BAB II

LANDASAN TEORI

2.1 Pemograman Android

2.1.1 Pemograman

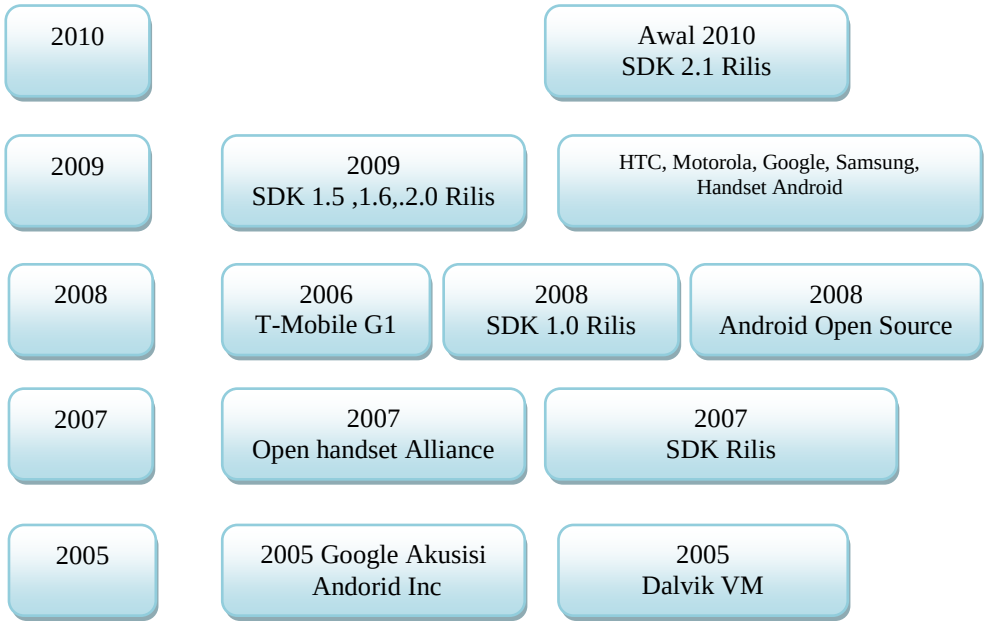
Pemrograman adalah proses menulis, menguji dan memperbaiki *(debug)*, dan memelihara kode yang membangun sebuah program komputer. Kode ini ditulis dalam berbagai bahasa pemrograman. Tujuan dari pemrograman adalah untuk memuat suatu program yang dapat melakukan suatu perhitungan atau pekerjaan sesui dengan keinginan si pemrogram, diperlukan keterampilan dalam algoritma, logika, bahasa pemrograman, dan banyak kasus, pengetahuan lain seperti matematika.

Pemrograman adalah sebuah seni dalam menggunakan satu atau lebih algoritma yang saling berhubungan dengan menggunakan sebuah bahasa pemrograman tertentu sehingga menjadi sebuah program komputer. Bahasa pemrograman yang berbeda mendukung gaya pemrograman yang berbeda pula. Gaya pemrograman ini biasa disebut paradigm pemrograman.

(*Phyton Programing : An Introduction To Computer Science, 2002 : 1)*

2.1.2 Android

Pada tahun 2005 Google mengakusisi Andoid Inc yang pada saat itu di motori oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Yang kemudian pada tahun itu juga memulai membangun platform Android secara intensif



Gambar 2.1. Android Timeline

(Sumber : Mulyadi,ST. 2010: 2)

Kemudian pada tanggal 12 November 2007 Google bersama Open Handset Alliance (OHA) yaitu konsorsium mobile terbuka, merilis Google Android SDK, setelah mengumumkan seminggu sebelumnya, dan sambutanya sangat luar biasa, hampir semua media berita tentang IT dan *Programming* memberitakan tentang dirilisnya Android SDK (Software Development Kit) ini.

Ini dikarenakan ide dari Platform Android ini sangat menarik untuk *developer* dan *programmer* di seluruh penjuru dunia, disisi lain, produsen pembuat handset ponsel juga segera berlomba lomba membuat handset dengan platform android ini.

Google bersama dengan OHA merilis paket software SDK yang lengkap untuk mengembangkan aplikasi pada perangkay mobile yaitu : Sistem Operasi *Middleware* dan Aplikasi utama untuk perangkat mobile, sebagai *Programmer* atau *Developer* kita bias melakukan segalanya mulai dari membuat aplikasi pengiriman SMS hanya dengan 2 baris kode, hingga mengganti event pada home screen perangkat android. Selain itu bahkan dengan mudah kita bisa membuat dan mengkostumisasi system Operasinya,

Semua aplikasi yang dibuat untuk android akan memiliki akses yang setara dalam mengakses seluruh kemampuan handset tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dengan kata lain melalui platform android ini, programmer atau developer secara penuh akan bisa mengkostumisasi perangkat android.

Android Built in pada Linux Kernel (Open Linux Kernel) dengan sebuah mesin virtual yang telah didesain untuk mengoptimalkan penggunaan sumberdaya memori dan hardware pada lingkungan perangkat mobile *(mobile Environment).* Dalvik adalah nama dari Android Virtual Mesin yang merupakan interpreter *(Interpreter-only)* visual mesin yang akan mengeksekusi file kedalam format *Dalvik Executeble* (*.dex) sebuah format yang telah dirancang untuk ruang

penyimpanan yang efisien dan eksekusi memori yang terpetakan *(memory-mappable execution)*

Dalvik Virtual Mesin (DalvikVM) berbasis register *(register based)* dan dapat mengeksekusi kelas *(class)* yang telah terkompilasi pada compiler bahasa java kemudian ditransformasikan ke dalam native format dengan menggunakan Tool DX yang telah terintegrasi. Kita mungkin telah mengenal JavaV*M (Java Virtual Mesin)* yang saat ini bisa kita temukan pada setiap computer desktop. Berbeda dengan DalvikVM, javaVM berbasis stack *(Stack-based Virtual Machines).*

Singkatnya android merupakan subset perangkat lunak untuk perangkat mobile yang meliputi system operasi, *middleware* dan aplikasi inti yang di release oleh Google. Sedangkan android SDK (*Software Development Kit*) menyediakan Tools dan API yang diperlukan untuk mengembangkan aplikasi pada platform android dengan menggunakan bahasa pemograman java.

(*Mulyadi,ST. 2010 : 2)*

Adapun android memeliki beberapa versi dalam perkembanganya yaitu:

1. Android Versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan *email*.

2. Android Versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan *Bluetooth* A2DP, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3. Android Versi 1.6 (*Donut*)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet* VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, *camcorder* dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan *Text-to-speech engine*, kemampuan *dial* kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel), pengadaan resolusi VWGA.

4. Android Versi 2.0/2.1 (*Éclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan *Google Maps* 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth* 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi *mobile* terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset* Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, *Backgrounds*, dan *WeatherBug*. Sistem operasi Android dalam situs internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh *MySpace* dan *Facebook*.

5. Android Versi 2.2 (*Froyo: Frozen Yoghurt*)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe *Flash* 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 *JavaScript engine* yang dipakai Google Chrome yang mempercepat

kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam *SD Card*, kemampuan WiFi *Hotspot portabel*, dan kemampuan *auto update* dalam aplikasi Android *Market*.

6. Android Versi 2.3 (*Gingerbread*)

Pada 6 Desember 2010, Android versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

7. Android Versi 3.0 (*Honeycomb*)

Android *Honeycomb* dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet pertama yang dibuat dengan menjalankan *Honeycomb* adalah Motorola Xoom.

8. Android Versi 4.0 (*Ice Cream Sandwich*)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah,

2.1.3 Arsitektur Aplikasi Berbasis Android

Akhmad Dharma Khasman (2013 :7) dalam bukunya dijelaskan Google mengibaratkan android merupakan sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi fungsi spesifik dari system operasi. Berikut ini susunan dari lapisan lapisan tersebut jika dilihat dari lapisan teratas.

1. Linux Kernel

Tumpukan paling bawah dari arsitektur android adalah kernel. Googel menggunakan kernel linux versi 2.6 untuk membangun system android yang mencangkup memory management, security setting, power management, dan beberapa driver hardware kernel berperan sebagai abstraction layer antara hardware dan keseluruhan software sebagai contoh, HTC GI dilengkapi dengan kamera. Kernel android terdapat driver kamera yang memungkinkan pengguna mengirimkan perintah ke hardware kamera.

2. Android Runtime

Lapisan setela kernel adalah android runtime yang berisi CORE Libraries dan Dalvik Virtual Machine Core Libraries mencangkup serangkaian inti library java, artinya android menyertakan saru set library-library dasar yang menyediakan sebagian besar fungsi fungsi yang ada pada library dasar pemograman java.

Dalvik adalah *Java Virtual Machine* yang member kekuatan pada system android, dalvik ini dioptimalkan untuk telepon seluler. Setiap aplikasi yang berjalan pada android berjalan pada posesnya sendiri dengan instance dari dalvik virtual mechine. Dalvik telah dibuat sehingga sebuah piranti yang memakainya dapat berjalan multi virtual machine dengan efisien. Dalvik VM dapat mengeksekusi file dengan format *Dalvik Executable* (.dex) yang telah dioptimasi untuk menggunakan minimal memory footprint. Virtual machine ini register-based dan menjalankan class-class yang di-compile menggunakan compiler java yang kemudian ditransformasikan menjadi format.dex menggunakan "dx" tool yang telah disertakan. *Dalvik Virtual Machine* menggunakan kernel linux untuk menjalankan fungsi fungsi seperti threading dan low-level memory management.

3. Libraries

Bertempat dilevel yang sama dengan Android *Runtime* adalah Libraries. Android menyertakan satu set library dalam bahasa C++/C yang digunakan oleh berbagai komponen yang ada pada system andoid. Kemampuan ini dapat diakses oleh programmer melewati android application framework. Sebagai contoh android mendukung pemutaran format audio, video dan gambar.

Berikut beberapa core library tersebut :

a. System C library

diturunkan dari implementasi standar C system library (libc) milik BSD. Dioptimasi untuk piranti embedded berbasis linux.

b. Media Libraries

Berdasarkan packetVideo's OpenCORE, library ini mendukung playback dan recording dari berbagai format audio dan vide popular, meliputi MPEG4, H264, MP3,AAC,AMR,JPG, dan PNG.

c. Surface manager

Mengatur akses pada display dan lapisan composites 2D dan 3D graphic daei berbagai aplikasi

d. LibWebCore

Web Browser engine modern yang mensupport android browser maupun embedded web view

e. SGL

Underlying 2D graphic engine

f. 3D libraries

Implementasi berdasarkan OpenGL ES 1.0 APIs, Library ini menggunakan hardware 3D acceleration dan highly optimized 3D software rasterizer.

g. FreeType

Bitmap dan Vector font rendering

h. SQlite

Relational database engine yang powerful dan ringan untuk semua aplikasi.

4. Application Framework

Application framework merupakan serangkaian tool dasar seperti alokasi resource smartphone, aplikasi telepon pergantian antar proses atau program dan pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki akses penuh kepada tool tool dasar tersebut, dan memanfaatkanya untuk menciptakan aplikasi yang lebih kompleks

Programmer mendapatkan akses penuh untuk memanfaatkan API-API *(Android Protocol Interface)* yang juga digunakan core applications. Arsitektur aplikasi didesain untuk menyederhanakan pemakaian kembali komponen komponen. Setiap aplikasi dapat menunjukan kemampuan dan aplikasi lain dapat memakai kemampuan tersebut, mekanisme yang sama memungkinkan pengguna mengganti komponen komponen yang dikehendaki.

Didalam semua aplikasi terdapat servis dan system yang meliputi :

a. Satu set **Views** yang dapat digunakan untuk membangun aplikasi meliputi lists, grids, text boxes, buttons dan embeddable web browser.
b. **Content Providers** yang mungkinkan aplikasi untuk mengakses data dari aplikasi lain, misalnya contacts atau untuk membagi data yang dimilikinya.
c. **Resource Manager** menyediakan akses ke non-code resources misalnya localized string, graphics dan layouts files.
d. **Notification Manager** yang memungkinkan semua aplikasi untuk menampilkan custom alerts pada status bar.
e. **Activity Manager** yang memanage life cycle of dari aplikasi dan menyediakan common navigation backstack

5. Applications

Android telah menyertakan aplikasi inti seperti email client, SMS, kalender, peta, browser, kontak, dan lainya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemograman Java. Pada layer ini lah developer menempatkan aplikasi yang dibuat. Yang istimewanya adalaha pada android semua aplikasi inti *(native)* maupun aplikasi pihak ketiga berjalan pada layer aplikasi dengan menggunakan library API yang sama, ini berate semua aplikasi yang dibuat untuk android akan memiliki akses yang setara dalam mengakses seluruh kemampuan handset tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain platform android ini programmer atau developer secara penuh akan bisa mengkostumisasi perangkat androidnya.

2.1.4 Aplikasi Android

Mulyadi, ST. (2010 : 10) mengatakan, Aplikasi android ditulis dalam bahasa pemograman Java yaitu kode java yang terkompilasi bersama sama dengan data dan file resources yang dibutuhkan oleh aplikasi yang digabungkan oleh *aapt tools* menjadi paket android, sebuah file yang ditandai dengan suffix *.apk* . file inilah yang didistribusikan sebagai aplikasi dan diinstall pada perangkat mobile, file ini yang didownload oleh pengguna ke perangkat mobile mereka. Semua kode dijadikan satu file *.apk* dan kemudian kita sebut sebagai sebuah aplikasi.

Dalam banyak cara masing masing aplikasi android aktif dalam lingkunganya sendiri :

a. Secara default semua aplikasi berjalan dalam proses Linux. Android memulai proses ketika salah satu dari kode aplikasi harus dijalankan dan menutup proses saat tidak lagi diperlukan dan sumber daya system tersedia untuk aplikasi lainya.

b. Setiap proses memiliki *virtual machine* (VM) sendiri, maka kode aplikasi yang dijalankan diisolasi dari semua kode aplikasi lain.

c. Secara default setiap alikasi diberikan sebuah user ID linux yang unik. Perizinan (*permissions)* diatur agar aplikasi dari file tersebut terlihat hanya untuk user yang memiliki ijin untuk aplikasi itu sendiri

1. Komponen Aplikasi

Salah satu fitur utama android adalah satu aplikasi dapat menggunakan elemen dari aplikasi yang lain. Misalnya jika kita memerlukan aplikasi untuk menampilkan *scrolling list* gambar dan aplikasi lain telah memiliki *scroller* yang sesuai dan membuatnya tersedia untuk aplikasi lain, kita dapat memanggil *scroller* tersebut untuk melakukan suatu pekerjaan, dari pada kita membuatnya lagi aplikasi yang lain atau me-*link* nya, sebaiknya hanya memanggil bagian dari aplikasi lainya saat membutuhkanya.

Agar ini bekerja, system harus dapat menjalankan sebuah proses aplikasi ketika salah satu bagian itu diperlukan dan menginisiasi objek java pada bagian tersebut. Karena itu, tidak seperti kebanyakan aplikasi lainya, pada system android tidak ada suatu *entry point* bagi semua yang ada dalam aplikasi misalnya tidak ada fungsi *main(),* sebaliknya mereka mempunyai komponen penting dalam system yang dapat diinisiasi *(instantiate)* dan dijalankan sesuai dengan kebutuhan.

Ada 4 jenis komponen yaitu :

a. *Activities*

Sebuah *activity* normalnya akan menyajikan UI bagi pengguna, dimana pengguna kemudian bisa melakukan interaksi padanya. Misalnya, sebuah *activity* dapat memberikan daftar menu kepada pengguna dan kemudian pengguna dapat memilih atau menampilkan foto. Sebuah aplikasi pesan teks mungkin memiliki *activity* yang menunjukan daftar kontak untuk mengirim pesan, *activity* kedua untuk menulis pesan pada kontak yang

dipilih dan *activity* lainya untuk memeriksa pesan pesan lama atau mengubah pengaturan. Walaupun mereka terlihat sebagai sebuah aplikasi, tetapi masing masing *activity* tersebut adalah mandiri. Masing masing diimplementasikan sebagai subkelas *(subclass)* dari kelas induk *(base class)* yang bernama *activity.*

Aplikasi mungkin hanya terdiri dari satu activity tetapi ada banyak aplikasi yang terdiri dari beberapa activity seperti aplikasi pengiriman pesan yang disebutkan diatas. Activity apa saja dan berapa banyak mereka tentu tergantung pada aplikasi dan desain. Biasanya, satu activity akan dipakai sebagai satu UI yang pertama diperlihatkan pada pengguna saat aplikasi diaktifkan. Untuk pindah dari satu activity ke activity yang lainya kita dapat melakukan satu event misalnya mengklik tombol, memilih opsi tertentu atau melalui trigger tertentu.

b. *Services*

Service tidak memiliki visual interface (UI) namun berjalan dilatar belakang untuk jangka waktu tak terbatas. Misalnya service mungkin memainkan music pada *backround* atau mungkin mengambil data melalui jaringan atau menghitung sesuatu dan memberikan hasil untuk *activity* yang membutuhkanya, setiap *services* meng-*extends* kelas induk *service*

Sebuah contoh adalah media player memutar lagu dari daftar lagu. Aplikasi player akan mungkin memiliki satu atau lebih *activity* yang memungkinkan pengguna untuk memilih lagu dan memainkanya. Namun,

player itu sendiri tidak akan ditangani oleh sebuah *activity* karena pengguna akan berharap musik tetap bermain bahkan setelah mereka meninggalkan player (missal mengirim pesan sms) untuk menjaga music tetap berjalan. *Activity* media player dapat menjalankan service untuk membuat musik tetap berjalan system kemudian akan membuat service tetap *running* bahkan setelah *activity* ditutup dari layar.

Ini memungkinkan untuk melakukan koneksi ke (*Binding)* service yang terus menerus *(running)* dan memulai servis jika belum berjalan. Saat tersambung, kita dapat berkomunikasi dengan layanan melalui sebuah interface yang di ekspos oleh servis tersebut. Untuk sevis musik, interface ini memungkinkan pengguna untuk *pause, rewind, stop and playback*

Seperti activity dan komponen lainya servis dijalankan pada *thread* utama dari proses aplikasi. Sehingga mereka tidak akan memblok komponen lain atau UI, mereka sering kali menimbulkan *thread* lain untuk *task* yang lama *time-consuming* (seperti pemutaran musik).

c. Broadcast Receiver

Broadcast Receiver merupakan komponen yang tidak melakukan apapun selain menerima dan bereaksi untuk menyiarkan notifikasi (*announcements).* Banyak broadcast berasal dalam sistem kode misanya notifikasi zona waktu yang telah berubah, bahwa baterai low, gambar telah diambil oleh kamera, atau pengguna mengubah preference bahasa, aplikasi juga dapat meninisiasi broadcast misanya memberikan informasi pada

aplikasi lain bahwa ada data yang telah didownload ke perangkat dan tersedia bagu mereka untuk digunakan.

Aplikasi dapat mempunyai sejumlah broadcasr receivers untuk merespon setiap notifikasi dan menanggapinya jika penting. Semua receiver meng-extend kelas induk *BroadcastReceiver*

Broadcast receiver tidak menampilkan UI. Namun mereka dapat memulai sebuah activity untuk merespon informasi yang mereka terima atau mungkin menggunakan *NotificationManager* mengingatkan pengguna, notifikasi dapat membuat pengguna melihat informasi yang disampaikan dalam berbagai cara, *flashing* lampu latar *backlight , vibrating*, pemutaran suara dan lain lain. Mereka biasanya menempatkan ikon yang akan terlihat dibar status dimana pengguna dapat membukanya untuk menerima pesan.

d. Content Providers

Content provider membuat set aplikasi data spesifik, tersedia, untuk aplikasi lain. Data dapat disimpan dalam file system dalam sebuah database SQlite atau cara lain yang bisa diterima. Content provider meng-extend kelas dasar *ContentProvider* untuk menerapkan standar method yang memungkinkan aplikasi lain unruk mengambil dan menyimpan data dari jenis yang dikontrolnya. Namun aplikasi ini tidak memanggil metode langsung sebaliknya mereka menggunakan objek *ContentResolver* dan memanggil metod tersebut. *ContentResolver* dapat berkomunikasi dengan

setiap konten provider yang bekerja sama dengan provider untuk mengelola setiap komunikasi interproses yang terlibat.

ContentProvider menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu aplikasi misalnya ketika kita akan menggunakan suatu aplikasi berbasis peta (Maps) dan membutuhkan cara untuk mengakses data kontak untuk prosedur navigasi, maka disinilah content provider akan berperan. Setiap kali ada permintaan yang harus ditangani oleh komponen tertentu android memastikan bahwa proses aplikasi dari komponen berjalan, memulainya jika diperlukan dan mengumumkan ketersediaan objek instant dari komponen, juga menciptakan objek instant jika diperlukan.

2.1.5 Pendekatan Terstruktur

Pendekatan Terstruktur adalah suatu proses untuk mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program. Pendekatan terstruktur merupakan pendekatan formal untuk memecahkan masalah-masalah dalam aktivitas bisnis menjadi bagian-bagian kecil yang dapat diatur dan berhubungan untuk kemudian dapat disatukan kembali menjadi satu kesatuan yang dapat dipergunakan untuk memecahkan masalah. Pendekatan terstruktur dalam pengembangan system informasi adalah proses yang berorientasi kepada teknik yang digunakan untuk merancang dan menulis program secara jelas dan konsisten.

Prinsip dari pendekatan terstruktur adalah jika suatu proses telah sampai pada suatu langkah tertentu, maka proses selanjutnya tidak boleh mengeksekusi langkah sebelumnya.

Ciri-ciri pendekatan terstruktur, yaitu:

1. Merancang berdasar modul. Modularisasi adalah proses yang membagi suatu system menjadi beberapa modul yang dapat beroperasi secara independen.
2. Bekerja dengan pendekatan *top-down*. Dimulai dari tingkat atas (secara global) kemudian diuraikan sampai ke tingkat modul (rinci).
3. Dilakukan secara iterasi.
4. Kegiatan dilakukan secara parallel.

2.1.6 CRUD

Dalam pemrograman komputer , Create, Read, Update dan Delete (sebagai akronim CRUD) Kadang-kadang disebut SCRUD dengan "S" untuk Search, adalah empat fungsi dasar penyimpanan persisten. Kadang-kadang *CRUD* diperluas dengan kata-kata *Retrieve* alih-alih *Read*, *Modify* bukannya *Update* , atau *Destroy* bukannya *Delete* . Hal ini juga kadang-kadang digunakan untuk menggambarkan User Interface (UI) konvensi yang memfasilitasi untuk melihat, mencari, dan mengubah informasi Istilah ini mungkin pertama kali dipopulerkan oleh James Martin dalam buku 1983 *Mengelola Lingkungan data-base* . akronim ini dapat diperpanjang menjadi CRUDL untuk L sebagai Listing, dari kumpulan data besar yang membawa

kompleksitas tambahan seperti pagination saat data set yang terlalu besar untuk dapat ditampung dengan mudah dalam memori.

Variasi lain dari CRUD adalah BREAD, singkatan dari "*Browse, Read, Edit, Add, Delete".*

Penerapan CRUD

Singkatan CRUD mengacu pada semua fungsi utama yang diimplementasikan dalam relasional database sebuah aplikasi. Setiap huruf dalam akronim dapat dipetakan ke standar pernyataan SQL ,Metode HTTP atau operasi DDS : **Tabel 2.1.** Penerapan CRUD pada database

Operasi	SQL	HTTP	DDS
Create	INSERT	PUT / POST	Write
Read (Retrieve)	SELECT	GET	Read / Take
Update (Modify)	UPDATE	PUT / PATCH	Write
Delete (Destroy)	DELETE	DELETE	Dispose

Fungsi CRUD dapat diimplementasikan dengan objek database , sebuah basis data XML, file teks , format file kustom, dan lainya. CRUD juga relevan di tingkat user interface dari sebagian besar aplikasi. Misalnya, dalam software buku alamat . dalam hal ini, perangkat lunak harus memungkinkan pengguna untuk :

- Membuat atau menambahkan entri baru
- Baca, mengambil, mencari, atau melihat entri yang ada
- Memperbarui atau mengedit entri yang sudah ada
- Hapus / menonaktifkan entri yang sudah ada

Tanpa setidaknya empat operasi ini, perangkat lunak tidak dapat dianggap lengkap. Karena operasi ini sangat mendasar, hal ini sering didokumentasikan dan dijelaskan di bawah satu judul yang komprehensif, seperti "manajemen kontak", "manajemen konten" atau "maintenance kontak" (atau "manajemen dokumen" secara umum, tergantung pada unit penyimpanan dasar untuk aplikasi tertentu).

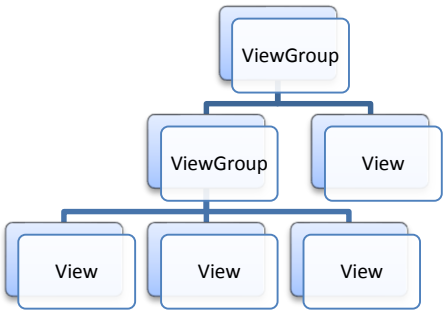
2.1.7 User Interface (UI)

User Interface (UI) menurut Mulyadi, ST. (2010: 45) pada android dibangun dengan menggunakan objek View dan ViewGroup. Ada banyak tipe dari View dan ViewGroup, setiap dari tipe itu merupakan turunan dari kelas Views. Objek view merupakan unit dasar dari UI pada platform android, kelas View menjadi kelas induk (*base class)* dari subkelas yang bernama Widgets, sebuah objek UI yang bisa dipakai dan tersedia secara lengkap. Seperti Teks Field, tombol dan lain lain. Sedangkan kelas ViewGroup tersedia sebagai kelas induk untuk subkelas yang dinamakan Layout yang menyediakan arsitektur layout seperti linear, tabular dan relative.

Objek view merupakan struktur data yang menyimpan properties dari paramenter layout dan konten secara spesifik untuk area layer (*rectangular area)* objek view menangani ukuran layout, drawing, perubahan fokus, *scrolling* dan interaksi key/gesture dari area layar. Sebagai sebuah objek pada UI, view juga merupakan titik interaksi bagi user dan tempat menerima interaksi even

1. Hirarki View

Pada platform android kita membuat UI actifity dengan menggunakan Hirarki View dan ViewGroup . seperti terlihat pada gambar dibawah ini. Pohon hirarki bisa sederhana atau rumit dan kompleks seperti yang kita butuhkan dan kita bisa membangunnya dengan widget dan layout yang telah ada pada android (*predefined)* atau dengan View yang kita buar mandiri.



Gambar 2.2. Hirarki View

(Sumber : Mulyadi,ST. 2010: 45)

Agar layar bisa menampilkan tampilan (sesuai dengan hirarki view) activity yang kita buat harus memanggil method *setContentView()* dan mempassing refrensi pada objek yang ingin ditampilkan. System android kemudian akan menerima refrensi itu dan menggunakan sebagai nilai tampilan yang akan ditampilkan.

2. Layout

Secara mendasar untuk membuat dan layout dan memperlihatkan Hirarki view adalah menggunakan file layout XML. XML memberikan struktur yang mudah dipahami seperti halnya dengan HTML, setiap elemen XML adalah salah satu objek View atau ViewGroup atau gabungan keduanya. Objek View seperti daun pada sebuah pohon dan objek ViewGroup adalah cabang pohonnya.

3. Widget adalah objek View yang bertugas sebagai sebuah antarmuka *(interface)* untuk interaksi dengan pengguna. Android menyediakan pengimplementasian widget secara penuh seperti tombol, checkbox, dan fields teks oleh karena itu kita bisa secara cepat membangun UI pada aplikasi android seperti date picker, jam, atau tombol zoom. Tetapi kita tidak dibatasi dengan widget yang tersedia oleh platform android, jika dibutuhkan kita bisa membuat sendiri suatu yang costumized atau elemen elemen acitionable dengan mendefinisikan sendiri objek View atau meng-Extend dan mengkombinasikan widget yang ada.

4. Even UI

Sekali kita menambah Views atau Widget kedalam UI kita akan berpikir bagaimana mekanisme interaksi pengguna dengan mereka untuk itu kita harus melakukan satu dari dua hal ini :

a. Tetapkan sebuah event listener dan mengregisternya dengan View

   Ini biasanya akan lebih sering kita lakukan daripada tidak karena ini adalah cara bagaimana kita memonitor adanya sebuah even. Kelas View memiliki koleksi sekumpulan interface dan dinamakan *On*

<sesuatu>Listener dari masing masing memiliki Methode untuk mengembalikan nilai (*callback)* On<sesuatu>(). Sebagai contoh *View.OnclickListener (untuk menghandle "klik" pada view), view.OnTouchListener (untuk menghandle even sentuh dilayar pada view) dan View.OnKeyListener (untuk menghandle keypad pada view)* maka apabila kita ingin View diberi pesan ketika di klik, implementasikan mengembalikan nilai *OnClick ()* dan meregisternya pada view dengan *setOnClickListener().*

b. Overide method pengembalian nilai (*callback)* yang ada untuk View

   Ini yang harus dilakukan ketika kita mengimplementasikan kelas View kira sendiri dan memonitor spesifik event yang terdapat padanya contohnya event yang terjadi ketika layar disentuh (*onTouchEvent()*) ketika trackball digerakan (*onTrackballEvent*()) atau ketika keypad ditekan *(onKeyDown()*) untuk itu kita dibolehkan untuk mendefinisikan default prilaku *(behavior)* setiap event dalam View dan menentukan apakah even akan diparsing kedalam Child View yang lain. Sekali lagi ini kembali ke kelas View, kita hanya diberikan kesempatan untuk mendefinisikan hanya ketika kita akan membuat komponen sendiri (*custom component)*

c. Menu aplikasi adalah hal penting dalam UI aplikasi, untuk memperlihatkan fungsi dan seting dari aplikasi. Menu dasar dari aplikasi terbuka dengan menekan tombol MENU pada perangkat. Selain itu kita

juga bisa menambahkan menu konteks (*context menu*) yang terbuka ketika pengguna menekan dan menentuh pada sebuah item.

Menu juga terstruktur dengan menggunakan hirarki View tetapi kita tidak membuat struktur sendiri melainkan kita membuat method dengan menggunakan *onCreateOptionsMenu()* atau *onCreateContextMenu()* untuk activity kita akan mendeklarasikan item yang akan kita masukan kedalam menu. Secara otomatis android akan membuat hirarki View yang dibutuhkan untuk menu dan menampilkan setiap item menu didalamnya

Menu juga menangani even mereka sendiri, maka dari itu tidak diperlukan meregister event listener didalam menu kita. Ketika item didalam menu di pilih, method *onOptionItemSelected()* atau method *onContextItemSelected()* akun dipanggil oleh framework.

Dan seperti layout aplikasi kita, memiliki opsi untuk mendeklarasikan item menu didalam file XML.

2.2 Basis Data

Perkembangan sistem basis data semakin tahun semakin berkembang dengan tujuan untuk mengelola data dan informasi agar dapat memenuhi tuntutan kebutuhan masyarakat akan data spasial yang semakin meningkat melalui kegiatan peningkatan pendayagunaan basis data untuk menunjang berbagai kegiatan yang dibutuhkan manusia. Dari berbagai kegiatan ini memberikan suatu kemudahan kepada pengguna data melalui pemanfaatan sistem penelusuran dan pemrosesan data.

2.2.1 Definisi Basis Data

Database adalah koleksi atau kumpulan data yang mekanis, terbagi / *shared*, terdefinisi secara normal dan dikontrol terpusat pada organisasi.
(Everest. Gordon C, 2005).

Sedangkan perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar disebut *Database Management Sistem (DBMS) .DBMS* dapat menjadi alternatif penggunaan secara khusus untuk aplikasi.

2.2.2 SQlite

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat ACID-compliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp.

Tidak seperti pada paradigma client-server umumnya, Inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead,latency times*, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

a. Fitur-fitur

Pustaka SQLite mengimplementasikan hampir seluruh elemen-elemen standar yang berlaku pada SQL-92, termasuk transaksi yang bersifat *atomic*, konsistensi basisdata, isolasi, dan durabilitas (dalam bahasa inggris lebih sering disebut ACID), trigger, dan kueri-kueri yang kompleks. Tidak ada pengecekan tipe sehingga data bisa dientrikan dalam bentuk string untuk sebuah kolom bertipe integer. Beberapa kalangan melihat hal ini sebagai sebuah inovasi yang menambah nilai guna dari sebuah basisdata, utamanya

ketika digunakan dalam bahasa pemrograman berbasis script (PHP, Perl), sementara kalangan lain melihat hal tersebut sebagai sebuah kekurangan.

Beberapa proses ataupun thread dapat berjalan secara bersamaan dan mengakses basisdata yang sama tanpa mengalami masalah. Hal ini disebabkan karena akes baca data dilakukan secara paralel. Sementara itu akses tulis data hanya bisa dilakukan jika tidak ada proses tulis lain yang sedang dilakukan; jika tidak, proses tulis tersebut akan gagal dan mengembalikan kode kesalahan (atau bisa juga secara otomatis akan mencobanya kembali sampai sejumlah nilai waktu yang ditentukan habis). Hanya saja ketika sebuah tabel temporer dibuat, mekanisme penguncian pada proses multithread akan menyebabkan masalah Update yang terkini (versi 3.3.4) dikatakan telah memperbaiki masalah ini. Sebuah program yang mandiri dinamakan sqlite disediakan dan bisa digunakan untuk mengeksekusi kueri dan memanajemen file-file basisdata SQLite. Program tersebut juga merupakan contoh implementasi penulisan aplikasi yang menggunakan pustaka SQLite.

Integrasi dengan bahasa lain

a. SQLite termasuk dalam framework REALbasic, yang memungkinkan aplikasi yang dikembangkan dengan menggunakan REALbasic dapat memanfaatkan basisdata SQLite.
b. Pustaka SQLite bisa digunakan secara langsung pada bahasa C/C++, namun untuk Tcl dan beberapa bahasa pemrograman berbasis script juga tersedia.
c. Modul DBI/DBD untuk Perl juga tersedia pada CPAN, *DBD:SQLite*, namun modul ini bukanlah antarmuka dengan SQLite melainkan memasukkan SQLite secara keseluruhan dalam modul tersebut.
d. Modul Python juga tersedia (PySQLite) yang diimplementasikan pada DB API Python versi 2.0 (PEP 249)
e. PHP dimulai dengan PHP5 telah memasukkan SQLite, versi PHP4 sebelumnya bisa juga digunakan untuk mengakses SQLite, namun modul SQLite tidak dimasukan secara standar bawaan.
f. Dimulai Lazarus versi 0.9.8 dan Free Pascal 2.0.0, SQLite didukung untuk digunakan oleh para programmer. Tutorial tersebut bisa didapatkan disini.
g. Meskipun borland tidak mempaketkan SQLite secara standar bawaan, Delphi sudah mendukung SQLite juga menggunakan pustaka yang dibuat oleh pihak ketiga (Aducom dan Zeos).
h. SQLite juga termasuk dalam paket yang dibundle secara standar bawaan pada Mac OS X, dan digunakan sebagai salah satu pilihan mekanisme penyimpanan data pada API Apple.

2.3 Pengertian Aplikasi

Aplikasi merupakan penerapan pengimplementasian suatu hal, data permasalahan pekerjaan dalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan permasalahan tersebut sehingga berubah menjadi bentuk yang baru tanpa menghilangkan nilai dasar dari hal, data, permasalahan, atau pekerjaan.

2.3.1 Karakteristik Aplikasi

Aplikasi adalah suatu sistem *software* yang berbasiskan teknologi dan standard dari konsorsium *world wide web* yang menyediakan sumber yang bersifat spesifik seperti konten atau layanan melalui sebuah *user interface* yang disebut *web browser*. Aplikasi web mempunyai beberapa karakteristik ,yang dapat di uraikan sebagai berikut :

1. Aplikasi web cenderung terus meningkat, dalam arti kata bahwa, pengembangan aplikasi web baru akan terealisasi pada saat aplikasi tersebut di implementasikan.
2. Aplikasi web berbeda dengan *software* aplikasi. Konten terdiri dari berbagai bentuk format data seperti teks, grafis, *image, audio, video* yang diintegrasikan oleh pemrosesan prosedural (*web programming*), metoda yang digunakan dalam menampilkan dan mengatur konten tersebut akan berdampak terhadap waktu respon dari sistem.
3. Aplikasi web ditujukan untuk digunakan oleh komunitas user yang besar,beragam dan sejumlah *user* yang tidak dikenali (*public user*) dengan berbagai kebutuhan,harapan dan kemampuan. Oleh karena itu pada saat

mengembangkan aplikasi web, *user interface* dan fitur-fitur kemudahan dan penggunaan (*usability features*) harus mempu menjawab kebutuhan dari semua user tersebut tanpa harus melalui suatu program latihan.

4. Semua aplikasi web menuntut aspek "*good look and feel*", suatu aspek yang memenuhi nilai estetika dan *art*, sehingga pengguna merasa nyaman dalam menggunakan dan mengakses aplikasi web yang dikembangkan.

5. Perubahan teknologi yang cepat menjadi tantangan bagi teknologi dan standar web seperti berkembangnya bahasa-bahasa baru, *tool* baru yang kemungkinan mempunyai error dan bagus (*technology instability*)

6. Media pengiriman konten aplikasi web sangat berbeda dengan *software* tradisional.

7. Keamanan dan privasi lebih dibutuhkan oleh sistem-sistem berbasis web jika di bandingkan dengan *software* tradisional.

2.4 Pengertian Kamus

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata. Ia berfungsi untuk membantu seseorang mengenal perkataan baru. Selain menerangkan maksud kata, kamus juga mungkin mempunyai pedoman sebutan, asal-usul (etimologi) sesuatu perkataan dan juga contoh penggunaan bagi sesuatu perkataan. Untuk memperjelas kadang kala terdapat juga ilustrasi di dalam kamus. Biasanya hal ini terdapat dalam kamus bahasa Perancis.

Kata kamus diserap dari bahasa Arab *qamus* (ق اموس), dengan bentuk jamaknya *qawamis*. Kata Arab itu sendiri berasal dari kata Yunani *Ωκεανός* (*okeanos*) yang berarti 'samudra'. Sejarah kata itu jelas memperlihatkan makna dasar yang terkandung dalam kata kamus, yaitu wadah pengetahuan, khususnya pengetahuan bahasa, yang tidak terhingga dalam dan luasnya. Dewasa ini kamus merupakan khazanah yang memuat perbendaharaan kata suatu bahasa, yang secara ideal tidak terbatas jumlahnya.

2.4.1 Jenis-jenis kamus

Berdasarkan penggunaan bahasa kamus dapat ditulis dalam satu atau lebih dari satu bahasa. Dengan demikian kamus terbagi menjadi beberapa jenis yaitu:

a. Kamus Eka Bahasa

Kamus ini hanya menggunakan satu bahasa. Kata-kata (entri) yang dijelaskan dan penjelasannya adalah terdiri dari bahasa yang sama. Kamus ini memiliki perbedaan yang jelas dengan kamus dwibahasa karena penyusunan dibuat berdasarkan pembuktian data korpus (data yang dipakai sebagai sumber bahan penelitian). Ini bermaksud definisi makna ke atas kata-kata adalah berdasarkan makna yang diberikan dalam contoh kalimat yang mengandung kata-kata berhubungan. Contoh bagi kamus ekabahasa adalah *Kamus Besar Bahasa Indonesia* (di indonesia).

b. Kamus Dwibahasa

Kamus ini menggunakan duaa bahasa, yaitu kata masukan dari bahasa yang dikamuskan diberi padanan atau penerjemahan dengan menggunakan bahasa yang lain. Contohnya *Kamus Inggris-Indonesia.*

c. Kamus Aneka Bahasa

kamus aneka bahasa sekurang-kurangnya menggunakan tiga bahasa atau lebih. Misalnya, kata bahasa Melayu, bahasa Inggris, dan bahasa Mandarin secara serentak. Contoh untuk kamus aneka bahasa adalah *Kamus Melayu-Cina-Inggris Pelangi* susunan Yuen Boon Chan pada tahun 2004.

2.5 Sejarah Bahasa Sunda

Bahasa Sunda adalah sebuah bahasa dari cabang Melayu-Polinesia dalam rumpun bahasa Austronesia. Bahasa ini dituturkan oleh sekitar 27 juta orang dan merupakan bahasa dengan penutur terbanyak kedua di Indonesia setelah Bahasa Jawa. Bahasa Sunda dituturkan di sebagian besar provinsi Jawa Barat (kecuali kawasan pantura yang merupakan daerah tujuan urbanisasi di mana penutur bahasa ini semakin berkurang), melebar hingga batas Kali Pemali (Cipamali) di wilayah Brebesdan Majenang, Cilacap Jawa Tengah, dan di kawasan selatan provinsi Banten. Dari segi linguistik, bersama bahasa Baduy, bahasa Sunda membentuk suatu rumpun bahasa Sunda yang dimasukkan ke dalam rumpun bahasa Melayu-Sumbawa.

Bahasa Sunda terutama dipertuturkan di sebelah barat pulau Jawa, di daerah yang dijuluki Tatar Sunda. Namun demikian, bahasa Sunda juga dipertuturkan di bagian barat Jawa Tengah, khususnya di Kabupaten Brebes dan Cilacap. Banyak nama-nama tempat di Cilacap yang masih merupakan nama Sunda dan bukan nama Jawa seperti Kecamatan Dayeuhluhur,Cimanggu, dan sebagainya. Ironisnya, nama Cilacap banyak yang menentang bahwa ini

merupakan nama Sunda. Mereka berpendapat bahwa nama ini merupakan nama Jawa yang "disundakan", sebab pada abad ke-19 nama ini seringkali ditulis sebagai "Clacap".

Selain itu menurut beberapa pakar bahasa Sunda sampai sekitar abad ke-6 wilayah penuturannya sampai di sekitar Dataran Tinggi Dieng di Jawa Tengah, berdasarkan nama "Dieng" yang dianggap sebagai nama Sunda (asal kata *dihyang* yang merupakan kata bahasa Sunda Kuna). Seiring mobilisasi warga suku Sunda, penutur bahasa ini kian menyebar. Misalnya, di Lampung, di Jambi, Riau dan Kalimantan Selatan banyak sekali, warga Sunda menetap di daerah baru tersebut.

2.5.1 Vokal, Konsonan, dan Suku Kata

1. Vokal

Vokal atau huruf hidup yaitu bunyi ujaran yang terjadi karena udara yang keluar dari paru-paru tidak mendapat rintangan sedikitpun. Jenis vokal bergantung pada 3 hal, yaitu :

a. Posisi bibir, yaitu bentuk bibir waktu mengucapkan bunyi. Bila bentuknya bundar, terjadilah vokal bundar, yaitu o, u, a. Bila bentuknya rata, terjadilah vokal tak bundar, yaitu i, e.

b. Tinggi rendahya lidah

1. Kalau ujung lidah dan belakang lidah dinaikkan, terjadi vokal depan, yaitu i, e.

2. Kalau hanya bagian belakang yang diangkat, terjadi vokal belakang yaitu u,o, a.

3. Kalau lidah itu rata terjadilah bunyi ujaran, disebut vokal pusat yaitu é (pepet).

c. Maju mundurnya lidah

Yang dimaksud dengan maju mundurnya lidah ialah jarak yang terjadi antara lidah dan alveolum.

2. Vokal dalam Bahasa Sunda

Dalam bahasa Sunda, vokal disebut juga aksara macakal, yaitu huruf yang mengubah bunyi. Masing-masing berupa istilah tersendiri, yaitu :

1. panghulu, huruf untuk mengubah bunyi menjadi i.
2. panyuku, huruf untuk mengubah bunyi menjadi u.
3. panéléng, huruf untuk mengubah bunyi menjadi é (téléng).
4. pamepet, huruf untuk mengubah bunyi menjadi e (pepet).
5. panolong, huruf untuk mengubah bunyi menjadi o.
6. paneuleung, huruf untuk mengubah bunyi menjadi eu.

Vokal dalam bahasa Sunda ada tujuh, yaitu a, i, u, é, o, e, eu. Distribusinya dalam kata dapat disimpan di awal kata, di tengah, atau di akhir kata.

3. Konsonan

Konsonan ialah bunyi ujaran yang terjadi karena udara keluar dari paruparu mendapat halangan. Halangan itu dapat seluruhnya atau sebagian dengan menggerakkan dan mengadukkan arus yang keluar.

Dalam bahasa Sunda, huruf konsonan disebut cacarakan, yaitu ha na ca ra ka da ta sa wa la pa (dha) ja ya nga ma ga ba tha nga. huruf dha dan tha, jarang dipakai

dalam bahasa Sunda. Pada sekarang ini, huruf yang digunakan yaitu alphabet (susunan haruf latin), yaitu b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z, ditambah dengan konsonan ng dan ny, walaupun dibentuk dua huruf tetapi satu bunyi. Seperti halnya vokal, konsonan pun distribusinya dalam kata bahasa Sunda dapat berada di awal, di tengah, atau di akhir kata. Hanya konsonan c dan ny yang tidak bisa berada di akhir kata.

4. Suku Kata (Engang)

Suku kata yaitu bagian kata yang diucapkan dalam satu nafas. Sifat suku kata Ada dua macam sifat suku kata, yaitu :

a. Suku kata terbuka, apabila diakhiri oleh vokal. Misalnya : a - ya, bu - ku.

b. Suku kata tertutup, apabila diakhiri oleh konsonan. Misalnya : ban - teng, kam - pak.

2.5.2 Ragam Hormat Bahasa Sunda

Menurut Kamus Besar Bahasa Indonesia bahasa sopan/hormat adalah ragam bahasa yang dipakai dalam situasi sosial yang mewajibkan adanya norma sopan santun (1997:78). Menurut hasil Kongres Bahasa Sunda tahun 1986 di Cipayung Bogor, tatakrama bahasa Sunda yang disebut juga undak usuk basa Sunda (UUBS). Menurut penelitian tatakrama bahasa Sunda atau sering disebut juga Undak Usuk Basa Sunda (UUBS) bentuknya ada beberapa ragam (tingkat/jenis) yang biasanya digunakan dalam bahasa Sunda, diantaranya yaitu, ragam bahasa hormat, ragam bahasa loma/kasar.

Pada hakekatnya digunakan ragam hormat tidak lain untuk menunjukkan rasa hormat dari pembicara kepada yang diajak bicara dan pada siapa yang menjadi bahan pembicaraan. Ragam bahasa Sunda juga memiliki parameter pemakaiannya dengan melihat usia (tua atau muda), berpendidikan atau tidak, pria atau wanita. Undak Usuk Basa Sunda ada enam jenis diantaranya :

1. Basa Kasar

Basa kasar disebut juga bahasa loma. Digunakan kepada sesama, kepada teman yang sudah akrab. Selain itu (jaman dulu) selalu dipakai juga untuk berbicara kepada orang yang umur dan pangkat dan kedudukannya dibawah si pembicara. Atau bisa juga digunakan untuk membicarakan orang yang umurnya dibawah si pembicara. Contoh:

„Maneh rek asup ayena?‟ Dudi nanya ka Dadan

„Moal ah, moal wara asup, rek ngadagoan Rini heula,‟ tembal Dadan.

(„kamu mau masuk sekarang?‟ Dudi bertanya kepada Dadan)

(„Engga ah, engga akan masuk dulu, mau nungguin Rini dulu,‟ jawab Dadan)

Budi Rahayu (1993 : 15).

2. Basa sedeng

Basa sedeng sering juga disebut sebagai bahasa lemes keur ka sorangan (halus untuk diri sendiri), yaitu bahasa yang digunakan untuk diri sendiri seperti misalnya berbicara menggunakan bahasa halus atau untuk berbicara kepada orang yang lebih tua. Selain itu bahasa Sedang juga dapat dipakai untuk berbicara kepada orang yang belum dikenal atau akrab apabila yang mengajak berbicara menggunakan bahasa halus.

Contoh :

„Dudi mah tos lebet ti payun,‟ ceuk Dadan ka Pa Asep.

„Abdi mah teu acab lebet soteh bade ngantosan Rini heula.‟

(„Dudi sudah masuk duluan,‟ kata Dadan ke Pak Asep.‟)

(„Saya belum masuk karena mau menunggu Rini dulu.‟)

Budi Rahayu (1993 : 15).

3. Basa lemes

Basa lemes sering disebut juga sebagai bahasa lemes keur ka batur (bahasa halus untuk orang lain). Bahasa ini digunakan untuk berbicara kepada orang yang umurnya di atas pembicara dan untuk membicarakan orang yang pangkat, kedudukan dan umurnya di atas kita. Bahasa halus juga dapat dipakai kepada orang yang belum kita kenal. Contoh:

„Ku margi Bu Lia henteu lebet, engke kelas IIA lebetan bae ku Bapa,‟ ceuk Pa Kapala Sakola ka Pa Maman.

(„Karena Bu Lia tidak masuk, nanti kelas IIA Bapak saja yang masuk,‟ kata Pak Kepala Sekolah kepada Pak Maman.)

Budi Rahayu (1993 : 15).

4. Basa lemes pisan

Ragam bahasa ini dipakai untuk menghormati orang yang kedudukannya lebih tinggi dari pembicara. Contoh:

„Manawi Ibu uninga, Bapa Gubernur nu ayena di mana nya linggihna?‟

(„Mungin Ibu ingat, Bapak Gubernur yang sekarang tinggalnya di mana?‟)

Budi Rahayu (1993 : 43).

5. Basa kasar pisan

Ragam bahasa ini dapat disebut juga sebagai bahasa cohag. Bahasa ini biasanya dipakai oleh orang-orang yang sedang marah atau bertengkar dengan maksud untuk saling menghina. Tetapi umumnya, ragam bahasa ini ditujukkan untuk binatang karena jika ditujukkan pada manusia, bahasa ini akan terasa sangat kasar dan menyinggung.

Contoh:

„Kawas nu euweuh gadag deui we, pagadagan teh ngan leweh!‟ ceuk Dada ka adina nu keur ceurik.

(seperti tidak ada kerjaan lagi, kerjanya hanya menangis saja!‟ kata Dadan kepada adiknya yang lagi menangis).

Budi Rahayu (1993 : 43).

6. Basa panengah

Ragam bahasa ini dipakai untuk berbicara dengan orang yang pangkat dan kedudukannya di bawah pembicara tetapi umurnya di atas pembicara. Ragam bahasa ini dipakai juga ketika berbicara dengan orang yang menggunakan bahasa halus dan orang yang dibicarakannya itu memiliki pangkat dan kedudukan dibawah mereka, tetapi umurnya di atas mereka. Ragam bahasa ini tingkatannya ada di bawah bahasa halus tetapi di atas bahasa kasar. Contoh:

„Ari Emang di mana sare teh?‟ ceuk Bapa ka mang Endin

„Kalau Paman di mana tidurnya?‟ kata Bapak kepada mang Endin

Budi Rahayu (1993 : 114).