikan komponen web yang lebih interaktif. Javascript makin populer sejak kemunculan konsep AJAX (Asynchronous Javascript and XML) yang memungkinkan interaksi antara client dan server lebih elegan dan fleksibel.

```javascript
document.getElementById("demo").innerHTML = "Hello JavaScript";
```

Gambar 1.35 Contoh Syntax Javascript

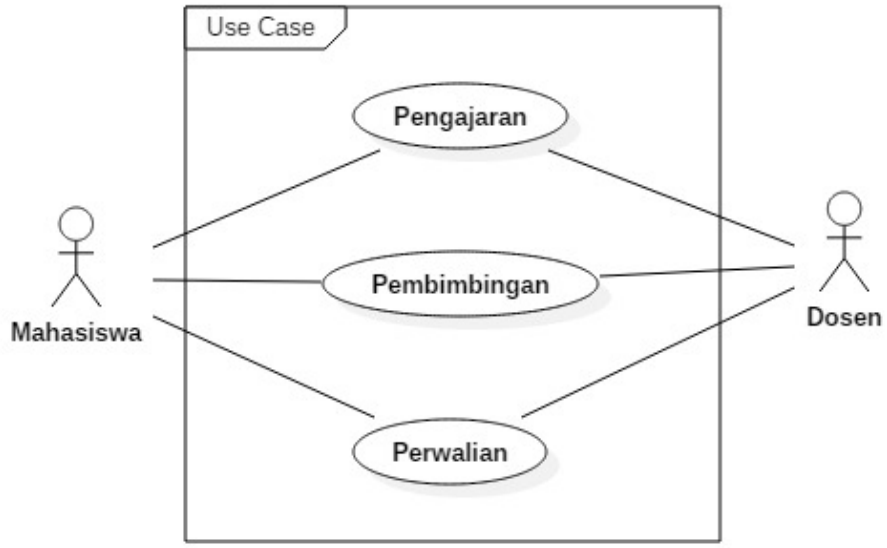
Peran HTML, CSS dan Javascript dalam pembuatan web dapat dianalogikan sebagaimana pembuatan baju kemeja. Bahan dasar dari pembuatan baju adalah bahan atau kain. Baju tidak akan bisa dibuat tanpa keberadaan kain. Demikian juga halaman web, tidak akan terbentuk tanpa keberadaan perintah HTML. Untuk melengkapi baju kemeja, biasanya ditambahkan satu atau beberapa saku yang berfungsi untuk menaruh sesuatu. Fungsionalitas baju bertambah tidak sekedar melindungi tubuh, namun dapat untuk menaruh dompet, handphone, catatan, dan sebagainya. Kantong saku inilah Javascript dalam pembuatan web. Sedangkan CSS pada perumpamaan ini adalah tata letak, pengaturan warna, serta hiasan – hiasan tertentu seperti renda, bros dan aksesoris lainnya.

2.22 UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah 'bahasa' pemodelan untuk sistem atau perangkat lunak yang berparadigma 'berorientasi objek'. Pemodelan (*modelling*) sesungguhnya digunakan untuk penyederhanaan permasalahan – permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Sebagai contoh, alih – alih membawa planet bumi ke meja kita, kita mungkin akan membawa peta – peta tentang bagian bumi tertentu ke meja kita untuk kita pelajari lebih jauh. Adapun tujuan pemodelan (dalam kerangka pengembangan sistem/perangkat lunak) adalah sebagai sarana analisis, pemahaman, visualisasi, dan komunikasi antar anggota tim pengembang, serta sebagai sarana dokumentasi perilaku prangkat lunak secara seksama serta bermanfaat untuk melakukan pengujian terhadap perangkat yang telah selesai dikembangkan. [41]

2.22.1 *Use Case* Diagram

Use Case digunakan untuk memodelkan fungsionalitas – fungsionalitas sistem/perangkat lunak dilihat dari pengguna yang ada di luar sistem (yang sering dinamakan sebagai *actor*). *Use Case* pada dasarnya merupakan unit fungsionalitas koheren yang diekspresikan sebagai transaksi – transaksi yang terjadi antara *actor* dan sistem. Kegunaan dari *use case* adalah untuk mendaftarkan setiap *actor* dan *use case* serta memperlihatkan *actor* mana yang berpartisipasi dalam masing – masing *use case.* Gambar 1.36 merupakan contoh penggambaran diagram *use case*. Ikon berbentuk orang adalah *actor* sementara elips – elips yang ada menggambarkan fungsionalitas fungsionalitas sistem (*use case*).



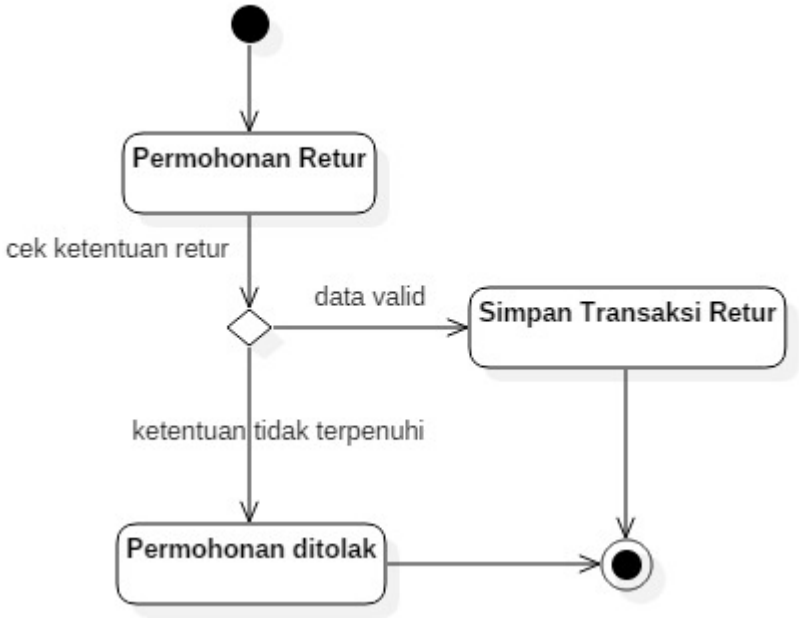
Gambar 1.36 Use Case Diagram

Sebuah *actor* merupakan idealisasi dari orang yang ada di luar sistem, idealisasi proses – proses yang berinteraksi dengan sistem, atau idealisasi sesuatu yang berinteraksi dengan sistem, subsistem, atau kelas, pada sistem/perangkat lunak yang sedang kita kembangkan. Sebuah *actor* pada dasarnya menggambarkan interaksi pengguna – pengguna sistem dengan sistem/perangkat lunak yang sedang kita kembangkan. Masing – masing *actor* bisa saja berpartisipasi dalam satu atau lebih *use case*. Suatu *actor* mungkin saja berupa manusia (pengguna sistem), sistem komputer yang lain, atau sejumlah proses yang dapat dieksekusi dan berinteraksi dengan sistem yang sedang kita kembangkan.

2.22.2 *Activity* Diagram

Activity Diagram menggambarkan berbagai aktivitas dalam sistem yang sedang dirancang, mulai dari titik awal, melalui kondisi (*decision*) yang mungkin terjadi, kemudian sampai pada titik akhir. Diagram ini juga mampu menggambarkan proses paralel yang munkin terjadi pada beberapa eksekusi. Diagram ini tidak menggambarkan perilaku/proses internal sebuah sistem maupun interaksi antar subsistem, tetapi lebih menggambarkan proses – proses dan jalur – jalur aktivitas secara umum (global).

Diagram ini dilambangkan dengan segi empat dengan sudut membulat (lebih bulat dari *statechart*) untuk menggambarkan aktivitas, dan memiliki (*decision*) yang digunakan untuk mengambil keputusan/percabangan. Untuk mengilustrasikan proses – proses paralel digunakan titik sinkronisasi yang dapat berupa garis vertikal atau horizontal. Contoh penggambaran *activity* diagram pada Gambar 1.37.



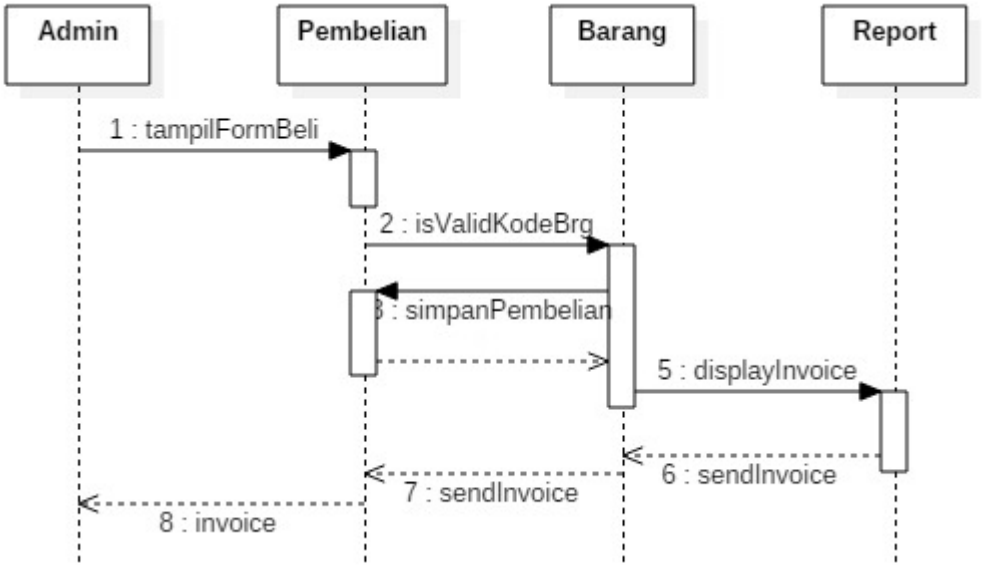
Gambar 1.37 *Activity* Diagram

Permohonan retur dimulai dengan melakukan pengecekan pada berkas retur dan ketentuan yang berlaku. Jika data valid, maka aktivitas berikutnya adalah menyimpan transaksi retur dan selesai ke titik akhir. Namun, apabila data retur tidak memenuhi ketentuan yang berlaku, maka permohonan retur ditolak dan aktivitas berikutnya menuju titik akhir (selesai).

2.22.3 *Sequence* Diagram

Diagram ini menggambarkan interaksi antarobjek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek – objek yang terkait). Biasa digunakan untuk menggambarkan skenario atau rangkaian langkah – langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu. Diawali

dari sebuah aktivitas tertentu, kemudian berproses mengikuti urutan tertentu, yang bisa terlihat melalui *message* antarobjeknya.



Gambar 1.38 *Sequence* Diagram