BAB 2
TINJAUAN PUSTAKA

2.1. Tinjauan Perusahaan

2.1.1. Sejarah Instansi

Ini merupakan penjelasan dari sejarah terbentuknya perusahan PT. POS Indonesia (Persero) dari awal terbentuknya sampai saat ini yaitu sebagai berikut :

Tabel 2.1. Sejarah Instansi

Tahun	Uraian
26-81746	Kantor Pos pertama Indonesia adalah di Batavia yang didirikan oleh Gubernur Jendral GW Baron
1906	Posts Telegraafend Telefon Diensts
27-9-1945	Jawatan PTT Republik Indonesia ditandai Pengambilalihan Kantor Pusat PTT di Bandung oleh Angkatan Muda PTT dari pemerintahan Militer Jepang. Tanggal tersebut diperingati sebagai Hari Bakti Postel.
1961	Berdasarkan Peraturan Pemerintahan No. 240 Tahun 1961 status Jawatan PTT berubah menjadi Perusahaan Negara (PN) Pos dan Telekomunikasi.
1965	PN Pos dan Telekomunikasi dibagi dua menjadi : PN Pos dan Giro berdasarkan Peraturan Pemerintah No. 29 Tahun 1965 dan PN Telekomunikasi berdasarkan Peraturan Pemerintah No. 30 Tahun 1965.
1978	Berdasarkan Peraturan Pemerintah No. 9 Tahun 19678, status PN Pos dan Giro diubah menjadi Perusahaan Umum (Perum) Pos dan Giro.
20-6-1995	Dasar Hukum : Undang-undang No. 1 Tahun 1995 tentang Perusahaan Perseroan; Peraturan Pemerintah RI No. 5 Tahun 1995 tentang Pengalihan Bentuk Perusahaan Umum (Perum) Pos dan Giro menjadi Perusajaan (Persero) (Lembaran Negara RI Tahun 1995 No. 11); Anggaran Dasar PT. Pos Indonesia (Persero) yang tercantum dalam akta Notaris Sutjipto, SH No. 117 tanggal 20 Juni 1995 tentang Pendirian Perusahaan Persero PT Pos Indonesia,

	sebagaimana telah diubah dengan akta Notaris Sutjipto, SH No. 89 tanggal 21 September 1998 dan No. 111 tanggal 28 Oktober 1998

2.1.2. Logo Instansi



Gambar 2.1. Logo Lama

Logo lama Perum Pos dan Giro terdiri dari unsur padi kapas kapas yang bersambung dengan banner di atas dengan tulisan RI, banner di bawah dengan tulisan Pos dan giro, mengelilingi unsur segi lima yang mengurung bola dunia dan burung. Diantara segilima dan padi kapas terdapat garis arsiran horizontal yang berfungsi sebagai *background* dari unsur segilima.



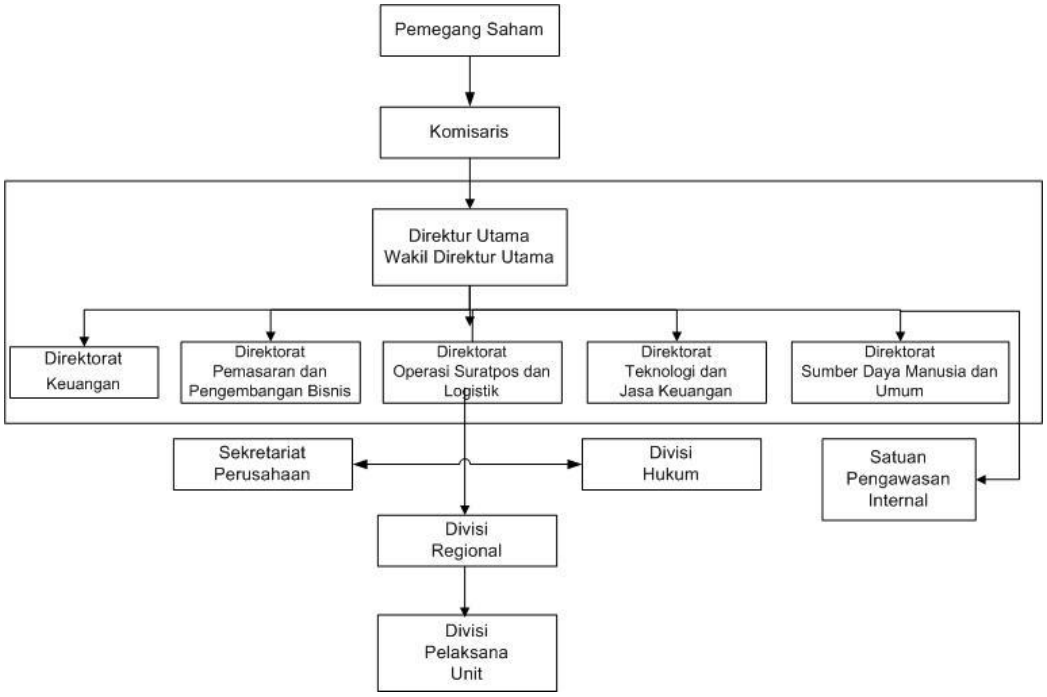
Gambar 2.2. Logo Baru

Ukuran burung dibuat lebih besar dibandingkan dengan bola dunia, dapat dimaknai bahwa burung sebagai tanda mempersentasikan PT Pos diharapkan dapat menguasai dunia. Logo meiliki warna dasar jingga untuk menandakan sesuatu yang penting. Warna ini juga digunakan untuk tiang-tiang pemisah pada perbaikan di jalan tol, seragam tukang parkir, pakaian penerbang, pakaian pendaki gunung, dan segala sesuatu yang dianggap penting dan perlu perhatian.

2.1.3. Struktur Organisasi

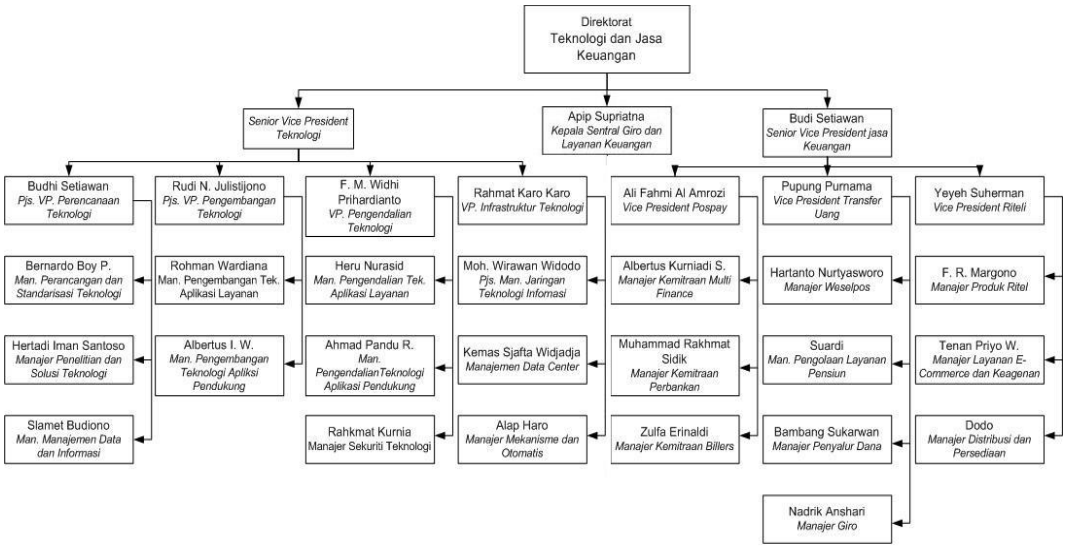
Struktur Organisai menjelaskan struktur organisasi yang ada di dalam PT. Pos Indonesia.

2.1.3.1. Struktur Organisasi PT. POS Indonesia (Persero).



Gambar 2.3. Organisasi PT. POS Indonesia (Persero)

2.1.3.2. Struktur Organisasi Direktorat Teknologi dan Jasa Keuangan PT. POS Indonesia (Persero).



Gambar 2.4. Organisasi Direktorat Teknologi dan Jasa Keuangan PT. POS Indonesia (Persero)

2.1.3.3. *Job Descriptions*

Deskripsi jabatan (*job description*) masing-masing bagian pada struktur organisasi di PT. POS Indonesia Bandung adalah sebagai berikut :

1. **VP. Infrastruktur Teknologi**

   Bertanggung jawab dalam pengelolaan, penjaminan, pengendalian guna mengembangkan arah strategis sistem teknologi khususnya dalam bidang Jaringan Komunikasi Data dan Data *Center* sesuai kebutuhan Perusahaan dalam rangka meningkatkan daya saing Perusahaan.

2. **Manajar Pengelola Jaringan Teknologi.**

   Bertanggung jawab dalam penyediaan solusi jaringan :

   a. *Wide Area Network* (WAN) yang berkualitas, aman, efektif dan efisien untuk menghubungkan UPT yang ada guna mendukung operasional Perusahaan.

b. *Lokal Area Network* (LAN) yang berkualitas, aman, efektif dan efisien untuk mendukung operasional UPT.

3. RS. Manajer Data *Center*

Bertanggung jawab dalam penyediaan sarana *hardware* berupa *server* dan peralatan pendukungnya dan *software*, yang berkualitas, aman, efektif dan efisien untuk mendukung operasional Perusahaan dan menjaga keamanan peralatan Data *Center* baik secara fisik maupun virtual.

2.1.3.4. Visi dan Misi

Adapun visi dan misi yang PT. POS Indonesia jalankan guna untuk kemajuan perusahaan, diantaranya sebagai berikut :

2.1.3.4.1. Visi

2009 - 2010 : *Integrated mail, logistic & financial services infrastructure*
2011 - 2013 : *Indonesia's leader in the mail logistics & financial services*
2014 - 2018 : *ASEAN Champion of Postal Industries*

2.1.3.4.2. Misi

POS Indonesia menyediakan solusi handal dalam *mail*, logistik dan jasa keuangan dengan menggunakan jejaring bisnis dan infrastruktur terluas dan terpadu serta mengembangkan hubungan kolaboratif.

2.2. Landasan Teori

Landasan teori menjelaskan beberapa teori yang berkaitan dengan permasalahan yang dibahas sebagai dasar pemahaman dalam sebuah sistem serta metode yang dipakai untuk kegiatan pengembangan terhadap sistem sendiri.

2.2.1. Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas

yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.2.2. Basis Data

2.2.2.1. Pengertian Basis Data

Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

Adpun manfaat dari basis data antara lain :

1. Kecepatan dan Kemudahan (*Speed*)

   Yaitu agar pengguna basis data bisa menyimpan data, melakukan perubahan/manipulasi terhadap data dan menampilkan kembali data dengan lebih cepat dan mudah dibandingkan dengan cara biasa (baik manual maupun elektronis).

2. Efisiensi Ruang Penyimpanan (*Space*)

   Dengan basis data kita mampu melakukan penekanan jumlah redundansi (pengulangan) data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi antara kelompok data yang saling berhubungan.

3. Keakuratan (*Accuracy*)

   Agar data sesuai dengan aturan dan batasan tertentu dengan cara memanfaatkan pengkodean atau pembentukan relasi antara data bersama dengan penerapan aturan/batasam (*constraint*) tipe data, domain, keunikan dsb.

4. Ketersediaan (*Availability*)

   Agar data bisa diakses oleh setiap pengguna yang membutuhkan, dengan penerapan teknologi jaringan serta melakukan pemimdahan/penghapusan data yang sudah tidak digunakan/kadaluwarsa untuk menghemat ruang penyimpanan.

5. Kelengkapan (*Completeness*)

   Agar data yang dikelola senantiasa lengkap baik relatif terhadap kebutuhan pemakai maupun terhadap waktu, dengan melakukan penambahan baris-baris data ataupun melakukan perubahan struktur pada basis data, yakni dengan menambahkan *field* pada tabel atau menambah tabel baru.

6. Keamanan (*Security*)

   Agar data yang bersifat rahasia atau proses yang vital tidak jatuh ke orang/pengguna yang tidak berhak, yakni dengan penggunaan *account* (*usernmae* dan *password*) serta menerapkan pembedaan hak akses setiap pengguna terhadap data yang bisa dibaca atau proses yang bisa dilakukan.

7. Kebersamaan (*Sharability*)

   Agar data yang dikelola oleh sistem mendukung lingkungan miltiuser (banyak pemakai), dengan menjaga/menghindari munculnya problem baru seperti inkonsistensi data (karena terjadi perbahan data yang dilakukan oleh beberapauser dalam waktu yang bersamaan) atau kondisi *deadlock* (karena ada banyak pemakai yang saling nunggu untuk menggunakan data).

Di dalam basis data terdapat beberapa istilah berikut yang juga dikenal sebagai urutan jenjang data :

1. *Character*

   Merupakan bagian data yang terkecil yaitu berupa karakter numerik, huruf, maupun karakter spesial (simbol) yang membentuk item data/*field*.

2. *Field*

   Merepresentasikan suatu atribut dari *record* yang menunjukan suatu item dari data, misalnya nama.

3. *Record*

   Merupakan kumpulan dari *field*, menggambarkan suatu unit data invidu tertentu. Kumpulan record membentuk suatu *file*. Misalnya, *file* personalia, tiap-tiap *record* mewakili data tiap-tiap karyawan.

4. *File*

   File terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Misalnya, *file* yang berisi data tentang semua mata kuliah yang ada.

5. *Database*

Merupakan kumpulan dari *file*/tabel saling berelasi antara tabel satu dengan tabel lainnya.

2.2.2.2. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem *software*.

a. Artifact UML

UML menyediakan beberapa notasi dan *artifact* standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. *Artifact* didalam UML didefinisikan sebagai informasi dalam bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat. Contohnya adalah *source code* yang dihasilkan oleh proses pemrograman.

Yang harus diperhatikan untuk menjaga konsistensi antar *artifact* selama proses analisis dan desain adalah bahwa setiap perubahan yang terjadi pada satu *artifact* harus juga dilakukan pada *atifact* sebelumnya.

Untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut :

1. *Use Case Diagram*
2. *Class Diagram*
3. *Statechart Diagram*
4. *Activity Diagram*

5. *Sequence Diagram*
6. *Collaboration Diagram*
7. *Component Diagram*
8. *Deployment Diagram*

Dibuatnya berbagai jenis diagram diatas karena :

1. Setiap sistem yang kompleks selalu paling baik jika didekati melalui himpunan berbagai sudut pandang yang kecil yang satu sama lain hampir saling bebas (*independent*). Sudut pandang tunggal senantiasa tidak mencukupi untuk melihat sistem yang besar dan kompleks.
2. Diagram yang berbeda-beda tersebut dapat menyatakan tingkatan yang berbeda-beda dalam proses rekayasa.
3. Diagram-diagram tersebut dibuat agar model yang dibuat semakin mendekati realitas.

b. Tujuan UML

Tujuan utama UML diantaranya untuk :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam bahasa pemodelan.

c. Notasi dalam UML

1. Actor

Actor menggambarkan segala pengguna *software* aplikasi (*user*). *Actor* memberikan suatu gambaran jelas tentang apa yang harus dikerjakan *software* aplikasi. Sebagai contoh sebuah *actor* dapat memberikan *input* ke dalam dan menerima informasi dari *software* aplikasi, perlu dicatat bahwa sebuah *actor* berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Sebuah *actor* mungkin seorang manusia, satu *device, hardware* atau sistem informasi lainnya.

2. Use Case

Use case menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, namun *use case* hanya menjelaskan apa yang dilakukan oleh *actor* dan sistem bukan bagaimana *actor* dan sistem melakukan kegiatan tersebut.

a. *Use-case Konkret* adalah *use case* yang dibuat langsung karena keperluan *actor*. *Actor* dapat melihat dan berinisiatif terhadapnya.

b. *Use-case Abstrak* adalah *use case* yang tidak pernah berdiri sendiri. *Use case abstrak* senantiasa termasuk didalam (*include*), diperluas dari (*extend*) atau memperumum (*generalize*) *use case* lainnya.

Untuk menggambarkannya dalam *use case* model biasanya digunakan *association relationship* yang memiliki *stereotype include, extend* atau *generalization relationship*. Hubungan *include* menggambarkan bahwa suatu *use case* seluruhnya meliputi fungsionalitas dari *use case* lainnya. Hubungan *extend* antar *use case* berarti bahwa satu *use case* merupakan tambahan fungsionalitas dari *use case* yang lain jika kondisi atau syarat tertentu terpenuhi.

3. *Class*

Class merupakan pembentuk utama dari sistem berorientasi *obyek*, karena *class* menunjukkan kumpulan *obyek* yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan *interface.*

Class digunakan untuk mengabstraksikan elemen-elemen dari sistem yang sedang dibangun. *Class* bisa merepresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata.

Notasi *class* berbentuk persegi panjang berisi 3 bagian: persegi panjang paling atas untuk nama *class*, persegi panjang paling bawah untuk operasi, dan persegi panjang ditengah untuk *atribut*.

Atribut digunakan untuk menyimpan informasi. Nama *atribut* menggunakan kata benda yang bisa dengan jelas merepresentasikan informasi yang tersimpan didalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh *obyek* dan menggunakan kata kerja.

4. *Interface*

Interface merupakan kumpulan operasi tanpa implementasi dari suatu *class*. Implementasi operasi dalam *interface* dijabarkan oleh operasi di dalam *class*. Oleh karena itu keberadaan *interface* selalu disertai oleh *class* yang mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan prinsip *enkapsulasi* dalam *obyek*.

5. *Interaction*

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar *obyek* maupun hubungan antar *obyek*. Biasanya *interaction* ini dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.

6. *Note*

Note digunakan untuk memberikan keterangan atau komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa disertakan ke semua elemen *notasi* yang lain.

7. *Depedency*

Dependency merupakan *relasi* yang menunjukan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada dibagian tanpa tanda panah.

Terdapat 2 *stereotype* dari *dependency*, yaitu *include* dan *extend*. *Include* menunjukkan bahwa suatu bagian dari elemen (yang ada digaris tanpa panah) memicu eksekusi bagian dari elemen lain (yang ada di garis dengan panah). *Extend* menunjukkan bahwa suatu bagian dari elemen di garis tanpa panah bisa disisipkan kedalam elemen yang ada di garis dengan panah.

8. *Association*

Association menggambarkan navigasi antar *class* (*navigation*), berapa banyak *obyek* lain yang bisa berhubungan dengan satu *obyek* (*multiplicity* antar *class*) dan apakah suatu *class* menjadi bagian dari *class* lainnya (*aggregation*).

Navigation dilambangkan dengan penambahan tanda panah di akhir garis. *Bidirectional navigation* menunjukkan bahwa dengan mengetahui salah satu *class* bisa didapatkan informasi dari *class* lainnya. Sementara *Uni Directional navigation* hanya dengan mengetahui *class* diujung garis *association* tanpa panah.

Aggregation mengacu pada hubungan "*has-a*", yaitu bahwa suatu *class* memiliki *class* lain, misalnya Rumah memiliki *class* Kamar.

9. *Generalization*

Generalization menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan *generalization, class* yang lebih spesifik (*subclass*) akan menurunkan *atribut* dan operasi dari *class* yang lebih umum (*superclass*) atau "*subclass is superclass*". Dengan menggunakan *notasi generalization* ini, konsep *inheritance* dari prinsip *hirarki* dapat dimodelkan.

10. *Realization*

Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah. Misalnya *class* merealisasikan *package*, *component* merealisasikan *class* atau *interface.*

2.2.2.3. Sistem Manajemen Basis Data

Sistem Manajemen Basis Data adalah perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar, serta menjadi alternatif penggunaan secara khusus untuk aplikasi, seperti penyimpanan data dalam *file* dan menulis kode aplikasi yang spesifik untuk pengaturannya. Adapun keuntungan dari DBMS (*Database Management System*) adalah :

1. *Data independence*, DBMS dapat mengurangi kebergantungan antar data.
2. *Efficient data access,* DBMS dapat menimpa data dan *merettive* data dengan efisien. Apabila suatu saat data tersebut dibutuhkan, maka efektif dalam pemanggilan.
3. *Data intergrity dan security,* saat diakses menggunakan DBMS, DBMS mengintergrasi *constraints* dari tiap data tersebut.
4. *Data admnistration,* administrasi data digunakan untuk mengatur hal-hal yang berhubungan dengan distribusi data, hak akses user yang mendistribusikan atau menggunakan data tersebut, mengorganisasikan

data untuk mengurangi redudansi data dan *fine-tuning* penyimpanan data sehingga *retive* data berjalan dengan efisien.

5. *Concurrent access and crash recovry*, DBMS dapat menjadwalkan *concurrent access* (pengaksesan secara bersamaan) terhadap suatu data sehingga menghindari dari *crash*.
6. *Reduced application deveploment time*, DBMS mendukung banyak fungsi penting yang diakses oleh banyak aplikasi yang tersimpan dalam media penyimpanan.

2.2.2.4. Sistem Manajemen Basis Data Relasional

RDBMS (*Relational Database Management System*) merupakan sekumpulan data yang saling beruhubungan yang disimpan sedemikian rupa sehingga mudah diambil informasinya bagi pengguna. Ada tiga prinsip dalam RDBMS :

1. *Data Definition*

   Mendefinisikan data yang akan dibuat, seperti nama tabel dan *field* serta keterangan mengenai *field* tersebut.

2. *Data Manipulation*

   Memanipulasi data yang sudah dibuat, seperti mengubah nama *field* dan keterangannya, menghapus *record*.

3. *Data Control*

   Bagian ini berkenaan pada mengendalikan data kepada siapa saja yang bisa melihat isi data.

2.2.2.5. Android

Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc*. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras,

peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

2.2.2.5.1. Versi Awal Android

Sekitar September 2007 sebuah studi melaporkan bahwa *Google* mengajukan hak paten aplikasi telepon seluler (akhirnya *Google* mengenalkan *Nexus One*, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC *Corporation* dan tersedia di pasaran pada 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM *Holdings, Atheros Communications*, diproduksi oleh *Asustek Computer Inc*, *Garmin Ltd*, *Softbank*, *Sony Ericsson*, *Toshiba Corp*, dan *Vodafone Group Plc*. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*mobile*) yang merupakan modifikasi kernel *Linux* 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Setelah itu Android banyak dikembangkan kembali seperti antara lain :

a. 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*.

b. 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.

c. 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.

d. Android versi 2.0/2.1 (*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.

e. 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, intergrasi 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.

f. 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.

g. Android versi 3.0/3.1 (*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*. Perangkat tablet dengan *platform* Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama *Eee Pad Transformer* produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

h. Android versi 4.0 (*Ice Cream*)

Diumumkan 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.

Berikut ini merupakan versi Android pada saat ini :

a. *Eclair* (2.0 / 2.1)

Versi Android awal yang mulai dipakai oleh banyak *smartphone*, fitur utama *Eclair* yaitu perubahan total struktur dan tampilan *user interface* dan merupakan versi Android yang pertama kali mendukung format HTML5.

b. *Froyo / Frozen Yogurt* (2.2)

Android 2.2 dirilis dengan 20 fitur baru, antara lain peningkatan kecepatan, fitur *Wi-Fi hotspot tethering* dan dukungan terhadap *Adobe Flash*.

c. *Gingerbread* (2.3)

Perubahan utama di versi 2.3 ini termasuk update UI, peningkatan fitur *soft keyboard* & *copy/paste, power management*, dan support *Near Field Communication.*

d. *Honeycomb* (3.0, 3.1 dan 3.2)

Merupakan versi Android yang ditujukan untuk *gadget* / *device* dengan layar besar seperti Tablet PC, fitur baru *Honeycomb* yaitu dukungan terhadap prosessor *multicore* dan grafis dengan *hardware acceleration*. Tablet pertama yang memakai *Honeycomb* adalah tablet *Motorola Xoom* yang dirilis bulan Februari 2011.

e. *Ice Cream Sandwich* (4.0)

Anroid 4.0 Ice Cream Sandwich diumumkan pada 10 Mei 2011 di ajang *Google I/O Developer Conference* (San Francisco) dan resmi dirilis pada tanggal 19 Oktober 2011 di Hongkong. "*Android Ice Cream Sandwich*" akan dapat digunakan baik di *smartphone* ataupun tablet. Fitur utama yang ditambahkan di Android 4.0 ialah *Face Unlock, Android Beam*, perubahan major *User Interface*, dan ukuran layar standar (*native screen*) beresolusi 720p *(high definition)*.

2.2.2.5.2. Kelebihan Android

1. Multitasking –Gadget Android boleh menjanakan pelbaggai aplikasi dalam suatu masa.
2. Kemudahan dalam Notifikasi – Setiap ada SMS, Email, atau bahkan artikel terbaru dari RSS Reader, akan selalu ada notifikasi di Home Screen Ponsel Android.
3. Akses Mudah terhadap Ribuan Aplikasi Android lewat Google Android App Market – Kalau Anda hobi install aplikasi ataupun games, lewat Google Android App Market, yang mana anda boleh mendownload pelbagai aplikasi system android.

4. Sehingga kehari ini pelbagai jenama boleh di dapati dipasaran yang menggunakan system android ini. Harga juga sangat berpatutan mengikut kecanggihan gadget.
5. Boleh menginstall ROM boleh diubahsuaikan mengikut penampilan yang anda inginkan – jika anda tidak puas dengan tampilan standard Android.
6. Terdapat Widget –dengan adanya Widget di homescreen,yang mana boleh mengorientasi pelbagai aplikasi di homescreen.
7. Google Maniac – Jika Anda pengguna setia layanan Google mulai dari Gmail sampai Google Reader, Gadget Android telah berintegrasikan dengan layanan Google, sehingga boleh melihat email dan blog anda.

2.2.2.5.3. Kekurangan Android

1. Kebanyakan handphone Android menggunakan input *touchscreen* yang kurang populer di Indonesia.
2. Android membutuhkan koneksi internet yang sangat cepat untuk memaksimalkan kegunaannya padahal *Internet* dari *Operator* selular Indonesia kurang dapat diandalkan.
3. Dan yang terakhir anggapan bahwa Android sulit untuk dioperasikan / dipakai bila dibandingkan dengan handphone lain macam Nokia atau Blackberry.

2.2.2.6. MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasinya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata nontransaksional. Pada modus operasi nontransaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun demikian pada modus nontransaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus nontransaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (*wordpress*), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus nontransaksional.

2.2.2.6.1. Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain :

1. **Portabilitas**. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. **Perangkat lunak sumber terbuka**. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, di bawah lisensi GPL sehingga dapat digunakan secara gratis.
3. **Multiuser**. MySQL dapat digunakan oleh beberapa oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masslah atau konflik.
4. **Performa tuning**. MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih SQL persatuan waktu.

5. **Ragam tipe data**. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed/unsigned integer, float, double, char, text, date, timestamp,* dan lain-lain.
6. **Perintah dan fungsi**. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. **Keamanan**. MySQL memiliki operator beberapa lapisan keamanan seperti level subnermask, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendatail serta sandi terenkripsi.
8. **Skalabilitas dan Pembatasan**. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. **Konektivitas**. MySQL dapat melakukan koneksi dengan *clients* menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. **Lokalisasi**. MySQL dapat mendeteksi pesan kesalahan pada *clients* dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. **Antar Muka**. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. **Clients dan Peralatan**. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. **Struktur tabel**. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.