BAB 2
TINJAUAN PUSTAKA

2.1 Profil PT. Dian Kencana Puri Prima (DKPP)

PT. DIAN KENCANA PURI PRIMA atau DKPP merupakan distributor resmi Indosat yang siap mendistribusikan voucher elektronik, kartu perdana serta voucher fisik ke seluruh outlet penjual pulsa di wilayah binaannya. Selain mendistribusikan, DKPP juga memberikan pelayanan ke seluruh outlet hingga outlet terkecil di pelosok.

Berbekal pengalaman yang cukup lama dalam bidang distribusi voucher elektronik kini DKPP memiliki jaringan outlet binaan lebih dari 20.000 outlet yang tersebar di kota-kota di Jawa Barat dan Jabodetabek. Konsep "*Simple Business Solution*" yang ditawarkan DKPP sangat menarik outlet untuk bergabung karena kemudahan menjalankan bisnis ini dan dapat mengantarkan mitra menjadi pengusaha sukses di bidang voucher elektronik.

Bisnis voucher elektronik sangat prospektif dan menguntungkan, pasalnya kebutuhan untuk berkomunikasi dan mengakses informasi dan data melalui ponsel telah menjadi kebutuhan utama, sehingga kebutuhan akan voucher elektronik pun menjadi sangat penting. Untuk itu, DKPP membuka peluang berbisnis kepada calon mitra untuk bergabung dalam penjualan voucher elektronik, voucher fisik dan kartu perdana dari Indosat.

2.1.1 Sejarah Singkat PT. DKPP

PT Dian Kencana Puri Prima (DKPP) didirikan pertama kali pada tahun 1997 di Jalan Panjang No.5B Kebon Jeruk Jakarta oleh Hj Dian Hariani. Perusahaan pada awalnya merupakan outlet yang menjual pulsa *all operator* langsung kepada pelanggan.

Perkembangan bisnis perusahaan yang semakin pesat dan tahun 2002 DKPP fokus terhadap penjualan produk Indosat dan bergabung menjadi dealer Indosat seiring kebijakan Indosat melakukan pola pendistribusian melalui distributor (*Indirect distribusi*). Kini DKPP telah beroperasi di wilayah Jawa Barat dan Jabodetabek dengan menghadirkan kantor penjualan dan pelayanan di Jakarta, Depok, Tangerang, Bandung, Lembang, Sukabumi dan Cianjur.

2.1.2 Visi dan Misi PT. DKPP

Visi

Menjadi perusahaan distribusi terbesar di bidang telekomunikasi dan produk pendukungnya yang berorientasi kepada kepuasan pelanggan dan mitra.

Misi

1. Melakukan pemasaran seluruh produk Indosat kepada seluruh lapisan masyarakat serta memberikan kemudahan dalam pelayanannya.
2. Menjadi distributor dengan jaringan terbanyak yang dapat diandalkan dalam pemasaran produk melalui infrastruktur lengkap.
3. Memberikan pelayanan terbaik atas kebutuhan komunikasi, baik kepada konsumen langsung maupun kepada mitra outlet atau reseller.
4. Mengembangkan jaringan usaha penjualan melalui kerjasama saling menguntungkan dengan pihak lain.
5. Meningkatkan keunggulan kompetitif, melalui inovasi teknologi yang berkelanjutan sesuai dengan kebutuhan konsumen dan perkembangan teknologi yang cepat.

2.1.3 Logo DKPP

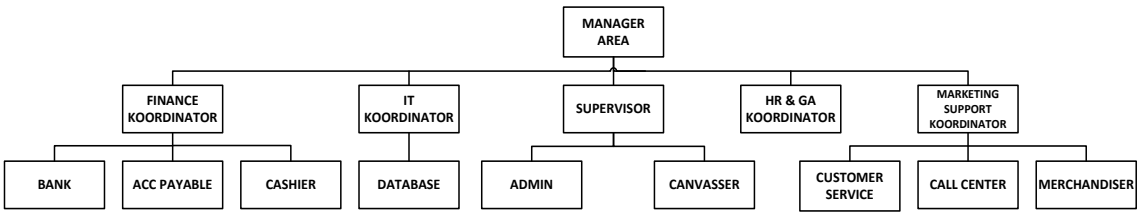
Tidak hanya memiliki desain yang bagus tetapi logo dibuat biasanya memiliki arti yang menggambarkan siapa pemilik logo tersebut seperti logo pada organisasi atau perusahaan baik milik pemerintah maupun swasta. Berikut logo PT.DKPP pada Gambar 2.1.

Gambar 2.1 Logo DKPP

PT. DKPP memiliki logo bertuliskan singkatan nama perusahaan yaitu DKPP (Dian Kencana Puri Prima), kemudian terdapat bidang berbentuk seperti ouval yang tidak penuh mengelilingi nama perusahaanyang memiliki makna tim DKPP yang solid dalam satu visi dan misi yang sama.Sedangkan warna kuning gelap pada bidang tersebut merupakan warna dominan atau ciri khas yang digunakan Indosat yaitu kuning.Dan dibawah bertuliskan *Authorized Indosat Distributor* yang berarti DKPP merupakan Distributor Eksklusif atau sah yang dimiliki Indosat.

2.1.4 Struktur Organisasi PT. DKPP

Struktur baru Organisasi yang terbentuk di PT. DKPP, terdiri dari *Head of Region, Head ofArea, Finance, IT, Sales, HR & GA,* dan *Marketing Support.* Dapat dilihat pada gambar 2.2 Struktur Organisasi PT.DKPP Bandung.



Gambar 2.2 Struktur Organisasi PT.DKPP Bandung

2.2 Landasan Teori

Menjelaskan beberapa teori yang berkaitan dengan permasalahan yang dibahas sebagai dasar pemahaman dalam sebuah sistem serta metode yang dipakai dalam sistem yang akan dibangun.

2.2.1 Metode Penilaian *Performance Canvasser* dan *Cluster*

Performance canvasser dinilai berdasarkan target *visit*, *reload,* dan target SP, sedangkan *Performance cluster* dinilai berdasarkan *Sell In* dan *Sell Out* yang terjadi di seluruh outlet yang berada di *subcluster-subcluster* dalam *cluster* BDG2. Metode penilaian yang digunakan PT.DKPP untuk menilai *Performance canvasser* dan *cluster* yaitu:

1. Metode Penilaian Target Canvasser
2. Metode Penilaian Performance Produk Cluster

2.2.1.1 Metode Penilaian Target Canvasser

Dalam menilai pencapaian target *visit* dan penjualan *canvasser* digunakan beberapa metode penilaian sebagai berikut:

1. Metode untuk menghitung pencapaian target *visit canvasser*

$$Visit = \frac{Achievement\ Visit}{Target\ Visit} x100\% \quad (2.1)$$

2. Metode untuk menghitung pencapaian target penjualan *reload canvasser*

$$Reload = \frac{Achievement\ Sell\ In\ Reload}{Target\ Sell\ In\ Reload} x100\% \quad (2.2)$$

3. Metode menghitung pencapaian target penjualan SP

$$SP = \frac{Achievement\ Sell\ In\ SP}{Target\ Sell\ In\ SP} x100\% \quad (2.3)$$

4. Metode menghitung Total Pencapaian Target *Canvasser*

$$Total\ Achievement = (Visit\ x\ 40\%) + (Reload\ x\ 30\%) + (SP\ x\ 30\%) \quad (2.4)$$

2.2.1.2 Metode Penilaian *Performance* Produk *Cluster*

Performance sub cluster merupakan hasil jumlah penjualan produk dari outlet ke *end user* (*Sell Out)* dibagi hasil jumlah penjualan produk dari *canvasser* ke outlet (*Sell In*).

1. Metode menghitung daya beli dan jual *Reload* di suatu *subCluster*

$$Reload\ On\ Sub\ Cluster = \frac{Total\ Sell\ Out\ Reload\ on\ Sub\ Cluster}{Total\ Sell\ In\ Reload\ on\ Sub\ Cluster}\ x\ 100\% \quad (2.5)$$

2. Metode Menghitung Daya Beli dan Jual Produk SP di suatu *subCluster*

$$SP\ On\ Sub\ Cluster = \frac{Total\ Sell\ Out\ SP\ on\ Sub\ Cluster}{Total\ Sell\ In\ SP\ on\ Sub\ Cluster}\ x\ 100\% \quad (2.6)$$

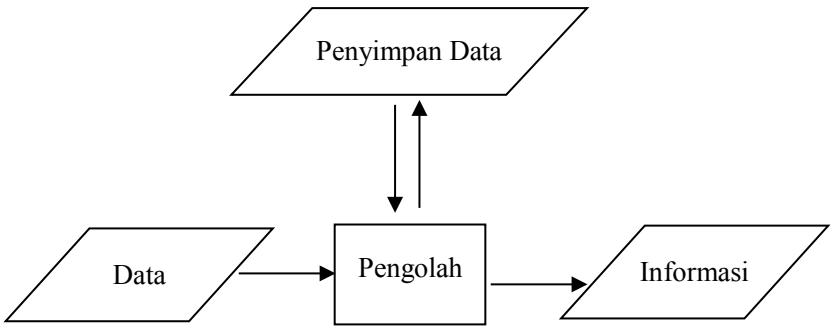
3. Metode Menghitung Total Daya Beli dan Jual Produk di suatu *subCluster*

$$Sub\ Cluster\ Performance = Mean(Reload\ On\ Sub\ Cluster + SP\ On\ Sub\ Cluster) \quad (2.7)$$

2.2.2 Sistem Informasi

Sistem adalah kumpulan elemen-elemen atau komponen-komponen atau subsistem-subsistem yang saling berintegrasi dan berinteraksi untuk mencapai tujuan tertentu.

Sumber informasi adalah data, data adalah bentuk jamak dari bentuk tunggal datum atau data item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.Kejadian-kejadian (*event)* adalah sesuatu yang terjadi pada saat tertentu.Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang. Hubungan antara data dengan informasi dapat dilihat pada gambar 2.3.



Gambar 2.3 Transformasi data menjadi informasi

Sistem Informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi. Bentuk-bentuk sistem informasi :

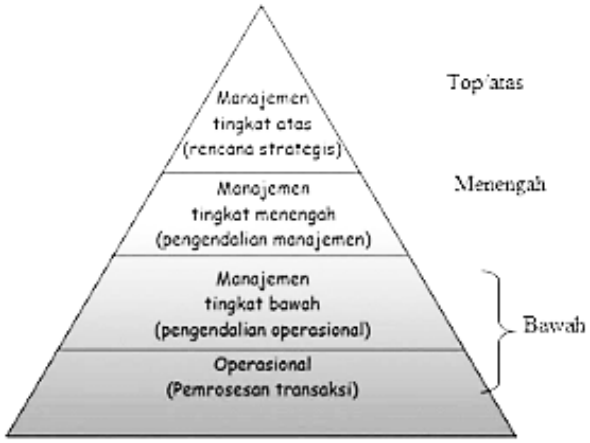
a. Sistem Informasi Manajemen (Management Information System)
b. Sistem Pendukung Keputusan (Decision Support System)
c. Sistem Informasi Eksekutif (Executive Information System)
d. Sistem Otomasi Perkantoran (Office Automation)
e. Sistem Pendukung Pakar (Expert Support System)
f. Sistem Proses Transaksi (Transaction Processing System)

2.2.2.1 Sistem Informasi Manajemen

Sistem Informasi Manajemen adalah suatu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan yang serupa. Sistem informasi manjemen merupakan suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolah transaksi harian, mendukung operasi yang bersifat manajerial dan kegiatan strategi dari suatu organisasi dan pihak luar tertentu dengan laporan yang diperlukan. Para pemakai sistem infomasi ini biasanya membentuk suatu entitas organisasi formal dalam suatu perusahaan, sedangkan informasi yang digunakan menjelaskan mengenai apa yang terjadi di masa lalu, masa sekarang dan masa yang akan datang. Sebagai contoh dalam sistem informasi manajemen pada umumnya memberikan informasi dalam bentuk laporan periodik, laporan khusus, dan simulasi bila diperlukan.Output dari sistem informasi manajemen berupa informasi dalam bentuk laporan-laporan yang disajikan untuk level manajemen tertentu. Karena sistem informasi ini merupakan salah satu sistem informasi yang mendukung kerja manajerial, maka perlu diketahui hal-hal yang berkaitan dengan manajemen dimulai dari pembagian level manajemen sampai dengan tipe informasi yang disajikan bagi level manajemen tertentu. Ini penting terutama jika dikaitkan dengan bagaimana merancang suatu sistem informasi manajemen bagi organisasi/instansi/institusi/perusahaan.

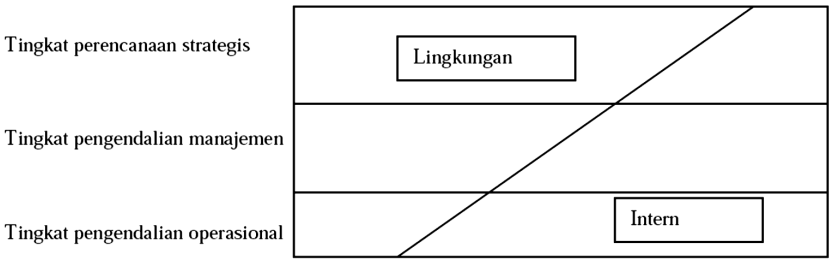
Fungsi manajemen meliputi: perencanaan, pembuatan keputusan, pengarahan, pengkoordinasian, pengontrolan, dan penyempurnaan. Dari sekian banyak fungsi manajemen, yang erat kaitannya dengan sistem informasi

manajemen adalah fungsi pembuatan keputusan. Pembuatan keputusan adalah aktivitas manajemen berupa pemilihan dari berbagai alternatif tindakan yang mungkin dipilih yang prosesnya melalui mekanisme tertentu dengan harapan akan menghasilkan keputusan yang terbaik. Level manajemen dikelompokkan menjadi 3 level : *top*/atas, menengah, dan bawah dapat dilihat pada gambar 2.4.



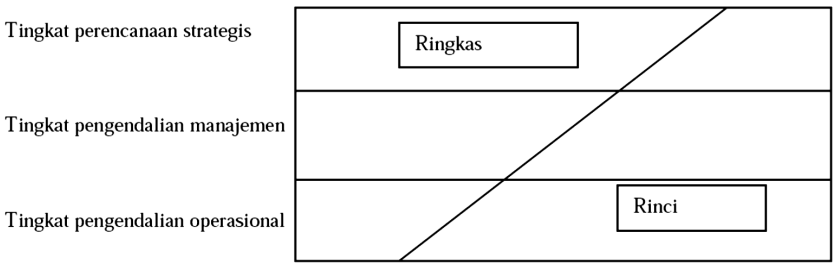
Gambar 2.4 Level manajemen

Gambar 2.5 menggambarkan dimana level manajemen dapat mempengaruhi sumber informasi dan bentuk penyajian informasi tersebut.



Gambar 2.5 Hubungan level manajemen dan sumber informasi

Gambar 2.6 merupakan hubungan level manajemen dengan tipe atau bentuk dari sistem informasi.



Gambar 2.6 Hubungan level manajemen dengan tipe / bentuk sistem informasi.

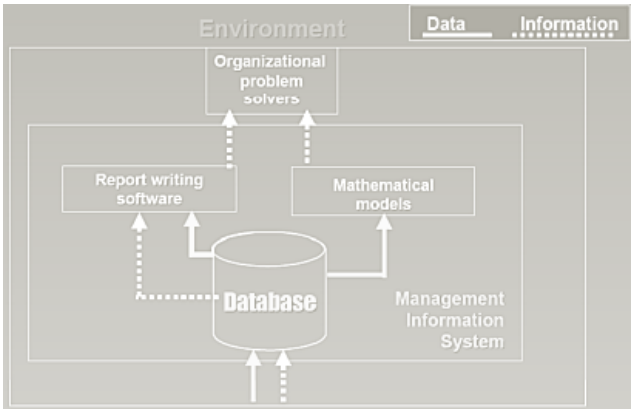
Dalam sistem informasi manajemen laporan periodik disiapkan sesuai dengan jadwal yang ditetapkan oleh suatu perusahaan, sedangkan laporan khusus disiapkan apabila sesuatu yang khusus/luar biasa terjadi dalam suatu pencatatan informasi, seperti kejanggalan informasi.Alasan perlu diberikan perhatian pada sistem manajemen informasi karena kompleksitas kegiatan bisnis meningkat (pengaruh ekonomi internasional, persaingan dunia, kompleksitas teknologi yang semakin meningkat, batas waktu yang singkat, dan kendala sosial), kemampuan komputer yang samakin baik (pemakai semakin tahu bagaimana mendayagunakan komputer untuk membantu pekerjaanya).

Membangun sebuah sistem informasi manajemen secara prinsip sama dengan membangun sistem informasi, yang perlu ditekankan adalah untuk siapa output sistem informasi ini akan digunakan, hal ini erat kaitannya dengan bentuk informasi yang akan dihasilkan oleh sistem informasi manajemen.

Model digunakan dalam sistem informasi manajemen untuk mempermudah pengertian, mempermudah komunikasi, dan memperkirakan masa depan. Para manajemen menggunakan model untuk permasalahan yang akan dipecahkan. Objek atau aktivitas yang menyebabkan permasalahan adalah entitas.Jenis dasar model ada 4 macam yaitu model fisik, model naratif, model grafik, dan model matematika. Pemodelan matematika terpenting dalam sistem informasi manajemen. Pemodelan matematika dipengaruhi oleh tiga dimensi yaitu pengaruh waktu, tingkat ketidakpastian, dan kemampuan mengoptimalkan suatu informasi. Tipe pemodelan matematika dibagi menjadi enam bagian yaitu bersifat statis (tidak meliputi waktu, seperti *snapshot*), dinamis (meliputi waktu, seperti film), probabilistik, deterministik, *optimizing* (pemilihan keputusan yang terbaik, secara terstruktur), *suboptimizing* (tingkat kepuasan). *Output* dari pemodelan matematika dapat berupa laporan, bentuk grafik (*line chart, bar chart*).

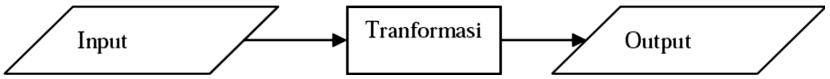
Pemodelan matematika bila perlu dapat disimulasikan untuk mempertimbangkan beberapa banyak pilihan, menyediakan ramalan/prediksi, dan lebih baik dibandingkan dengan metode *trial/error*. Kerugian yang didapatkan dalam permodelan matematika adalah kesulitan untuk memodelkan dalam suatu

bisnis dan perlu ketrampilan matematis. Pada gambar 2.7 dapat dilihat model dari manajemen sistem informasi yang seharusnya.



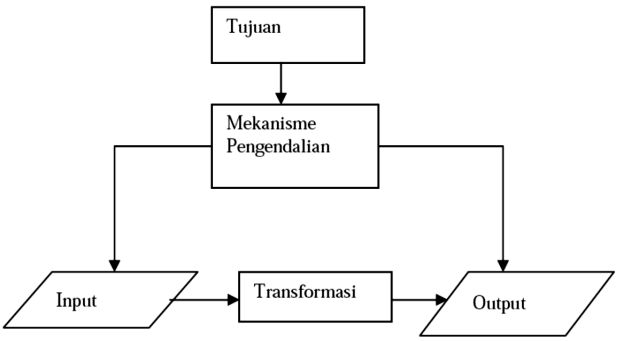
Gambar 2.7 Model manajemen sistem informasi

Ada 2 model sistem yang umumnya digunakan dalam sebuah perusahaan yaitu sistem lingkaran terbuka dan sistem lingkaran tertutup. Sistem lingkaran terbuka berhubungan langsung dengan lingkungannya, arus sumber dayanya tanpa ada umpan balik, dengan tidak adanya umpan balik maka sulit untuk dilakukan kontrol produksi. Sistem lingkaran terbuka dapat dilihat pada gambar 2.8.



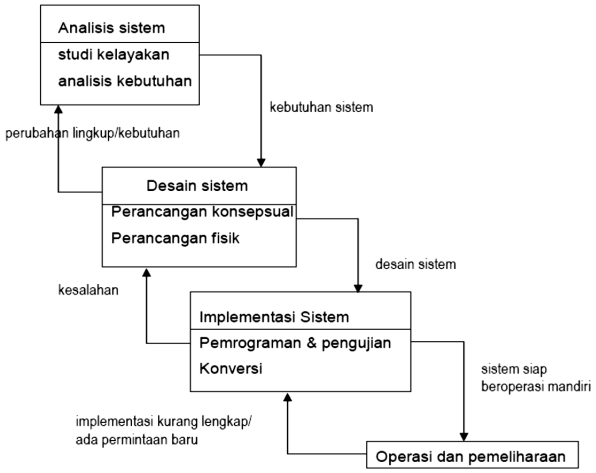
Gambar 2.8 Sistem lingkaran terbuka

Sistem lingkaran tertutup memiliki tiga elemen kontrol berupa tujuan, mekanisme kontrol, dan lingkaran umpan balik. Umpan balik berguna untuk proses pengendalian informasi sehingga informasi yang diterima relevan, akurat, tepat waktu dan lengkap. Berikut bentuk dari sistem lingkaran tertutup pada gambar 2.9.



Gambar 2.9 Sistem lingkaran tertutup

Sistem informasi manajemen menyediakan informasi pemecahan masalah untuk menetapkan tahap pemenuhan area lain seperti *Decising Support System* (Sistem Pengambilan Keputusan), *Office Automation* (Otomatisasi Kantor), dan *Knowlegde-based System* (Sistem Berbasis Pengetahuan). Dalam membangun sistem informasi manajemen digunakan metodologi untuk melaksanakan proses standar yang diperlukan seperti menganalisa, merancang, mengimplementasikan, dan memelihara sistem informasi. Daur hidup dari pengembangan sistem ini disebut dengan daur hidup pengembangan sistem (SDLC = *System Development Life Cycle*) dapat dilihat pada gambar 2.10.



Gambar 2.10 SDLC

Analisis sistem dilakukan karena adanya permintaan sistem yang baru. Analisis sistem menentukan hal-hal terkait dengan apa yang akan dikerjakan oleh sistem. Tahap analisis sistem ini meliputi studi kelayakan dan analisis kebutuhan. Desain sistem dilakukan setelah proses analisis sistem dikerjakan. Perancangan sistem ini dibagi menjadi 2 kelompok: perancangan konsepsual (menentukan alternatif rancangan yang bisa dipakai meliputi masukan, penyimpanan data, keluaran berupa penyiapan laporan), perancangan fisik (rancangan keluaran, rancangan masukan, rancangan layar, rancangan antar muka, rancangan *platform*, rancangan basis data, rancangan modul, rancangan kontrol, dokumentasi, rencana pengujian, rencana konversi).

Aktivitas-aktivitas yang ada dalam fase implementasi sistem meliputi: pemrograman dan pengujian, instalasi perangkat keras dan perangkat lunak,

pelatihan/training, pembuatan dokumentasi, pengembangan, dan konversi. Sistem informasi manajemen menyediakan informasi pemecahan masalah untuk menetapkan tahap pemenuhan area lain seperti *Decising Support System* (Sistem Pengambilan Keputusan), *Office Automation* (Otomatisasi Kantor), dan *Knowlegde-based System* (Sistem Berbasis Pengetahuan).

2.2.2.2 Sistem Informasi Berbasis Komputer

Sistem Informasi Manajemen adalah suatu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan yang serupa.Penggunaan sistem informasi berbasis komputer dalam sebuah organisasi atau perusahaan memiliki manfaat dapat meningkatkan keuntungan perusahaan, mengurangi biaya bisnis, meningkatkan pangsa pasar (ekspansi berupa jangkauan dan variasi produk), perbaikan relasi pelanggan atau pelayanan, keamanan data, meminimalkan kesalahan, dan dapat memperbaiki dalam pengambilan keputusan.Fokus utama sistem informasi berbasis komputer salah satunya untuk aplikasi otomatisasi perkantoran (*Office Automation*/OA).Otomatisasi sangat berkaitan erat dengan mekanisasi dan komputerisasi. Hal ini mengisyaratkan bahwa otomatisasi berarti penggunaan alat-alat mekanis dan lebih khususnya komputer. Membahas otomatisasi berarti mengupas berbagai peralatan mekanis dan komputer dengan tetap memperhatikan relevansinya dengan objek yang diotomatisasi, dalam hal ini perkantoran. Otomatisasi perkantoran berarti pengalihan fungsi manual peralatan kantor yang banyak menggunakan tenaga manusia kepada fungsi-fungsi otomatis dengan menggunakan peralatan mekanis khususnya komputer. Era otomatisasi perkantoran dimulai bersamaan dengan berkembangnya teknologi informasi, penggunaan perangkat komputer untuk keperluan perkantoran. Otomatisasi perkantoran sering juga diistilahkan dengan kegiatan perkantoran elektronis (*electronic office/e-office)*. Perkantoran elektronis adalah aplikasi perkantoran yang mengganti proses administrasi berbasis manual ke proses berbasis elektronis dengan memnfaatkan fasilitas jaringan lokal (*Local Area Network*). Otomatisasi perkantoran tidak bisa dipisahkan dari teknologi perkantoran sebab otomatisasi merupakan bentuk pengembangan teknologi, dalam hal ini pergeseran dari teknologi manual kepada teknologi otomatis. Teknologi

perkantoran adalah bagaimana proses mencatat, menghimpun, mengolah, memperbanyak, mengirim dan menyimpan bahan-bahan keterangan secara efisien dengan menggunakan mesin-mesin. Oleh karena itu perlu sekali mengenal tentang jenis mesin untuk keenam macam kegiatan itu dan bagaimana cara menggunakannya.

Konsep-konsep otomatisasi:

1. Proses yang terjadi diperkantoran seperti halnya proses manufaktur selalu mengarah ke otomatisasi.
2. Otomatisasi kantor berevolusi dari aplikasi-aplikasi yang terpisah dan tanpa rencana menuju aplikasi yang terencana dan terpadu.
3. Otomatisasi kantor memudahkan penerimaan dan pengiriman informasi.
4. Otomatisasi kantor memberikan keuntungan lebih besar melalui pengambilan keputusan yang lebih baik.
5. Otomatisasi kantor sebagai pelengkap bagi metode komunikasi tradisional bukan sebagai pengganti.

Otomatisasi perkantoran merupakan kaitan berbagai komponen dalam menangani informasi mulai dari input hingga distribusi dengan memanfaatkan bantuan teknologi secara optimal dan campur tangan manusia secara minimal. Dengan demikian akan membuat informasi menjadi lebih mudah dan murah digunakan, dipindahkan, dan dirawat. Pada akhirnya dapat meletakkan landasan yang kuat untuk integrasi informasi sehinggga perusahaan mampu berkompetisi lebih baik. Otomatisasi penting dilakukan dalam upaya meraih efektivitas dan efisiensi proses/kegiatan perkantoran. Seiring dengan desakan global dan perkembangan teknologi informasi yang menuntut terselesaikan proses pengolahan informasi secara cepat dan akurat, kebutuhan peralihan metode dari manual ke otomatis sudah menjadi keniscayaan untuk segera dipenuhi. Namun, bukan berarti dengan serta merta meninggalkan seluruh proses manual dan memangkas tenaga kerja, sebab banyak aspek-aspek lain yang harus menjadi pertimbangan dalam melakukan otomatisasi[6].

2.2.3 *Customer Relationship Management* (CRM)

CRM merupakan salah satu sarana untuk menjalin hubungan yang berkelanjutan antara perusahaan dengan para *stakeholder* dan *shareholdernya*. Saat ini banyak perusahaan yang memanfaatkan CRM untuk menjalin hubungan dengan pelanggan. Dengan memanfaatkan CRM, perusahaan akan mengetahui apa yang diharapkan dan diperlukan pelanggannya sehingga akan tercipta ikatan emosional yang mampu menciptakan hubungan bisnis yang erat dan terbuka serta komunikasi dua arah.

2.2.3.1 CRM dan Tujuannya

Adalah strategi bisnis yang terdiri dari *software* dan layanan yang didesain untuk meningkatkan keuntungan (*profit*), pendapatan (*revenue*) dan kepuasan pelanggan pelanggan (*customer satisfaction*). Caranya adalah dengan membantu berbagai bentuk perusahaan untuk mengidentifikasi pelanggannya dengan tepat, memperoleh lebih banyak pelanggan dengan lebih cepat, dan mempertahankan kesetiaan pelanggannya.

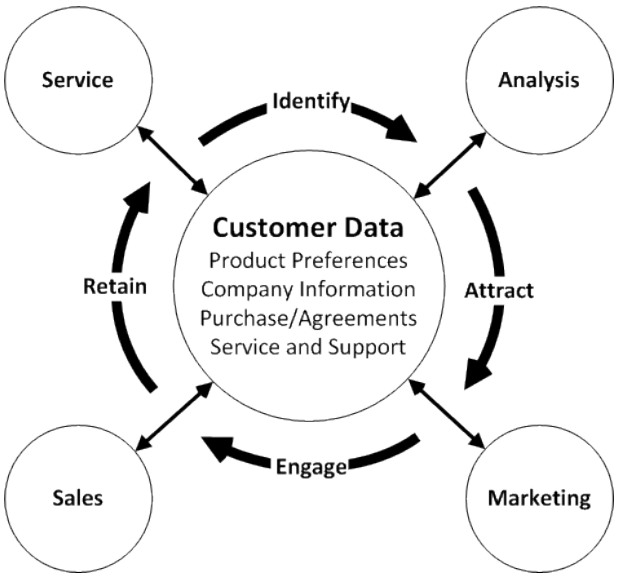
Sistem CRM berusaha menyediakan sebuah pendekatan terintegrasi terhadap semua aspek dalam perusahaan dalam kaitannya pelanggannya, yang meliputi *marketingsalesand support*. Tujuannya adalah dengan penggunaan teknologi diharapkan terjadi jalinan hubungan yang kuat antara perusahaan dengan pelanggannya. Dengan kata lain, perusahaan berusaha mengelola kinerja perusahannya dengan lebih baik.

2.2.3.2 Kebutuhan CRM

CRM meliputi sentralisasi semua data pelanggan perusahaan dan otomatisasi kegiatan mengelola *sales*, *marketing,* dan *customer service* yang membosankan sehingga para professional dapat menggunakan waktunya untuk melayani pelangganyya secara lebih efektif dan efisien terutama dalam melakukan kegiatan administrasinya.

Keuntungan CRM sudah jelas yaitu dapat mempercepat proses dan memberikan kemudahan bagian *sales, marketing,* dan *service personnel* dengan informasi pelanggan yang lebih baik dan lengkap. CRM memberi kemudahan

pada perusahaan untuk membangun *customer relationship* dan mengurangi biaya operasional. Alur aktivitas CRM dapat dijelaskan melalui gambar 2.11 berikut [2].



Gambar 2.11 Alur Aktivitas CRM

2.2.4 *Database* (Basis Data)

Data merupakan data mentah. Manajemen data adalah disiplin ilmu yang berkonsentrasi dalam penyesuaian generasi, penyimpanan, dan data retrival. Manajemen data secara tipikal membutuhkan komputer. adalah *sharing* data, dalam struktur komputer yang terintegrasi dengan koleksi penyimpanan dari :

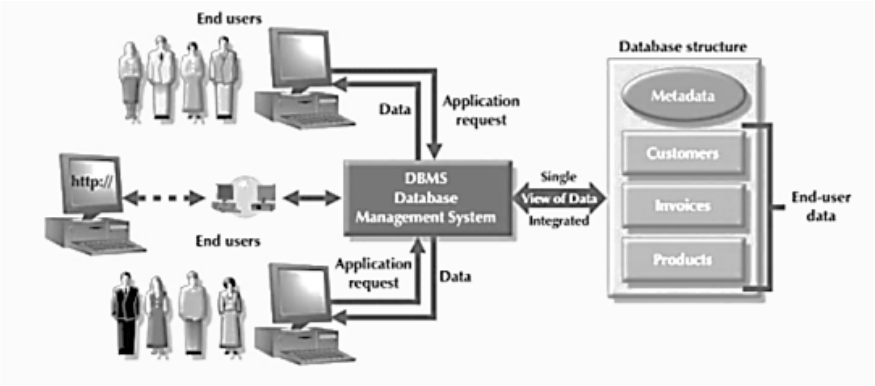
a. Data *end user* yang merupakan data mentah yang bermanfaat bagi *end user*

b. Meta data atau data apapun yang nantinya data terintegrasi dan teratur oleh *end user*

DBMS atau *management system* adalah koleksi dari program yang mengatur secara terstruktur dan mengkontrol akses data penyimpanan dalam. Dalam perumpamaan, menyerupai rak kabinet yang sangat terorganisasi dengan baik dimana keunggulan *software* ini dikenal dengan istilah DBMS, untuk membantu mengatur isi kabinet.

2.2.4.1 Keuntungan dalam DBMS

Server DBMS merupakan media antara pengguna dengan. Strukturnya menyimpan kumpulan dari file. Cara mengakses data melalui file yaitu menggunakan DBMS. Pada Gambar 2.12 menekankan bahwa DBMS menyajikan kepada *end user* (program aplikasi), mengintegrasikan data kedalam. DBMS menerima semua permintaan aplikasi dan menterjemahkannya kedalam operasi yang kompleks yang dibutuhkan untuk memenuhi request yang diminta. DBMS menyembunyikan kompleksitas internal dari program aplikasi dan *user*. DBMS mungkin ditulis oleh programmer menggunakan bahasa pemrograman seperti VB.Net, Java atau C++ atau mungkin bisa dibangun melalui sistem DBMS.



Gambar 2.12 DBMS mengatur interaksi antara *end user* dan

DBMS antara *end user* dan aplikasi memiliki beberapa keuntungan. Pertama, DBMS memungkinkan mengambil data dalam untuk di-*sharing* dibeberapa aplikasi atau *user*. Kedua, DBMS terintegrasi dengan beberapa *level user* yang mencakup keseluruhan data penyimpanan.

DBMS membantu manajemen data lebih efisien dan efektif. Keuntungan DBMS sebagai berikut :

1. Meningkatkan Sharing Data
2. Meningkatkan keamanan Data
3. Integrasi Data yang lebih baik
4. Minimalisasi ketidakonsistensian data
5. Meningkatkan akses data
6. Meningkatkan pembuat keputusan
7. Menaikan produktifitas end-*user*

Keuntungan menggunakan DBMS tidak hanya terbatas pada yang tertulis diatas. Dalam kenyataannya, dapat dicari beberapa keuntungannya yang berhubungan mengenai detail teknikal dan desain yang sesuai.

2.2.4.2 Tipe *Database*

DBMS banyak mendukung semua tipe. dapat diklasifikasikan menurut banyaknya *user*, lokasi, tipe data yang digunakan, dan kegunaan secara luas.

1. Banyaknya *user* diklasifikasikan sebagai,
   a. *single user*

      Saat *user* A menggunakan maka *user* B atau C harus menunggu hingga *user* A selesai mengakses. *Single user* menjalankan *personal computer* yang disebut *desktop*.

   b. *Multiuser*

      Multiuser mendukung *user* menggunakan di waktu yang bersamaan. *Multiuser* yang -nya mendukung *user* dalam jumlah kecil (kurang dari 50 *user*) atau berdasarkan departemen tertentu dalam suatu organisasi disebut *workgroup.* Sedangkan ketika mendukung pengaksesan yang *user*-nya lebih dari 50 (biasanya dalam jumlah ribuan) dan berhubungan dengan banyak departemen maka ini dikenal *enterprise.*

2. Pengaruh lokasi dapat digunakan sebagai pengklasifikasian :
   a. yang ditempatkan ke dalam site tunggal disebut tersentral
   b. yang mendukung distribusi data di semua site yang berbeda disebut terdistribusi.

Yang paling banyak dipakai saat ini dalam pengklasifikasian mungkin berdasarkan bagaimana digunakan dalam waktu yang sempit untuk mengumpulkan suatu data. Contohnya yakni transaksi produk atau layanan penjualan/sales, pembayaran, dan stok pembelian menggambarkan operasi dari hari ke hari. Seperti transaksi harus direkam secara akurat dan segera. Sebuah di desain untuk mendukung operasi perusahaan dari hari ke hari dan diklasifikasikan sebagai *operasional* (sering disebut sebagai transaksional atau produksi). Sedangkan data *warehouse* fokus kepada penyimpanan data yang digunakan untuk menghasilkan informasi yang dibutuhkan untuk membuat keputusan

strategi, contohnya seperti keputusan pemberian harga, *sales forecast*, posisi *market*.

Database juga dapat diklasifikasikan darimana data disusun yakni data terstruktur (data yang diambil dari data mentah yang kemudian dikumpulkan untuk mendapatkan sebuah informasi), data tidak terstruktur (data mentah yang masih disimpan ditempat data itu dikumpulkan) dan data semi terstruktur (data yang telah diproses yang kemudian dapat diperluas fungsinya, contohnya XML). Menyimpan dan mengatur data tidak terstruktur dan semi terstruktur membutuhkan pengalamatan melalui generasi baru dari yang dikenal sebagai XML (*Extensible Markup Language*). XML adalah bahasa khusus yang digunakan untuk menyajikan dan memanipulasi elemen data ke dalam format *text*. Berikut pada gambar 2.13 dapat dilihat dasar dari terminology *database*.

TERM	DEFINITION
Data	"Raw" facts, such as a telephone number, a birth date, a customer name, and a year-to-date (YTD) sales value. Data have little meaning unless they have been organized in some logical manner.
Field	A character or group of characters (alphabetic or numeric) that has a specific meaning. A field is used to define and store data.
Record	A logically connected set of one or more fields that describes a person, place, or thing. For example, the fields that constitute a record for a customer might consist of the customer's name, address, phone number, date of birth, credit limit, and unpaid balance.
File	A collection of related records. For example, a file might contain data about the students currently enrolled at Gigantic University.

Gambar 2.13 Terminology Dasar *Database*

a. *Data* : adalah fakta mentah seperti nomor telepon, angka kelahiran, nama pelanggan, tahun dan tanggal, nilai penjualan. Data memiliki manfaat yang sedikit yang harus diatur kedalam cara yang logis.
b. *Field* : merupakan karakter atau grup dari karakter (alphabet atau numeric) yang memiliki arti tertentu *field* digunakan untuk menggambarkan penyimpanan data.
c. *Record* : secara logika terhubung oleh satu atau lebih *field* yang mendeskripsikan orang, tempat, benda contohnya *field* mendasari rekor untuk pelanggan yang terdiri dari nama pelanggan, alamat, nomer telepon, tanggal lahir, batas kredit, blanace yang belum dibayar.
d. *File* : kumpulan dari *record* yang saling berhubungan. Contohnya sebuah file berisi data mengenai pelajar yang saat ini terdaftar di Universitas Gigantic[7].

2.2.5 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan aturan-aturan pemodelan yang digunakan untuk mendeskripsikan sistem perangkat lunak dalam bentuk kumpulan objek. UML bukanlah suatu metode untuk mengembangkan sistem tetapi notasi-notasi yang digunakan secara umum sebagai standar untuk pemodelan objek.

Objek merupakan sesuatu yang dapat dilihat, disentuh atau dirasakan dan digunakan pengguna serta akan disimpan data dan perilakunya. Objek bisa berupa: orang, tempat, benda atau kejadian, pegawai, pelanggan, guru, dosen, mahasiswa, murid, gudang, kantor, bangunan, ruangan, kendaraan, produk, komputer, video.

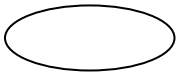
a. *Attribut :* merupakan data yang menyatakan sifat dari objek.
b. *Object instance :* merupakan orang, tempat, benda atau kejadian khusus beserta nilai untuk setiap atribut dari objek.
c. *Behavior*/perilaku : merupakan hal-hal yang dapat dikerjakan objek dan merupakan fungsi yang dikenakan pada data/atribut dari objek tersebut. Dalam paradigma orientasi berbasis objek, perilaku objek disebut metode, operasi, atau layanan.

UML sendiri terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 9 diagram, yaitu *use-case, class, object, state, sequence, collaboration, activity, component,* dan *deployment diagram.*

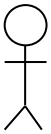
2.2.5.1 *Use Case* Diagram

Menggambarkan fungsionalitas yang diharapkan dalam sebuah sistem. Dalam *Use Case* Diagram yang ditekankan adalah ‘apa’ yang diperbuat sistem dan bukan ‘bagaimana’. Menggambarkan kebutuhan sistem dari sudut pandang, mengfokuskan pada proses komputerisasi, menggambarkan hubungan antara *use case*, aktor, *association, generalization, boundary boxes.*

Use case dibuat berdasarkan keperluan aktor, *use case* biasanya menggunakan kata kerja, tidak boleh ada 2 *use case* yang memiliki nama yang sama. Berikut simbol *user case* pada gambar 2.14 dan aktor pada gambar 2.15



Gambar 2.14 Simbol *Use Case*



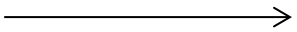
Gambar 2.15 Simbol Aktor

Association digunakan untuk menggambarkan bagaimana aktor terlibat dalam *use case.* Ada 4 jenis relasi yang timbul pada *use case* sebagai berikut.

1. *Association* antara aktor dan *use case*

   Digambarkan pada gambat 2.16 dan gambar 2.17

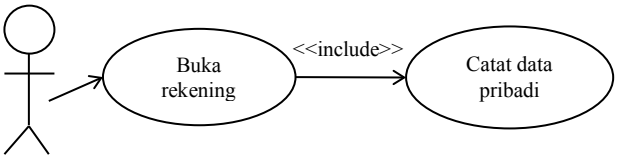
Gambar 2.16 Simbol Garis Tanpa Panah



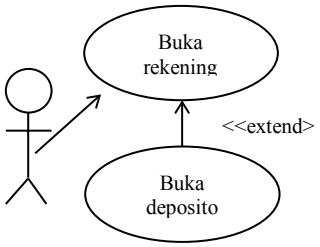
Gambar 2.17 Simbol Panah Terbuka

2. *Association* antara *use case*

   Gambar. 2.18 merupakan contoh dari *Association* <<*include*>>, sedangkan pada gambar 2.19 merupakan contoh dari *Association* <<*extend*>>.



Gambar 2.18 Contoh *Association* <<*include*>>



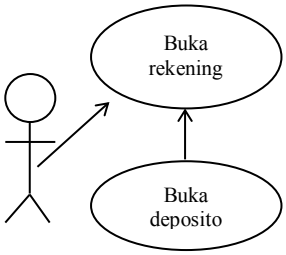
Gambar 2.19 Contoh *Association* <<*extend*>>

3. *Generalization* antara *use case*

   Gambar 2.20 merupakan garis berpanah tertutup yang digunakan dalam *generalization* antar *user case*. Contoh dari *generalization* antar *use case* dapat dilihat pada gambat 2.21.



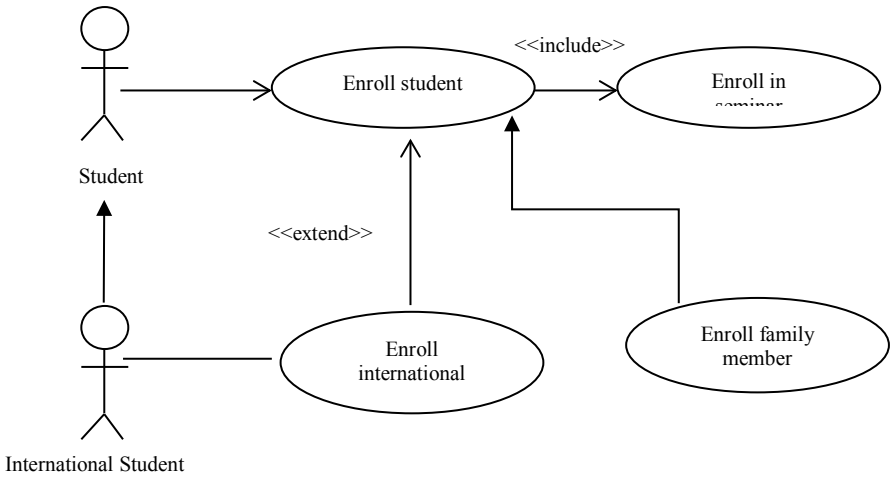
Gambar 2.20 Simbol Garis Berpanah Tertutup



Gambar 2.21 Contoh *Generalization* Antar *User Case*

4. *Generalization* antara aktor

   Contoh dari *generalization* antar aktor pada gambar 2.22.



Gambar 2.22 Contoh *Generalization* Antar Aktor

Boundary boxes digambarkan dengan kotak sekitar *use case* untuk menggambarkan jangkauan sistem yang dibuat. Digunakan untuk memberikan beberapa alternatif sistem sebagai pilihan dan bersifat optional.

2.2.5.2 Activity, Sequence, Collaboration, Class, Component dan Deployment Diagram

1. *Activity diagram* menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. *Activity diagram* digunakan pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. Struktur ini mirip *flow chart* pada perancangan terstruktur.
2. *Sequence Diagram m*erupakan diagram interaksi yang menekankan urutan waktu pertukaran pesan.
3. *Collaboration Diagram* hampir sama dengan *sequence diagram,* hanya saja interaksi antara objek dan aktor tidak ditampilkan berdasarkan urutan waktu.
4. *Class Diagram* menggambarkan struktur dan deskripsi *class*, *package,* dan objek beserta hubungan satu sama lain seperti *containment,* pewarisan, asosiasi, dan lain-lain. Tiga area pokok dalam *class diagram* yaitu nama (dan *stereotype),* atribut, metoda. Atribut dan metoda memiliki sifat *private, protected, dan public.*
5. *Component Diagram* memperlihatkan organisasi dan ketergantungan anyara sekumpulan komponen(*implementation view*). Dihubungkan dengan *class diagram* dimana komponen tersebut memetakan satu atau lebih *class, interface/collaboration.*
6. *Deployment Diagram* memperlihatkan konfigurasi *run-time* pada node pemrosesan dan komponen yang berjalan padanya. Dihubungkan dengan *component diagram* dimana sebuah node menyertakan satu atau lebih komponen[3].

2.2.6 Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk 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.

2.2.6.1 *The Dalvik Virtual Machine* (DVM)

DVM merupakan salah satu elemen kunci dari Android.Android berjalan di dalam DVM bukan di JVM (*Java Virtual Machine*).Sebenarnya banyak persamaanya dengan JVM seperti Java ME.Tetapi android menggunakan mesin sendiri yang dikustomisasi dan dirancang untuk memastikan bahwa beberapa fitur berjalan lebih efisien pada perangkat *mobile*.

Dalvik Virtual Machine (DVM) adalah *register bases* sementara *Java Virtual Machine* (JVM) atau *stack based*, DVM disesain dan ditulis oleh Dan Bornsten dan beberapa *engineers* Google lainya. Jadi bias kita katakana *Dalvik equals(Java) == False*. *Dalvik Virtual Machine* menggunakan *kernel* linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis Aplikasi C/C+ sama halnya seperti pada OS Linux kebanyakan. Meskipun dalam kenyataan kita harus memahami arsitektur dan proses sistem dari *kernel* linux yang digunakan dalam android tersebut.

Dengan demikian semua *hardware* yang berbasis Android dijalankan dengan menggunakan *Virtual Machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu.*Dalvik Virtual Machine* mengeksekusi *executable file*, sebuah format yang di optimalkan untuk memastikan memori yang digunakan sangat kecil.*Executable file* diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan *tools* yang disediakan dalam SDK Android.

2.2.6.2 Arsitektur Android

Secara garis besar arsitektur Android dapat dijelaskan sebagai berikut:

1. *Application* dan *Widgets*

   Application dan *Widgets* ini adalah *layer* di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita unduh aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti

termasuk klien email, program SMS, Kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemgrograman Java.

2. *Applications Frameworks*

   Android adalah *Open Development platform* yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas mengakses perangkat keras, informasi *resources*, *service background*, mengatur alarm, dan menambahkan *notifications*, dan sebagainya. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi inti.Arsitektur aplikasi dirancang supaya kitadengan mudah menggunakan kembali komponen yang sudah digunakan.

   Bisa disimpulkan *Applications Frameworks* ini adalah *layer* di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa SMS dan panggilan telepon.

   Komponen-komponen yang termasuk di dalam *Application Frameworks* adalah sebagai berikut:

   a. *Views*
   b. *Conten Provider*
   c. *Resource Manager*
   d. *Notification Manager*
   e. *Activity Manager*

3. *Libraries*

   Adalah *layer* di mana fitur-fitur Android berada. *Layer* ini meliputi berbagai *library* C/C++ inti seperti Libc dan SSL, serta:

   a. *Libraries* media untuk pemutaran media audio dan video
   b. *Libraries* untuk manajemen tampilan
   c. *Libraries Graphics* mencakup SQL dan OpenGL untuk grafis 2D dan 3D
   d. *Libraries* SQLite untuk dukungan.

e. *Libraries* SSL dan *WebKit* terintegrasi dengan *web browser* dan *engine embedded web view*

f. *Libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API's

4. *Android Run Time*

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

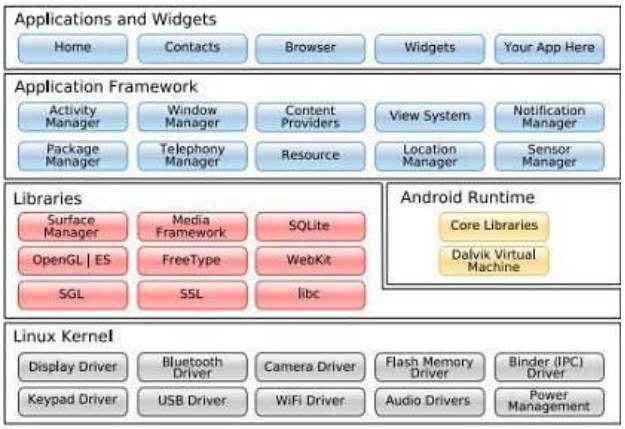
a. *Core Libraries*: Aplikasi Android dibangun dalam bahasa java, sementara *Dalvik* sebagai *virtual* mesinnya bukan *Java Virtual Machine*, sehingga diperlukan sebuah *Libraries* yang berfungsi untuk menterjermahkan bahasa java/c yang ditangani *Core Libraries*

b. *Dalvik Virtual Machine*: vitual mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, di mana merupakan pengembangan yang mampu membuat linux *kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

5. Linux *Kernel*

Adalah *Layer* di mana inti dari sistem operasi dari Android itu berada. Berisi *file-file* sistem yang mengatur sistem *processing, memory, resource, drivers,* dan sistem-sistem operasi Android lainnya. Linux *kernel* yang digunakan Android adalah Linux *kernel release* 2.6.

Arsitektur android dapat dilihat pada gambar 2.23 berikut.



Gambar 2.23 Arsitektur Android

2.2.6.3 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 system operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Sebagai *platform* aplikasi-netral, Android memberi kesempatan untuk membuat aplikasi yang kita dibutuhkan yang bukan merupakan bawaan *handphone/smartphone*. Beberapa fitur penting android:

a. *Framework* aplikasi yang mendukung penggantian komponen *reusable*.
b. *Dalvik Virtual Machine* dioptimalkan untuk perangkat *mobile.*
c. *Integrated browser* berdasarkan *engine open source WebKit*.
d. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, 3D berdasarkan sepsifikasi OpenGL ES 1.0 (Opsional akselerasi *hardware*).
e. SQLite untuk penyimpanan data.
f. *Media Support* yang mendukung audio, video, dan gambar, GSM *Telephony* (tergantung *hardware*).
g. *Bluetooth*, EDGE, 3G, dan WiFi (tergantung *hardware*).
h. Kamera, GPS, kompas, dan *accelerometer* (tergantung *hardware*).
i. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat e*mulator*, *tools* untuk *debugging*, *profil* dan kinerja memori, dan *plugin* untuk IDE *Eclipse*[4].

2.2.7 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan lainnya. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai *Java Virtual Machine* (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus di desain untuk

memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa *platform* sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

Contoh program *Hello World* yang ditulis menggunakan bahasa pemrograman Java dapat dilihat pada gambar 2.24:

```
public class HelloWorld {
public static void main(String args[]) {
System.out.println("Hello, world!");
  }}
```

Gambar 2.24 Contoh Program *Hello Word* pada java

2.2.7.1 Kelebihan Java

1. *Multiplatform*. Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform*/sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin/*bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. *Platform* yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebanya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan *bytecode* tersebut.
2. OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek)
3. *Class Libraries* yang lengkap, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini

ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

4. Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.
5. *Automatic Memory Management*, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2.2.7.2 Kekurangan Java

1. Tulis sekali, jalankan di mana saja - Masih ada beberapa hal yang tidak kompatibel antara *platform* satu dengan *platform* lain. Untuk J2SE, misalnya SWT-AWT *bridge* yang sampai sekarang tidak berfungsi pada Mac OS X.
2. Mudah di dekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET *Platform*. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/*direverse-engineer*.
3. Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun[5].

2.2.8 PHP

PHP merupakan kepanjangan dari *Hypertext Prepocessor*. PHP merupakan bahasa *scripting* yang membawa suatu *website* untuk dapat menjalankan proses sebagai berikut :

1. Mengirimkan *feedback* dari *website* secara langsung ke dalam kotal email
2. Mengirimkan email dengan *attachment* (memasang) suatu file
3. Mengupload file ke dalam halaman web
4. Watermarking gambar
5. Menghasilkan *thumbnail* dari gambar yang besar
6. Menampilkan dan meng-*update* informasi secara dinamik
7. Menggunakan *database* untuk menampilkan dan menyimpan informasi
8. Membuat *website* mudah dalam pencarian, dan lain-lain

PHP mudah dipelajari, karena merupakan *platform* yang netral, sehingga kode yang sama dapat berjalan di Windows, MAC OS, Linux. Semua *software* yang dibutuhkan dalam merancang PHP adalah opensource dan gratis.

PHP dimulai dari singkatan *Personal Home Page* di tahun 1995, tetapi setelah itu diputuskan untuk mengganti namanya setelah 2 tahun.*Personal Home Poge* terdengar seperti sebuah hobi dan tidak menjelaskan dari kecanggihan fitur yang ditambahkan. Kemudian PHP dikembangkan lebih lanjut, dengan menambahkan dukungan untuk OOP (*Object Oriented Programming*) dalam PHP 5. PHP merupakan bahasa yang memiliki atraksi yang bagus dan kita dapat menulis bahasa *scripting* dengan sangat cepat tanpa membutuhkan pembelajaran teori yang banyak.Meskipun PHP mendukung OOP, PHP bukanlah bahasa berbasis objek (OOP).

PHP adalah bahasa *server side*. artinya ini berjalan dalam *web server*, tidak seperti CSS atau javascript, PHP berjalan di sisi *client* (artinya komputer perseorangan mengunjungi website yang kita buat), ini akan memberikan kontrol yang besar, selama kode yang dipakai bekerja di dalam *server* kita, setiap perseorang menerima output tampilan yang sama. contoh, saat PHP bekerja dalam menghasilkan sebuah gambar disni dapat dilakukan dengan menggunakan javascript, tetapi apa yang pengunjung lihat tergantung kepada 2 hal yakni

ketersediaan javascritp di dalam browser terssebut dan tergantung dari versi javascript yang ada dalam browser. Ini tidak terlalu menjadi permasalahan karena proses dinamik mengambil kedalam *server* dan membuat XHTML yang dibuthkan untuk ditampilkan dalam halaman acak dalam gambar. *Server* memilih *filename* gambar dan memasukannya ke dalam tag <img> sebelum dikirim kedalam *browser*. Bahkan kamu akan melihat gambar dalam ukuran yang berbeda, karena PHP akan mendeteksi dimensi dari gambar dan memasukannya secara tepat kedalam atribut tinggi dan lebar[8].

2.2.9 MySQL

Dikembangkan dipertengahan 1990, kini menjadi teknologi yang matang yang saat ini memberikan kekuatan dan paling banyak dikunjungi sebagai tujuan saat mengakses internet. Alasan kesuksesan yang paling utama seperti PHP, yakni gratis untuk digunakan, tetapi juga sangat *powerful* dan kecepeatan yang luar biasa. Bahkan dapat dijalanan di dalam dasar *hardware* dan diletakan dengan sangat baik di sistem *resource*.MySQL juga sangat mudah di ukur, yang artinya MySQL dapat berkembang dengan baik di dalam suatu *website*. Dan faktanya perbandingan beberapa dari eWEEK, Mysql dan Oracle keseluruhannya memiliki performa terbaik dan skalabilitas terbesar.

2.2.9.1 Dasar Mysql

merupakan kumpulan terstruktur dari beberapa *record* atau data yang tersimpan di dalam sistem komputer dan terorgnisasi sehingga dapat memudahkan dalam pencarian dan informasi dapat diterima dengan cepat.

SQL dalam MySQL kepanjangan dari *Structured Query Language*. Bahasa ini terlepas dari dasar bahasa inggris dan juga dapat digunakan dalam lainnya sepeti Oracle dan MS SQLServer. Dan di desain untuk mengijinkan permintaan yang sederhana dari *database* melalui perintah seperti pada gambar 2.25 contoh *query languange* sebagai berikut :

```
SELECT title FROM publications WHERE author='Charles Dickens';
```

Gambar 2.25 Contoh *Query Languange*

MySQL *database* berisi satu atau lebih tabel, dimana setiap tabel berisi *record* atau baris, dimana setiap baris memiliki berbagai kolom atau *field* yang berisi data. Contoh *database* dengan 5 publikasi detail yaitu *Author, Title, Type, Year* pada gambar 2.26.

Author	Title	Type	Year
Mark Twain	*The Adventures of Tom Sawyer*	Fiction	1876
Jane Austen	*Pride and Prejudice*	Fiction	1811
Charles Darwin	*The Origin of Species*	Non-fiction	1856
Charles Dickens	*The Old Curiosity Shop*	Fiction	1841
William Shakespeare	*Romeo and Juliet*	Play	1594

Gambar 2.26 Contoh *Database*