

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Sistem informasi menjadi hal yang sangat penting bagi salah satu unit kegiatan mahasiswa untuk meningkatkan sebuah layanan informasi yang memberikan kemudahan-kemudahan dalam melakukan pengolahan data kader yang sangat mendukung manajemen kearsipan.

Sistem informasi kesekretariatan telah diterapkan oleh UKI JASHTIS yang merupakan salah satu Unit Kegiatan Mahasiswa di STMIK AMIKOM Yogyakarta dalam pendataan jenjang kader sesuai dengan integritas dan spesifikasi kader anggotanya. Namun perlu pengembangan lebih lanjut dikarenakan timbul beberapa permasalahan dan follow-up dari penggunaan data tersebut.

Permasalahan sekarang yaitu sistem informasi tersebut berbasis desktop dan user yang menggunakan aplikasi tersebut tidak hanya dianalisis dan dirancang untuk Sekretariat Jenderal (SEKJEN) UKI JASHTIS saja, namun dipergunakan juga oleh bidang pendataan kader. Aplikasi tersebut memiliki kelemahan yaitu tidak bisa diakses dari jarak jauh, sulit untuk dideploy pada banyak user karena harus melakukan instalasi, dan sulit dalam pembaharuan data.

Oleh sebab itu perlu diadakan pengembangan sistem. Dengan aplikasi android bisa digunakan di smartphone dan fleksibel untuk pengolahan data

integritas dan spesifikasi kader oleh banyak user dibandingkan dengan menggunakan aplikasi desktop.

Berdasarkan keadaan tersebut, dapat diambil penelitian yang berjudul **“Pengembangan Sistem Informasi Kesekretariatan Unit Kerohanlan Islam Jamaah Shohwatul Islam (UKI JASHTIS) STMIK AMIKOM Yogyakarta Berbasis Android dan SMS Gateway”**. Hasil Penelitian berupa aplikasi sistem informasi kesekretariatan UKI Jashtis berbasis Android dan SMS Gateway.

### **1.2 Rumusan Masalah**

Adapun rumusan masalah dalam perancangan *system* yang akan di buat adalah sebagai berikut :

“Bagaimana mengembangkan sistem informasi kesekretariatan UKI JASHTIS yang saat ini berbasis desktop menjadi sistem informasi kesekretariatan UKI JASHTIS yang berbasis android dan SMS Gateway?”

### **1.3 Batasan Masalah**

Dalam penyusunan skripsi ini, agar pembahasan tidak terlalu meluas dan untuk mempermudah dalam penyelesaian nantinya, maka dituliskan beberapa batasan masalah sebagai berikut :

1. Informasi yang diberikan kepada admin adalah segala informasi yang tentang kader UKI JASHTIS diantaranya biodata, integritas, spesifikasi, dan jenjang kader anggotanya.
2. Sistem yang dibangun berbasis sistem aplikasi handphone smartphone android yang menggunakan sistem operasi android 2.2 (froyo) ke atas.

3. Sistem dapat digunakan dan dikelola sebagai media informasi dan sarana publikasi untuk kader dan anggota UKM tersebut yang didukung oleh SMS Gateway.
4. Data yang digunakan berasal dari Pengelola Sumber Daya Manusia atau bidang kaderisasi UKI JASHTISSTMIK AMIKOM Yogyakarta untuk mendukung validasi data.
5. Data yang digunakan dalam pembuatan aplikasi hanya sebagai sample untuk menjaga keamanan data.
6. Software yang digunakan untuk membangun aplikasi android tersebut adalah Eclipse dan java JDK.

#### **1.4 Tujuan Penelitian**

1. Mengembangkan sistem informasi kesekretariatan UKI Jashtis dalam pengolahan data biodata, integritas, spesifikasi dan jenjang kader.
2. Sebagai syarat kelulusan program SI serta untuk memperoleh gelar sarjana di STMIK Amikom Yogyakarta pada jurusan Sistem Informasi.

#### **1.5 Manfaat Penelitian**

##### **1.5.1 Bagi Penulis**

1. Bisa mengimplementasikan ilmu pengetahuan yang di dapat selama masa studi di STMIK Amikom Yogyakarta pada jurusan Sistem Informasi dan dapat memahami apa yang telah di dapat pada kuliah teori dan kuliah pratikum.

2. Dapat membuat aplikasi mobile sebagai media informasi dan sarana *publikasi* untuk kader dan anggota UKM tersebut yang didukung oleh SMS Gateway.

#### **1.5.2 Bagi UKI JASHTIS STMIC AMIKOM Yogyakarta**

1. Meningkatkan efisiensi dan efektifitas perekapan data biodata, spesifikasi, integritas, dan jenjang kader UKI JASHTIS STMIC AMIKOM Yogyakarta.
2. Memudahkan dalam mengakses informasi yang cepat, tepat dan akurat bagi pihak UKI JASHTIS STMIC AMIKOM Yogyakarta.
3. Menjaga ketersediaan dan keamanan arsip kader UKI JASHTIS STMIC AMIKOM Yogyakarta.
4. Sebagai sarana media informasi dan *publikasi* untuk kader dan anggota UKI JASHTIS STMIC AMIKOM Yogyakarta yang didukung oleh SMS Gateway.

#### **1.6 Metode Penelitian**

Sebagai usaha dalam memperoleh data yang akurat, relevan dan terarah sesuai dengan permasalahan yang dihadapi, maka perlu adanya suatu metode yang tepat untuk mencapai tujuan dalam penelitian. Untuk itu penulis mengembangkan berbagai metode pengumpulan data dalam penelitian skripsi ini yaitu :

##### **1.6.1 Metode Pengumpulan Data**

1. Metode Survei

Yaitu suatu pengumpulan data dengan cara melakukan tanya jawab atau wawancara secara langsung kepada Pengelola Sumber Daya Manusia atau bidang kaderisasi UKI JASHTIS STMIK AMIKOM Yogyakarta, untuk memperoleh data yang diperlukan.

## 2. Studi Literatur

Merupakan metode yang dilakukan dengan memanfaatkan literatur yang tersedia, seperti memanfaatkan fasilitas internet yaitu dengan mengunjungi situs web yang berhubungan dengan masalah yang dihadapi oleh penulis. Serta mengumpulkan referensi buku – buku yang tersedia.

## 3. Metode Kepustakaan

Metode dengan perolehan data dari buku-buku yang telah diterbitkan atau literatur-literatur yang merupakan sumber pengetahuan teori mengenai cara pembuatan website dimana semuanya berkaitan dengan permasalahan yang akan dibahas.

### 1.6.2 SDLC

#### 1. Analisis

Metode ini dilakukan dengan menganalisis permasalahan yang dihadapi penulis sehingga dapat menyelesaikannya dengan mudah.

#### 2. Perancangan Sistem

Tahap ini merancang sistem yang akan dibuat berdasarkan hasil analisa yang telah dilakukan.

### 1.6.3 Implementasi

#### 1. Pembuatan Program

Tahap ini melakukan implementasi dari hasil perancangan sistem yang telah dilakukan.

#### 2. Pengujian Program

Pada tahap ini dilakukan pengujian program apakah sudah berjalan dengan baik atau belum, dan dapat digunakan sesuai harapan.

### 1.7 Sistematika Penulisan

Sistematika penulisan skripsi ini penulis susun dengan format sebagai berikut:

#### BAB I PENDAHULUAN

Bab ini merupakan bagian pengantar dari pokok masalah yang dibahas dalam skripsi ini. Adapun hal-hal yang dibahas berisikan latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika penulisan skripsi.

#### BAB II LANDASAN TEORI

Dalam bab ini diuraikan diuraikan literatur yang berhubungan dengan penelitian yang berisi pengenalan sistem informasi secara umum, konsep dasar pengembangan sistem, konsep desain sistem, konsep

pengujian sistem, konsep dasar aplikasi mobile, komponen perangkat lunak (software) yang digunakan dan pengenalan tentang SMS Gateway.

### **BAB III ANALISIS DAN PERANCANGAN SISTEM**

Pada Bab ini akan diuraikan tentang tinjauan umum, analisis sistem, dan perancangan program.

### **BAB IV IMPLEMENTASI DAN PEMBAHASAN**

Pada Bab ini membahas implementasi dan pembahasan yaitu : implementasi sistem, uji coba sistem, pengguna sistem dan lain lain.

### **BAB V PENUTUP**

Bab ini merupakan bagian akhir dari penulisan skripsi yang berisikan kesimpulan dan saran dari keseluruhan isi laporan.

### **BAB V PENUTUP**

Bab ini merupakan bagian akhir dari penulisan skripsi yang berisikan kesimpulan dan saran dari keseluruhan isi laporan.

### **DAFTAR PUSTAKA**

Daftar pustaka berisi referensi-referensi yang digunakan dalam pembuatan sistem informasi berbasis Android dan SMS Gateway.

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Penelitian yang dilakukan oleh Nur Adhani (2010) dalam tugas akhirnya dengan judul “Sistem Informasi Kesekretariatan Unit Kerohanian Islam Jamaah Shohwatul Islam (UKI Jashtis) STMIK AMIKOM Yogyakarta”[1]. Dari hasil penelitian yang telah dilakukan oleh Adhani menunjukkan bahwa adanya manfaat positif dari pembuatan sistem informasi kesekretariatan tersebut sebagai media informasi dan promosi. Adhani juga menyimpulkan bahwa Sistem Informasi Kesekretariatan Unit kerohanian Islam Jamaah Shohwatul Islam(UKI JASHTIS) STMIK AMIKOM Yogyakarta ini dapat digunakan untuk membantu sistem kesekretariatan yang lama, dimana dalam proses pengolahan datanya dengan cara pencatatan manual. Kesamaan dengan peneliti adalah pemanfaatan sistem informasi kesekretariatan UKI Jashtis sebagai media informasi yang dapat memberikan informasi tentang spesifikasi, integritas dan jenjang kader yang dibutuhkan. Sedangkan yang berbeda adalah dari sisi penggunaan pembuatan aplikasinya dan *follow up* data dengan *sms gateway*.

Penelitian yang dilakukan oleh Majid Rahardi (2011) dalam skripsinya dengan judul “Pengecekan Resi Pengiriman Barang Berbagai Ekspedisi Berbasis SMS Gateway dan Aplikasi Android”[2]. Dari hasil penelitian yang telah dilakukan oleh Majid menunjukkan bahwa adanya manfaat positif dari penggunaan aplikasi



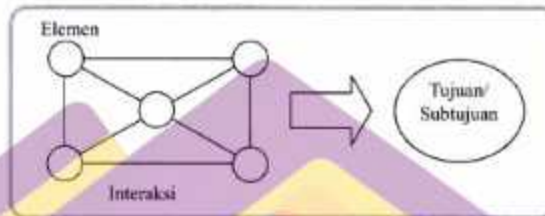
Android dan SMS Gateway dalam penelitiannya. Majid menyimpulkan bahwa Aplikasi berbasis android ini dibangun melalui tahap analisis yaitu dengan menggunakan analisis kebutuhan dan analisis kelayakan, setelah itu tahap perancangan mulai dari rancangan database, dan rancangan antar muka. Selain itu ia juga menyimpulkan bahwa aplikasi ini mampu bekerja menjalankan sistem sehingga dapat menampilkan *output* dari cek resi dan menyimpan data ke *database* untuk informasi bagi pengguna. Kesamaan dengan peneliti adalah pemanfaatan aplikasi android dan *sms gateway* pada objek skripsi. Sedangkan yang berbeda adalah dari sisi objek penelitian dan implementasinya .

## **2.2 Konsep Dasar Sistem**

### **2.2.1 Pengertian Sistem**

Sistem adalah suatu kesatuan usaha yang terdiri dari bagian – bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks[3]. Pengertian tersebut mencerminkan adanya beberapa bagian dan hubungan antar bagian, ini menunjukkan kompleksitas dari sistem yang meliputi kerja sama antara bagian yang interdependen satu sama lain. Selain itu, dapat dilihat bahwa sistem berusaha mencapai tujuan. Pencapaian tujuan ini menyebabkan timbulnya dinamika, perubahan yang terus menerus perlu di kembangkan dan dikendalikan. Definisi tersebut menunjukkan bahwa sistem sebagai gugus dari elemen – elemen yang saling berinteraksi secara teratur dalam rangka mencapai

tujuan atau sub tujuan. Pengertian sistem secara skematis dapat dilihat pada Gambar 2.1.



**Gambar 2.1 Pengertian Sistem**

### 2.2.2 Karakteristik Sistem

Sistem memiliki beberapa karakteristik yang merupakan sifat-sifat dari system tersebut[4], yaitu:

a. **Komponen sistem (*components*)**

Sejumlah komponen yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

b. **Batasan sistem (*boundary*)**

Merupakan daerah yang membatasi antara sudut sistem dengan sistem lain atau dengan lingkungan luarnya. Batasan sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. **Lingkungan luar sistem (*environments*)**

Lingkungan luar dari sistem adalah apapun dari sistem yang mempengaruhi operasi sistem.

d. Penghubung sistem (*interface*)

Merupakan media penghubung antara komponen sistem yang satu dengan komponen sistem yang lainnya, atau juga penghubung antara subsistem dengan subsistem lainnya.

e. Masukan sistem (*input*)

Merupakan suatu masukan yang dibutuhkan oleh suatu sistem untuk kemudian diproses menjadi *output* yang berguna sesuai dengan yang diinginkan.

f. Keluaran sistem (*output*)

*Output* merupakan keluaran dari hasil pemrosesan dalam suatu sistem atas masukan yang diberikan. Misalnya sebuah sistem informasi kampus mengolah data yang diinputkan ke sistem menjadi informasi yang berguna bagi *user*.

g. Pengolahan sistem (*process*)

Sebuah sistem harus mempunyai suatu bagian pengolahan yang bertugas mengolah *input* menjadi *output* yang diinginkan.

h. Sasaran sistem (*objectives*)

Tujuan yang ingin dicapai oleh sistem, jika suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya.

## 2.3 Konsep Dasar Informasi

### 2.3.1 Pengertian Informasi

Informasi merupakan kumpulan data yang sudah diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi pengguna.

Dari definisi diatas dapat diambil kesimpulan bahwa sumber informasi adalah data. Informasi dikatakan bernilai bila manfaatnya lebih besar dibandingkan biaya untuk mendapatkannya.

### 2.3.2 Kwalitas Informasi

Informasi yang berkualitas memiliki 3 kriteria[5], yaitu:

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan, tidak bisa ataupun menyesatkan.

2. Tepat pada waktunya (*timeliness*)

Informasi yang datang pada penerima tidak boleh terlambat. Didalam pengambilan keputusan, informasi yang sudah usang tidak lagi bernilai.

3. Relevan (*relevance*)

Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahas dengan informasi tersebut sehingga informasi mempunyai manfaat bagi pemakainya.

## 2.4 Konsep Dasar Sistem Informasi

### 2.4.1 Pengertian Sistem Informasi

Menurut Robert A. Laitch dan K. Roscoe Bavis (2007, h 8), Sistem informasi adalah suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sistem juga dapat diartikan sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap pengolah data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan.

#### **2.4.2 Komponen Sistem Informasi[6]**

Komponen sistem informasi memiliki komponen sebagai berikut:

a. Perangkat keras (*hardware*)

Komponen untuk melengkapi kegiatan memasukkan data, memproses data, dan keluaran data.

b. Perangkat lunak (*software*)

Program dan instruksi yang diberikan ke komputer.

c. Basis data (*database*)

Sekumpulan tabel, hubungan dan lain-lain yang berkaitan dengan penyimpanan data.

d. Jaringan komputer dan komunikasi data

Sistem penghubung yang memungkinkan sumber dipakai bersama-sama atau diakses oleh sejumlah pemakai.

e. Prosedur (*procedure*)

Sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang di harapkan.

f. Manusia (*human*)

Semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan dan pengguna keluaran sistem informasi.

## 2.5 Sistem Informasi Kesekretarlatan

Sekretaris adalah sebuah profesi yang dituntut adanya profesionalitas. Dikatakan sebuah profesi karena pekerjaan yang dilakukan berhubungan dengan orang lain. Dikatakan profesi yang membutuhkan profesionalitas, karena pekerjaan sekretaris juga terikat oleh tuntutan profesionalitas dan terikat pada kode etik sekretaris[7].

Konsep dasar ergonomi dikembangkan atas dasar pemikiran bahwa dalam setiap aktivitas manusia selalu dihadapkan kepada kendala fisik dan psikis. Untuk memenuhi aktivitas manusia yang nyaman, aman, sehat, efektif, dan efisien diperlukan desain peralatan, lingkungan kerja, tata cara kerja dan kondisi psikologi yang sesuai[8].

Dalam hubungannya dengan sekretaris, tugas seorang sekretaris secara garis besar dapat digolongkan menjadi dua yaitu tugas sekretaris dalam arti sempit dan

luas. Tugas sekretaris dalam arti sempit yaitu sebagai orang yang dipercaya oleh pimpinan untuk menyimpan rahasia pimpinan. Sedangkan tugas sekretaris dalam arti luas adalah sebagai pelaksanaan tugas-tugas yang bersifat membantu manajer atau pimpinan untuk menjalankan roda organisasi, perusahaan maupun kantor.

Secara umum tugas-tugas sekretaris yang dijalankan adalah menerima telepon, melaksanakan korespondensi, menyimpan dokumen atau arsip yang dinilai penting, menerima tamu-tamu pimpinan, membuat jadwal pertemuan dan perjanjian-perjanjian pimpinan dengan relasi maupun kegiatan lainnya, menyiapkan bahan-bahan keterangan kepada pimpinan dalam rapat maupun kegiatan lainnya, bertindak sebagai perantara pimpinan dan bawahan, mengatur rapat-rapat dan seminar pimpinan dan bawahan, menemani pimpinan dalam pertemuan penting dan menyusun sambutan atau pidato untuk pimpinan.

Dalam menjalankan tugasnya seorang sekretaris memerlukan sarana dan prasarana agar dapat bekerja secara efisien untuk mencapai efektivitasnya maka ia perlu mempelajari seluk beluk perusahaan, mempelajari pekerjaan, menjalankan rencana kerja yang fleksibel, mengelola pekerjaan kecil, memfokuskan pada satu tugas, mendahulukan tugas yang sulit, merencanakan tugas berselang-seling mengelompokkan tugas-tugas yang setara, merencanakan pekerjaan agar sekali jadi.

Sistem Pengolahan Transaksi (*Transaction Processing System* disingkat TPS) adalah sistem yang menjadi pintu utama dalam pengumpulan dan pengolahan data pada suatu organisasi. Tugas utama TPS adalah mengumpulkan dan

mempersiapkan data untuk keperluan sistem informasi yang lain dalam organisasi, misalnya kebutuhan sistem informasi manajemen, atau kebutuhan sistem informasi eksekutif[9].

Tugas pokok dari sistem pengolahan transaksi antara lain:

1. Pengumpulan data : setiap organisasi berinteraksi langsung dengan lingkungannya dalam penyediaan jasa dan produk, pasti memerlukan sistem yang mengumpulkan data transaksi yang bersumber dari lingkungan.
2. Manipulasi data : data transaksi yang dikumpulkan biasanya diolah lebih dahulu sebelum disajikan sebagai informasi untuk keperluan bagian-bagian dalam organisasi atau menjadi bahan masukan sistem informasi yang lebih tinggi. Beberapa tugas manipulasi data adalah sebagai berikut:
  3. Klasifikasi : data dikelompokkan menurut kategori tertentu, misalnya menurut jenis kelamin, menurut agama, menurut golongan, dsb.
  4. Sortir : data dikelompokkan menurut urutan tertentu agar lebih mudah dalam pencarian data, misalnya di-sortir menurut abjad nama, atau menurut nomor induk, dsb.
  5. Perhitungan : melakukan operasi aritmatika terhadap elemen data tertentu, misalnya menjumlahkan penerimaan dan pengeluaran setiap hari, atau menghitung jumlah hutang pelanggan, dsb.
  6. Pengikhtisaran : melakukan peringkasan data (summary) seperti sintesa data menjadi total, sub-total, rata-rata, dsb.



7. Penyimpanan data : data transaksi harus disimpan dan dipelihara sehingga selalu siap memenuhi kebutuhan para pengguna.
8. Penyiapan dokumen : beberapa dokumen laporan harus disiapkan untuk memenuhi keperluan unit-unit kerja dalam organisasi.

Karakteristik sistem pengolahan transaksi antara lain:

1. Volume data yang diproses relatif sangat besar.
2. Kapasitas penyimpana data (*database*) sangat besar.
3. Kecepatan pengolahan yang diperlukan sangat tinggi agar banyak data yang diproses dalam waktu singkat.
4. Sumber data umumnya internal dan keluarannya umumnya untuk keperluan internal.
5. Pengolahan data dilakukan secara periodik, harian, mingguan, bulanan, dsb.
6. Orientasi data yang dikumpulkan umumnya mengacu pada data masa lalu.
7. Masukan dan keluaran terstruktur, data diformat menurut suatu standar.
8. Komputasi tidak terlalu rumit.

## **2.6 Android**

### **2.6.1 Sejarah Sistem Operasi Android**

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. Membeli Android Inc. yang merupakan pendatang baru yang

membuat peranti lunak untuk ponsel atau *smartphone* Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia[10].

### **2.6.2 Versi Android**

Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android. Adapun versi-versi Android yang pernah dirilis adalah sebagai berikut :[11]

#### **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 bulan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 yang disebut *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)

Android versi 1.6 atau yang disebut *Donut* 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 diintegrasikan, 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 2.1/2.1 (Eclair)

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.

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

Pada tanggal 20 Mei 2010, Android versi 2.2 atau dikenal dengan nama *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 tanggal 6 Desember 2010, Android versi 2.3 atau yang disebut *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*), didukung kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

#### **7. Android Versi 3.0/3.1 (Honeycomb)**

Android *Honeycomb* atau Android versi 3.0 atau 3.1 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. Perangkat tablet dengan *platform* Android 3.0

#### **8. Android Versi 4.0 (ICS: Ice Cream Sandwich)**

Dirilis pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu

kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara *offline*, dan berbagi informasi dengan menggunakan NFC. Posel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

#### **9. Android Versi 4.1 (Jelly Bean)**

Android Jelly Bean yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dengan fitur baru. Penambahan baru diantaranya meningkatkan input *keyboard*, desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat. Tak ketinggalan Google Now juga menjadi bagian yang diperbaharui.

Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 muncul pertama kali dalam produk tablet Asus, yakni Google Nexus 7.

#### **10. Android versi Kitkat**

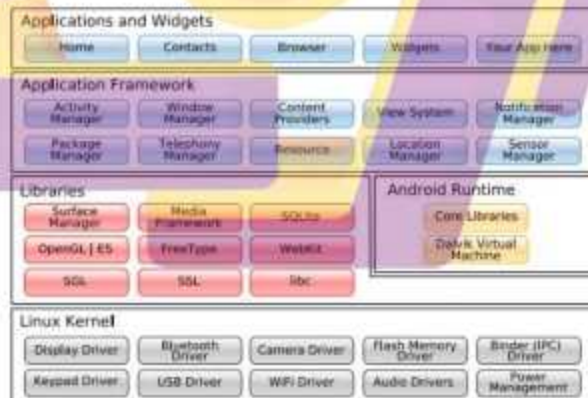
Android merupakan sistem operasi yang dikembangkan oleh google telah beberapa waktu kemarin telah meluncurkan Android versi 4.4 yang diberi nama KitKat[12]. Berbagai fitur yang telah disediakan oleh OS Android KitKat ini adalah perbaikan sistem penyimpanan sementara pada penggunaan memori, yang mana kinerja prosesor telah diminimalisir terhadap penyimpanan registry data sementara pada RAM dan secara langsung akan ditampung oleh kapasitas memori internal yang tersedia, sehingga loading prosesor akan terasa lebih ringan.

## 11. Android versi Lollipop

Android akhirnya secara resmi merilis sistem operasi Android versi 5.0 Lollipop. Secara teknis, android lollipop diklaim sebagai update sistem operasi terbesar yang pernah dilakukan Google. Konon, perangkat yang menggunakan OS Android L ini akan mampu berintegrasi antar perangkat seperti seperti smartphone, tablet dan smartwatch berbasis Android. Perangkat smartphone yang beruntung mencicipi sistem operasi ini untuk pertama kali adalah dari keluarga baru Nexus yang akan segera diluncurkan dalam waktu dekat ini, antara lain nexus 6 dan nexus 9.[13]

### 2.6.3 Arsitektur Android[14]

Secara garis besar Arsitektur dapat dijelaskan dan digambarkan sebagai berikut :



**Gambar 2.1 Arsitektur Android**

### 2.6.3.1. *Application dan Widgets*[15]

*Applications* dan *Widget* adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa sms dan panggilan telepon.

### 2.6.3.2. *Application Framework*

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resource*, menjalankan *service background*, mengatur alarm, dan menambah status *notifications*, dan sebagainya. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*).

Sehingga bisa kita simpulkan *Applications Frameworks* adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa sms dan panggilan telepon.

Komponen – komponen yang termasuk di dalam *Aplication Frameworks* adalah sebagai berikut :

1. *Views*
2. *Content provider*
3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager*

#### **2.6.3.3 Libraries**

*Libraries* adalah *layer* dimana fitur-fitur android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya, *layer* ini meliputi berbagai *library*.

1. *Libraries* media untuk memutar audio dan video.
2. *Libraries* untuk manajemen tampilan.
3. *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
4. *Libraries* SQLite untuk dukungan database.
5. *Libraries* SSL dan WebKit terintegrasi dengan *web browser* dan *security*.
6. *Libraries* LiveWebcore mencakup *modern web browser* dengan *engine embeded web view*.
7. *Libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API's.

#### **2.6.3.4 Android Run Time**



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

1. *Core Libraries* : Aplikasi Android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa java yang ditangani oleh *Core Libraries*.
2. *Dalvik Virtual Machine*: Virtual mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

#### **2.6.3.5 Linux Kernel**

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

#### **2.6.4 Fundamental Aplikasi**

Aplikasi Android ditulis dalam bahasa pemrograman java. Kode java terkompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi , di mana prosesnya dipackage oleh *tools* yang dinamakan “*apt tools*” yang kedalam

paket Android sehingga menghasilkan *file* dengan ekstensi apk. *File* apk itulah yang kita sebut dengan aplikasi, dan nantinya dapat diinstall di perangkat *mobile*. [16]

Aplikasi android sendiri memiliki komponen – komponen sebagai berikut:

a. *Activities*

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Secara hirarki sebuah *windows activity* dinyatakan dengan *method* `Activity setContentView()`. `ContentView` adalah objek yang berada pada *root* hirarki.

b. *Service*

*Service* tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*, sehingga aplikasi bisa tetap berjalan walau kita menjalankan aplikasi lain. *Service* dijalankan pada *thread* utama dari proses aplikasi.

c. *Broadcast Receiver*

*Broadcast Receiver* berfungsi menerima dan beraksi untuk menyampaikan notifikasi. *Broadcast Receiver* tidak memiliki *user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang mereka terima atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna.

d. *Content Provider*

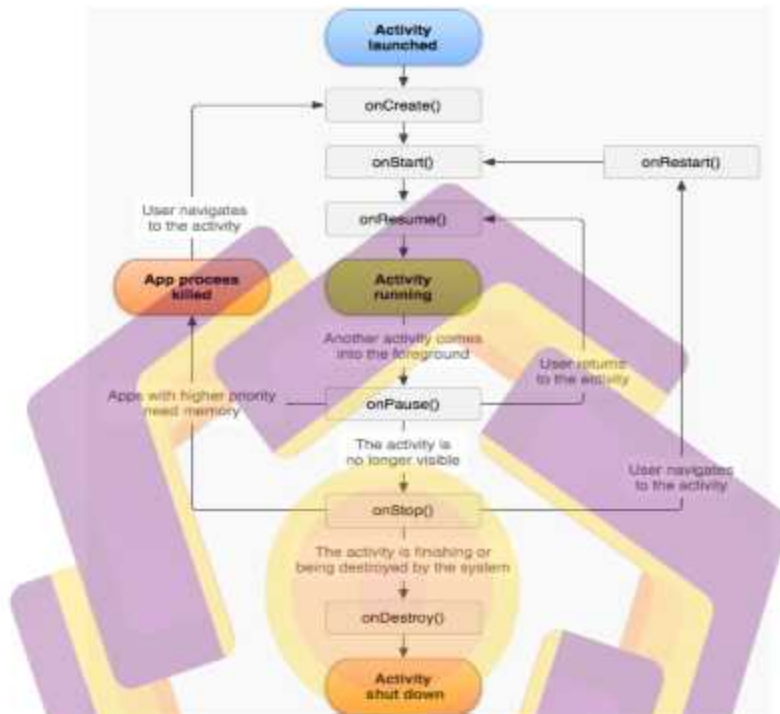
*Content Provider* membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti

*database* SQLite. *Content Provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*.

### 2.6.5 *Activity dan Widget*

*Activity* merupakan *public class* dalam aplikasi Android. Setiap *activity* merupakan sesuatu yang unik yang ditujukan untuk meng-*handle* macam-macam hal yang bisa dilakukan oleh user. Biasanya, *activity* berhubungan dengan user di mana *activity* menciptakan UI yang ditampilkan dengan *concept setContentView(View)*. [17]

*Activity* aplikasi android dikelola dengan sistem yang dikenal dengan *activity stack*. Ketika suatu *activity start*, *activity* diletakkan pada *stack* yang paling atas dan *activity-activity* yang sudah berjalan berada dibawahnya dan akan terus berada pada posisi atas *stack* sampai muncul *activity* yang baru. Berikut ini adalah gambaran mengenai *activity lifecycle* di dalam android sistem :



**Gambar 2.3 Activity Lifecycle[18]**

Widget merupakan visualisasi dari elemen *userinterface* yang digunakan pada layar aplikasi android dimana kita merancang sendiri kebutuhan kita. *Widget* didalam android ditampilkan dengan konsep *View*. Aplikasi android pada umumnya menggunakan *widget* sebagai *layout XML*. [19]

## 2.7 SMS Gateway

SMS Gateway merupakan aplikasi teknologi SMS yang digunakan untuk mengirim, menerima dan mengolah sms melalui komputer.

Pada jaman sekarang ini, hampir setiap orang memiliki telepon selular atau *handphone*. Bahkan ada yang memiliki lebih dari 1 *handphone*. SMS merupakan salah satu fitur yang terdapat pada *handphone* yang pasti digunakan oleh pengguna *handphone*. Selain tarif SMS yang terjangkau bahkan bisa dikatakan murah, selain itu sms juga praktis. Bisa dibaca kapan saja dan dimana saja termasuk pada saat ketika *handphone* kita tidak aktif, pesan dapat kita baca setelah kita mengaktifkan *handphone*.

Bagi perusahaan dan bisnis, teknologi SMS berupa SMS Gateway dapat dimanfaatkan dalam hal pemasaran dan pengumuman terhadap *customer/* pelanggan mereka. Perusahaan dapat memberikan informasi seputar perkembangan produk mereka atau memberikan promo dengan cepat, mudah, murah serta tepat sasaran hanya dengan menggunakan SMS Gateway.

Selain digunakan untuk mengirim SMS secara massal, SMS Gateway dapat juga digunakan sebagai *auto responder*, dimana perusahaan dapat melakukan komunikasi 2 arah dengan *customer/* pelanggan. Fitur ini banyak dimanfaatkan untuk pendaftaran agen pulsa, pendaftaran member MLM, pendaftaran anggota komunitas, program iklan televisi, dll.

Kegunaan lain dari SMS Gateway adalah dapat digunakan sebagai pengumpulan hasil polling seperti Indonesian Idol, Pilkada, dll. Banyak stasiun TV yang menggunakan SMS Gateway untuk mendukung siaran program acara mereka.

SMS Gateway juga dapat digunakan sebagai tambahan fitur pada Tugas akhir atau skripsi bagi mahasiswa jurusan Informatika.

Manfaatkan teknologi SMS yang mutakhir dan canggih ini dengan memiliki Software SMS Gateway yang dapat anda peroleh di [SMSCenterGateway.com](http://SMSCenterGateway.com).

## 2.8 *Software Development Life Cycle (SDLC)*

*System Development Life Cycle (SDLC)* merupakan sebuah metodologi dalam pembangunan atau pengembangan sistem. *System Development Life Cycle* memberikan kerangka kerja yang konsisten terhadap tujuan yang diinginkan dalam pembangunan dan pengembangan sistem. Metodologi SDLC dimulai dengan ide – ide yang berasal dari pengguna, melalui studi kelayakan, analisis dan desain sistem, pemrograman, *pilot testing*, implementasi, dan analisis setelah diimplementasikan (evaluasi).[20] Dokumentasi yang dibuat selama melakukan pembangunan atau pengembangan sistem digunakan untuk perubahan – perubahan di masa yang akan datang, misalnya melanjutkan pengembangan sistem, modifikasi atau penghilangan (*deletion*).

### 2.8.1 **Waterfall Model**

Model ini sama seperti linear sequential model. Waterfall model merupakan model pengembangan perangkat lunak yang paling kuno, tetapi

merupakan model yang paling banyak dipakai didalam Software Engineering. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, *design*, *coding*, *testing*, dan *maintenance*. Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Sebagai contoh tahap *coding* harus menunggu tahap *design* selesai.[21] Secara umum tahapan pada model waterfall dapat dilihat pada gambar berikut :



**Gambar 2.3 Waterfall Model**

1. *Software Requirements Analysis*

Proses ini mencari dan menganalisis kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software*.

2. *Design*

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk “blueprint” *software* sebelum *coding* dimulai.

Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.

### 3. *Coding*

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh programmer.

### 4. *Testing*

Dalam tahap ini dilakukan pengujian *software* yang sudah dibuat. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari error, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

### 5. *Maintenance*

Pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada *errors* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan



dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

**Kelebihan:**

- Dituntut bekerja secara disiplin
- Dokumen lengkap
- Selalu dalam kontrol SQA
- *Maintenance* mudah, karena dokumen lengkap

**Kekurangan:**

- Konsumen kesulitan membaca dokumen, komunikasi menjadi sulit
- Alur linier, proses lambat
- Konsumen tidak dapat melihat hasil hingga akhir tahapan
- Personil tidak bekerja optimal, karena ada waktu tunggu sebuah tahapan selesai.

## 2.9 Konsep UML (*Unified Modelling Language*)

### 2.9.1 Definisi UML

Berikut ini definisi *Unified Modeling Language* (UML) menurut para ahli:

1. Menurut Adi Nugroho (2005).

*Unified Modeling Language* (UML) adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek.

2. Menurut Fowler (2005)

UML adalah keluarga notasi grafis yang didukung oleh model-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi obyek.

3. Menurut Jeffrey L. Whitten L. Jeffery et al (2004)

UML merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan obyek.




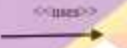
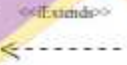


## 2.9.2 Diagram UML

### 2.9.2.1 Use Case Diagram[22]

*Use Case Diagram* adalah diagram yang menggambarkan interaksi antara sistem dengan eksternal sistem dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem, Adapun simbol-simbol dari use case sebagai berikut :

**Table 2.1 Simbol Use Case Diagram**

NO	GAMBAR	NAMA
----	--------	------

1		<i>Actor</i>
2		<i>Use Case</i>
3		<i>Association</i>
4		<i>Uses (Include)</i>
5		<i>Extend</i>
6		<i>Inheritance</i>
7		<i>System Boundary</i>

Sumber: Whitten L. Jeffery et al, 2004

Dalam *use case diagram* memiliki pemodelan sebagai berikut :

1. *Use case*

*Use case* merupakan urutan langkah-langkah yang secara tindakan saling terkait (*scenario*), baik otomatis maupun secara manual.

2. *Actor* (Pelaku)

*Actor* merupakan segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi.

3. *Relationship* (Hubungan)

Pada diagram *use case, relationship* digambarkan sebagai sebuah garis antara dua simbol. Pemaknaan *relationship* berbeda-beda tergantung bagaimana garis tersebut digambar dan tipe simbol apa yang digunakan untuk menghubungkan garis tersebut. Berikut ini adalah perbedaan di antara *relationship* yang ada pada sebuah diagram *use case*:

a. *Association*

*Association* merupakan *relationship* antara *actor* dengan *use case* dimana terjadi interaksi di antara mereka.

b. *Extends*

*Extends use case* merupakan *use case* yang terdiri dari langkah yang terekstraksi dari *use case* yang lebih kompleks untuk menyederhanakan masalah dan karena itu memperluas fungsinya.

c. *Uses (includes)*

Hubungan *uses* menggambarkan bahwa satu *use case* seluruhnya meliputi fungsionalitas dari *use case* lainnya.

d. *Depends on*

Terkadang suatu *use case* memiliki ketergantungan pada *use case* yang lainnya yang bertujuan untuk menentukan urutan dalam pengembangan *use case*. Ketergantungan ini dimodelkan menggunakan *depends on relationship*.








e. *Inheritance*

Hubungan *inheritance* terjadi ketika dua atau lebih *actor* menggunakan *use case* yang sama.

### 2.9.2.2 Activity Diagram[23]

*Activity diagram* secara grafis digunakan untuk menggambarkan rangkaian aliran aktifitas baik proses bisnis atau *usecase*. Adapun simbol-simbol dari *activity diagram* adalah sebagai berikut :

**Table 2.2 Simbol Activity Diagram**






NO	GAMBAR	NAMA
1		<i>State</i>
2		<i>Control Flow</i>
3		<i>Initial State</i>
4		<i>Final State</i>
5		<i>Transition</i>
6		<i>Decision</i>
7		<i>Swimlane</i>

Sumber: Whitten L, Jeffery et al, 2004

### 2.9.2.3 Class Diagram[24]

Class diagram gambar grafis mengenai struktur obyek statis dari suatu sistem, menunjukkan kelas-kelas obyek yang menyusun sebuah sistem dan juga hubungan antara kelas obyek tersebut. Adapun simbol-simbol dari class diagram terdapat pada daftar simbol.

**Table 2.3 Simbol Class Diagram**

No	Simbol	Nama
1		1. <i>Class name</i> 2. <i>Attributes</i> 3. <i>behaviors</i>
2		<i>Association</i>
3		<i>Agregation</i> <i>(composite)</i>
4		<i>Agregation</i> <i>(cshared)</i>
5		<i>Generalization</i>








Sumber: Whitten L. Jeffery et al, 2004

#### 2.9.2.4 Sequence Diagram[25]

*Sequence diagram* secara grafis menggambarkan bagaimana object berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah *usecase*

atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara *object* dan *sequence* (ruang waktu). Adapun simbol-simbol dari *sequence* diagram sebagai berikut :

**Table 2.4 Simbol *Sequence* Diagram**

No	Simbol	Keterangan
1		<i>Object</i>
2		<i>Actor</i>
3		<i>Lifeline</i>
4		<i>Message</i>
5		<i>Message (Return)</i>
6		<i>Message (Call)</i>
7		<i>Activation</i>





Sumber: Whitten L. Jeffery et al, 2004

#### 2.9.2.5 Statechart Diagram[26]

*Statechart* diagram mengilustrasikan siklus hidup *object* dan keadaan yang dapat diasumsikan oleh *object* dan *events* yang menyebabkan *object* beralih

dari satu *state* ke *state* yang lain. Adapun simbol-simbol dari *statechart diagram* sebagai berikut:

**Table 2.5 Statechart Diagram**

No	Simbol	Nama
1		<i>State</i>
2		<i>Initial State</i>
3		<i>Finish State</i>
4		<i>Trasition with event and condition</i>

*Sumber: Mathiasen et al, 2000*

#### 2.9.2.6 Keunggulan UML [27]

UML diterapkan dalam pengembangan sistem/perangkat lunak berorientasi obyek sebab metodologi UML ini umumnya memiliki keunggulan-keunggulan sebagai berikut:

*a. Uniformity*

Pengembang cukup menggunakan 1 metodologi dari tahap analisis hingga perancangan. Perancangan komponen antarmuka terintegrasi dengan perancangan PL dan struktur data.

*b. Understandability*



Kode yang dihasilkan dapat diorganisasi kedalam kelas-kelas yang berhubungan dengan masalah sesungguhnya sehingga lebih mudah untuk dipahami.

c. *Stability*

Kode program yang dihasilkan relatif stabil sepanjang waktu.

d. *Reusability*

Dengan metodologi berorientasi obyek, dimungkinkan penggunaan ulang kode, sehingga pada gilirannya akan sangat mempercepat waktu pengembangan perangkat lunak.

## 2.10 Bahasa Pemrograman

### 2.10.1 Java

Java merupakan bahasa pemrograman yang dikembangkan dari bahasa pemrograman C++. Berawal dari proyek penelitian perusahaan Sun Microsystems dengan nama sandi *green* pada tahun 1991. Terdapat prediksi bahwa mikroprosesor akan digunakan luas pada peralatan-peralatan elektronik. Karena adanya bermacam tipe mikroprosesor, maka dibutuhkan sebuah bahasa pemrograman yang dapat berjalan di semua mikroprosesor. Terciptalah sebuah bahasa pemrograman baru oleh James Gosling yaitu salah satu orang yang berperan besar dalam proyek tersebut, program ini diberi nama *Oak*. Sesuai dengan pohon Oak yang tumbuh dan bisa dilihat melalui jendela kerjanya ddengan pohon Oak yang tumbuh dan bisa dilihat melalui jendela kerjanya di *Sun Microsystems*.

Selang beberapa waktu kemudian, ditemukan bahwa sudah ada bahasa pemrograman dengan nama Oak. Akhirnya setelah beberapa pegawai Sun

mengunjungi sebuah kedai kopi nama bahasa pemrograman ini diganti dengan *Java*. *Java* merupakan salah satu jenis biji kopi yang ada di kedai tersebut yaitu biji kopi jawa. *Sun Microsystems* mengumumkan kehadiran bahasa *Java* secara formal di tahun 1995. Bahasa ini mulai disambut hangat masyarakat luas seiring dengan meledaknya era *internet*. [28]

### 2.10.2 Konsep OOP (*Object Oriented Programming*) [29]

OOP merupakan cara berpikir, pandangan atau paradigma baru untuk membuat program atau merancang sistem dengan memerhatikan objek; ciri objek dan perilakunya, OOP ini sangat berbeda dengan membuat program prosedural yang memerhatikan aspek input, proses dan output.

Coad-Yourdon merumuskan pendekatan berorientasi objek dengan persamaan :

*Berorientasi objek = objek + klasifikasi + pewarisan + komunikasi/pesan*

Beberapa konsep berorientasi yang penting adalah :

1. Pengkapsulan (*encapsulation*)
2. Pewarisan (*inheritance*)
3. *Polymorphism*
4. Pesan

## 2.11 Basis Data (*Database*)

### 2.11.1 Definisi Basis Data [30]

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol). Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan file atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Selain berisi data, *database* juga berisi meta data. Meta data adalah data yang menjelaskan tentang struktur dari data itu sendiri. Data nama kolom dan tipe yang ditampilkan tersebut disebut metadata.

### 2.11.2 Tujuan Database[31]

1. Kecepatan dan kemudahan (*Speed*)

Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data, melakukan perubahan, manipulasi terhadap data, menampilkan kembali data tersebut dengan lebih cepat dan mudah. Dari pada kita menyimpan data secara manual.

## 2. Efisiensi ruang penyimpanan (*Space*)

Dengan basis data, optimisasi penggunaan ruang penyimpanan dapat dilakukan karena kita dapat melakukan penekanan jumlah redundansi data, baik menerapkan sejumlah pengkodean atau membuat relasi-relasi (dalam bentuk file) antar kelompok data yang saling berhubungan.

## 3. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan / batasan (*constraint*) tipe data, domain data, keunikan data dan sebagainya yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan dalam penyimpanan data.

## 4. Ketersediaan (*Availability*)

Dengan pemanfaatan teknologi jaringan komputer, data yang berada di suatu lokasi dapat diakses (*available*) dari lokasi yang lain.

## 5. Kelengkapan (*Completeness*)

Untuk mengakomodasi kebutuhan dan kelengkapan data yang semakin berkembang, maka tidak hanya dapat menambah record-record data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam penambahan objek baru (tabel) atau menambahkan *field* baru pada tabel.

#### 6. Keamanan (*Security*)

Kita dapat menentukan user yang boleh menggunakan basis data beserta objek-objek didalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukan.

#### 7. Kebersamaan pemakaian (*Sharebility*)

Data yang dapat digunakan oleh banyak pemakai, dari sejumlah departement dalam perusahaan atau banyak sistem, basis data yang dikelola oleh sistem (aplikasi) yang mendukung multiuser akan dapat memenuhi kebutuhan ini.

### 2.11.3 *Database Management System (DBMS)*[32]

*Database Management System* adalah suatu sistem piranti lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, serta mengatur akses terhadap basis data.

*Database Management System* adalah suatu sistem perangkat lunak yang digunakan untuk membuat, merawat, dan menyediakan pengontrolan akses pada pengguna basis data.

DBMS berperan sebagai antara *user* dengan basis data. DBMS membantu *user* mengakses data dalam basis data.

### 2.11.4 Fasilitas DBMS[33]

DBMS menyediakan beberapa fasilitas sebagai berikut :

#### 1. *Data Definition Language (DDL)*

*Data Definition Language* memperbolehkan pemakai untuk membuat spesifikasi tipe data, mendefinisikan basis data, struktur data dan constraint data untuk disimpan dalam basis data.

*Data Definition Language* adalah suatu bahasa yang memungkinkan administrator basis data atau pemakai untuk mendeskripsikan dan memberi nama entitas-entitas, atribut-atribut dan hubungan yang diperlukan untuk aplikasi, bersama dengan integritas terkait dan batasan keamanannya.

*Data Definition Language* adalah suatu bahasa yang memungkinkan administrator basis data untuk menentukan struktur basis data, skema, komponen-komponen subskema.

## 2. *Data Manipulation Language (DML)*

*Data Manipulation Language* memperbolehkan pemakai untuk memasukan, memperbaharui, menghapus dan mengirim atau mengambil data dari basis data.

*Data Manipulation Language* adalah suatu bahasa yang menyediakan sel operasi untuk mendukung pengoperasian dan manipulasi data yang terdapat dalam basis data.

*Data Manipulation Language* adalah suatu bahasa (kumpulan *command*) yang memperbolehkan pengguna akhir untuk memanipulasi data dalam basis data.

3. Menyediakan fasilitas untuk mengontrol akses ke basis data:
  - a. Sistem keamanan (*Security system*), mencegah *user* yang tidak berwenang agar tidak mengakses ke basis data.
  - b. Sistem integritas (*Integrity system*), menjaga konsistensi data yang disimpan.
  - c. Sistem kontrol (*Concurency control*), mengijinkan agar data dapat dipakai bersama-sama oleh *user* lainnya.
  - d. Sistem kontrol perbaikan (*Recovery control system*), memperbaiki atau mengembalikan basis data ke kondisi sebelumnya jika terjadi kerusakan pada perangkat keras dan perangkat lunak.
  - e. Katalog yang dapat diakses *user* (*User-accessible catalog*), catatan yang berisi deskripsi data pada basis data.

#### 2.11.5 SQLite

SQLite adalah sebuah *embedded database* yang sangat terkenal karena menggabungkan antarmuka SQL dengan memori yang sangat kecil dan kecepatan yang baik : Murphy (2010, p225). SQLite adalah sebuah *open source database* yang telah ada cukup lama, cukup stabil, dan sangat terkenal pada perangkat kecil, termasuk Android : Gargenta (2011, p119) .

Android menyediakan *database* relasional yang ringan untuk setiap aplikasi menggunakan SQLite : Meier (2010, p7). Aplikasi dapat mengambil

keuntungan dari itu untuk mengatur *relational database engine* untuk menyimpan data secara aman dan efisien. Murphy (2010, p225) Untuk Android, SQLite dijadikan satu di dalam Android runtime, sehingga setiap aplikasi Android dapat membuat basis data SQLite. Karena SQLite menggunakan antarmuka SQL, cukup mudah untuk digunakan orang-orang dengan pengalaman lain yang berbasis *databases*.

Terdapat beberapa alasan mengapa SQLite sangat cocok untuk pengembangan aplikasi Android, yaitu:

1. *Database* dengan konfigurasi nol. Artinya tidak ada konfigurasi *database* untuk para *developer*. Ini membuatnya relatif mudah digunakan.
2. Tidak memiliki server. Tidak ada proses *database* SQLite yang berjalan. Pada dasarnya satu set *libraries* menyediakan fungsionalitas *database*.
3. *Single-file database*. Ini membuat keamanan *database* secara langsung.
4. *Open source*. Hal ini membuat *developer* mudah dalam pengembangan aplikasi.

## **2.12 Perangkat Lunak Yang Digunakan**

### **2.12.1 IDE Eclipse[34]**

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platformindependent*).

Berikut ini adalah sifat dari Eclipse:



1. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. *Multi-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya.

Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan *open source*. *Open source* berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan membuat komponen yang disebut *plugin*.

### 2.12.2 Android SDK (*Software Development Kit*)

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*)

sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi netral Android memberi anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone* atau *Smartphone*. [35]

### 2.12.3 ADT (*Android Development Tools*)

Menurut Nazruddin Safaat H (2011) ADT adalah kepanjangan dari *Android Development Tools* yang menjadi penghubung antara IDE Eclipse dengan *Android SDK*. ADT atau lebih di kenal dengan plugin Eclipse, plugins ini yang dapat membuat eclipse dapat membuat project yang berbasis Android.

### 2.12.4 Gammu[36]

Gammu adalah sebuah aplikasi *cross-platform* yang digunakan untuk menjembatani / mengomunikasikan antara *databaseSMS Gateway* dengan *sms devices*. Aplikasi Gammu berupa *daemon* yang berjalan secara *background*. Setiap saat, gammu memonitor *sms devices* dan *databasesms gateway*. Saat ada sms masuk ke *sms devices*, maka gammu langsung memindahkannya ke dalam *inbox* dalam *database sms gateway*. Sebaliknya saat Aplikasi Pengirim SMS memasukkan sms ke dalam *outbox* dalam *database sms gateway*, maka gammu mengirimkannya melalui *sms devices*, dan memindahkan sms ke sentitem dalam *database*.

### 2.12.5 WampServer[37]

WampServer adalah singkatan dari Windows, Apache, MySQL dan PHP. WampServer adalah aplikasi yang menggabungkan antara Apache, MySQL, dan PHP. Sehingga, WampServer akan mencukupi semua persyaratan yang diminta oleh Joomla. Dengan demikian, tidak perlu lagi meng-install masing – masing aplikasi apache, mysql, dan php.

#### 2.12.6 Notepad++

Notepad++ adalah sebuah *text editor* yang sangat berguna bagi setiap orang dan khususnya bagi para *developer* dalam membuat program. Notepad++ menggunakan komponen Scintilla untuk dapat menampilkan dan menyuntingan teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan diatas sistem operasi Microsoft Windows.[38]

Selain manfaat dan kemampuannya menangani banyak bahasa pemrograman, Notepad++ juga dilisensikan sebagai perangkat free. Jadi, setiap orang yang menggunakannya tidak perlu mengeluarkan biaya untuk membeli aplikasi ini karena sourceforge.net sebagai layanan yang memfasilitasi Notepad++ membebaskannya untuk digunakan. Beberapa daftar bahasa program yang didukung oleh Notepad++ adalah C, C++, Java, C#, XML, HTML, PHP, Javascript, Dll.