BAB 2
TINJAUAN PUSTAKA

2.1. Profil Tempat Penelitian

Profil tempat penelitian yang akan dibahas pada subbab ini meliputi sejarah instansi, visi, misi dan tujuan instansi, logo instansi, badan hukum instansi, struktur organisasi instansi dan portofolio instansi.

2.1.1. Sejarah Instansi

Kelompok usaha PT. Kahar Duta Sarana (KDS) berdiri pada tanggal 16 Februari 1979 dengan badan usaha yang memasok alat tulis dan mesin-mesin kantor. PT. Kahar Duta Sarana dengan memiliki enam karyawan dan sebuah ruang kantor sederhana didaerah Senen Jakarta Pusat. PT. Kahar Duta Sarana meniti tahun-tahun pertamanya dengan ketekunan dan kerja keras.

Ketekunan dan kerja keras akhirnya membuahkan mutu dan layanan yang baik, sehingga membuat PT. Kahar Duta Sarana dipercaya memasok keperluan mesin kasir sebuah instansi pemerintah. Kepercayaan semakin bertambah dikala PT. Kahar Duta Sarana telah berhasil menyakinkan sebuah perusahaan Jepang yaitu Tokyo Electrik Corporation (TEC), untuk bermitra dan untuk pertama kalinya PT. Kahar Duta Sarana menjadi *sale distributor* mesin kasir TEC.

PT. Kahar Duta Sarana sukses memasarkan mesin kasir TEC, sehingga dipercaya untuk memasarkan produk-produk TEC yang lain, seperti: *digital scale, point of sale system,* dan lain-lain. PT. Kahar Duta Sarana pada pertengahan tahun 1980 lebih menegaskan kiprahnya untuk fokus kepada penyelenggaraan *retail system solution* dengan memasuki bidang *marketing system,* yaitu dengan memasarkan *barcode* printer dan memasuki era produksi *marking labels* pada tahun 1990. Perkembangan bisnis PT. Kahar Duta Sarana saat ini meliputi *industrial labels* yang sebagian besar pelanggannya termasuk *multinational corporation*. Produksi PT Kahar Duta Sarana telah memenuhi syarat kualitas dan mendapat sertifikasi dari UL (*Underwriting Laboratory*) USA.

Bidang pemasaran makin lengkap dengan ditunjuknya PT. Kahar Duta Sarana sebagai distributor oleh PSC Inc. untuk produk *laser scanner* dan oleh Love shaw (UK) untuk produk *large character* printer. Perkembangan yang paling akhir PT. Kahar Duta Sarana adalah telah berdirinya divisi niaga pada awal dekade 1990 yang mengayomi sejumlah outlet swalayan fotografi di beberapa tempat di Jakarta, kemudian disusul dengan berdirinya divisi-divisi lain seperti: divisi casio yang memegang agen tunggal digital camera, star mini printer dan *personal data assitant* (PDA). Memasuki dekade pertama abad 21, Indonesia dengan pasti akan berkembang menjadi pemain yang penting dalam ekonomi global. PT. Kahar Duta Sarana akan terus mengambil peran aktif dan dinamis serta berkembang terus memenuhi tuntutan zaman.

2.1.2. Logo Instansi

Logo PT. Kahar Duta Sarana dapat dilihat pada gambar 2.1.



Gambar 2.1. Logo PT. Kahar Duta Sarana

Arti dan makna logo PT. Kahar Duta Sarana adalah sebagai berikut :

1. Arti dan makna warna merah

   Warna merah mempunyai arti dan makna bahwa PT. KDS mampu menjaga kualiatas maupun kuantitas produk yang dipasarkan.

2. Arti dan makna warna biru

   Warna biru mempunyai arti dan makna bahwa PT. KDS mampu menguasai pangsa pasar.

3. Arti dan makna warna putih

Warna putih mempunyai arti dan makna bahwa PT. KDS mampu menguasai pangsa pasar tanpa melanggar hukum yang berlaku.

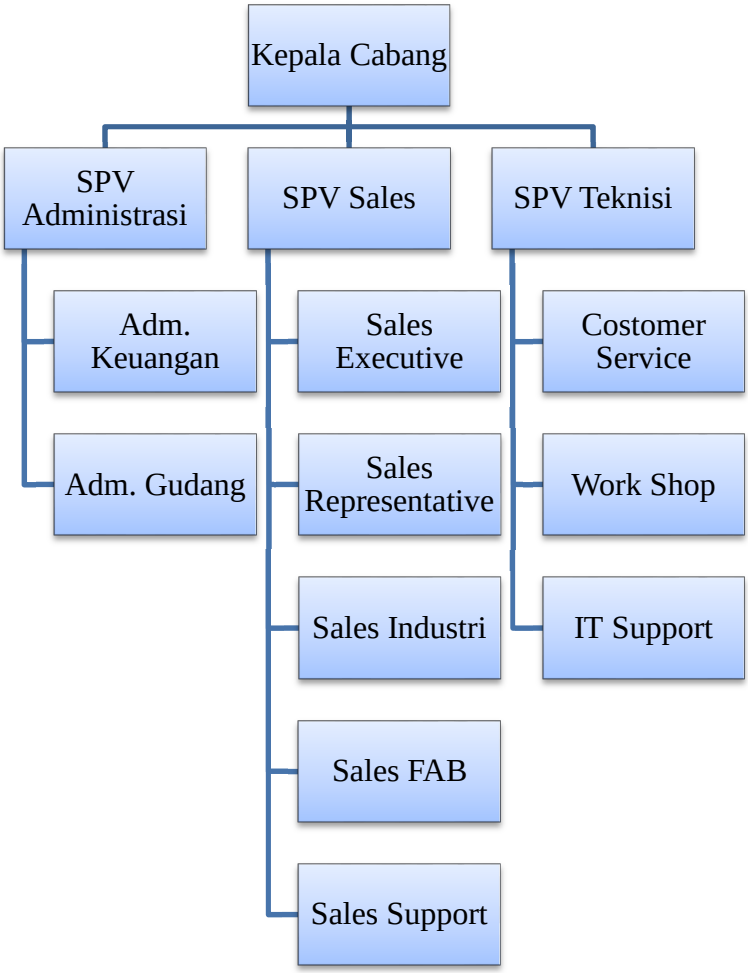
2.1.3. Badan Hukum Instansi

Berdasarkan akta notaris tanggal 16 Februari 1979 perusahaan ini disebut perseroan terbatas dengan nama PT. Kahar Duta Sarana. PT. Kahar Duta Sarana tidak akan berkembang seperti sekarang ini tanpa adanya jasa-jasa dari para pemimpin yang hingga saat ini masih menjabat. Para pemimpin yang dimaksud adalah sebagai berikut :

Chairman / CEO	: H. Koendjono Harjadi
President Director	: Frans B. Sungkono
Director	: Djoko Wirjono
Director	: Wayan Wandi
Director	: Kemal Toha
Director	: Jeffrey Ramsay

2.1.4. Struktur Organisasi Instansi

Struktur organisasi PT. Kahar Duta Sarana dapat dilihat pada gambar 2.2.



Gambar 2.2. Struktur Organisasi PT. Kahar Duta Sarana

Wewenang dan tanggung jawab adalah sebagai berikut :

1. Kepala Cabang

Mempunyai wewenang untuk memimpin seluruh karyawan di PT. KDS cabang Bandung, memasarkan seluruh produk PT. KDS dan menentukan tujuan perusahaan dalam jangka pendek maupun jangka panjang.

2. SPV Administrasi

Mengendalikan seluruh administrasi meliputi keuangan dan gudang, serta bertanggung jawab kepada kepala cabang.

a. Administrasi Keuangan

Membuat buku besar, mengatur arus kas, menyiapkan rekap kas harian, dan membuat laporan mingguan keuangan.

b. Administrasi Gudang

Mencatat keluar masuknya barang, dan membuat laporan mingguan keluar masuknya barang.

3. SPV Sales

Mengawasi seluruh kegiatan penjualan produk-produk PT. KDS yang dilaksanakan oleh sales *executive*, sales *representatif*, sales industri, sales FAB, dan sales *support*, serta bertanggung jawab kepada kepala cabang.

a. Sales *Executive*

Melakukan pendekatan, penawaran produk-produk PT. KDS kepada perusahaan *executive* dan membuat rekap penjualan.

b. Sales *Representative*

Melakukan pendekatan, penawaran produk-produk PT. KDS kepada perusahaan *representative* dan membuat rekap penjualan.

c. Sales Industri

Melakukan pendekatan, penawaran produk-produk PT. KDS kepada perusahaan industri dan membuat rekap penjualan.

d. Sales FAB

Melakukan pendekatan, penawaran produk-produk PT. KDS kepada perusahaan *food and beverage* dan membuat rekap penjualan.

e. Sales *Support*

Melaksanakan persentasi dan demo produk-produk yang ditawarkan oleh sales *executive*, sales *representative*, sales industri dan sales FAB kepada pelanggan.

4. SPV Teknisi

Mengawasi seluruh kegiatan perawatan dan perbaikan yang dilaksanakan oleh *customer service*, *work shop*, dan IT *support*, serta bertanggung jawab kepada kepala cabang.

a. *Customer Service*

Melaksanakan perawatan dan perbaikan *hardware* maupun *software* produk PT. KDS, serta membuat laporan perawatan dan perbaikan.

b. *Work Shop*

Melaksanakan perawatan dan perbaikan *hardware* maupun *software* dilingkungan PT. KDS, serta membuat laporan perawatan dan perbaikan.

c. *IT Support*

Membantu perawatan dan perbaikan yang dilaksanakan oleh *customer service* dan *work shop*.

2.1.5. Visi, Misi dan Tujuan Instansi

Visi, misi dan tujuan PT. Kahar Duta Sarana adalah sebagai berikut :

1. Visi Instansi

Menjadi perusahaan terkemuka di bidang "*Provider Retail Solution*" yang handal dan terpercaya dalam memberikan layanan kepada pelanggan baik layanan pra jual maupun purna jual dan mengutamakan kepuasan pelanggan.

2. Misi Instansi

Misi PT. Kahar Duta Sarana adalah sebagai berikut :

1. Memberikan pelayanan yang terbaik demi kepuasan pelanggan dengan kualitas pekerjaan dan integerasi sistem yang baik dan handal serta mengutamakan kecepatan, ketepatan, dan layanan purna jual yang handal dan tepat waktu
2. Meningkatkan kemampuan dan mengoptimalkan sumber daya manusia yang baik, unggul dan dapat dipercaya
3. Mengedepankan penggunaan teknologi yang tepat guna untuk mendapatkan hasil yang terbaik dan berkualitas
4. Membangun kemitraan strategis dan bersinergi dengan klien maupun partner dengan prinsip saling menguntungkan

3. Tujuan Instansi

Tujuan PT. Kahar Duta Sarana adalah sebagai berikut :

1. Kepuasan Pelanggan
2. Profesional dan Komitmen
3. Produk dan Jasa yang bernilai tinggi
4. Integritas dan Kesejahteraan Karyawan

2.1.6. Portofolio Instansi

Portofolio PT. Kahar Duta Sarana dapat dilihat pada tabel 2.1.

Tabel 2.1. Portofolio PT. Kahar Duta Sarana (KDS)

Tahun	Keterangan
1979	Berawal dari bisnis pensuplai peralatan alat tulis dan perlangkapan perkantoran, mutu dan layanan yang baik membuat PT. KDS dipercaya memasok keperluan mesin kasir sebuah instansi pemerintah dan bekerjasama dengan TEC.
1980	KDS mengembangkan tiga divisi lainya yaitu Divisi Pengembangan IT, Divisi Label dan Percetakan, dan Divisi identifikasi otomatis dan pengumpulan data (AIDC)
1990	Setelah memiliki citra yang baik di setiap divisi, KDS membangun kerjasama dengan Wincor Nixdorf, IBM, PSC, dan Casio.
2005	KDS bekerjasama dengan Bizerba, Metrologic, dan Microsoft.
2006	KDS bersertifikasi ISO 9001
2009	KDS bekerjasama dengan Hawlett Packard sebagai distributor tunggal mesin kasir HP di Indonesia.
2010	KDS mendirikan *Enterprise Solution Division* (ESD) untuk fokus pada *software enterprise* solution dan membangun kerjasama dengan Aldata.
2011	Membangun kerjasama dengan Checkpoint EAS dan Motorola Solution. KDS terus memimpin sebagai *provider* penyedia *Retail IT Solution* di Indonesia.

2.2. Landasan Teori

Landasan teori berisi teori-teori tentang ilmu yang mendasari tentang bidang yang sedang diteliti. Penyajian teori dalam landasan teori bersumber dari buku dan jurnal. Landasan teori yang akan dibahas pada subbab ini meliputi basis data, sistem informasi, sistem pendukung keputusan, *analytic hierarchy proses*

(AHP), *unified modeling language* (UML) dan arsitektur jaringan komputer berbasis *client server*.

2.2.1. Basis Data (*Data Base*)

Basis data adalah sekumpulan data yang terintegrasi dan terorganisasi untuk memenuhi kebutuhan pemakai untuk keperluan organisasi. DBMS (*Data Base Management System*) adalah perangkat lunak yang menangani semua pengaksesan ke *data base*. Basis data yang akan dibahas pada subbab ini meliputi struktur *database*, perbedaan *file* manajemen, keuntungan dan kelemahan *file* manajemen, komponen DBMS, keuntungan dan kerugian DBMS, fungsi-fungsi DBMS, model data dan lingkungan basis data [1].

2.2.1.1. Struktur *Database*

Struktur *database* terdiri dari data, record dan field [1]:

1. Data merupakan suatu fakta atau keterangan yang jika berdiri sendiri belum mempunyai arti atau nilai serta data dapat dijadikan kajian analisis atau kesimpulan. Data juga merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Pemggambaran fakta tersebut direkam dalam bentuk simbol seperti angka, tulisan atau gambar. Data biasanya terdiri dari beberapa elemen data (*data item*). Elemen data adalah unit terkecil dari data yang ada artinya bagi pihak yang menggunakannya (*user*). Data juga adalah dalam suatu sistem basis data, elemen data ini disebut dengan *field*. satu satuan informasi yang akan diolah, dimana sebelum diolah dikumpulkan di dalam suatu *file database*. Pengumpulan data dilakukan secara sistematis menurut struktur *file database* tersebut.
2. *Record* adalah data yang isinya merupakan satu kesatuan seperti Nama, Alamat, Nomor Telepon. Setiap keterangan yang mencakup Nama, Alamat dan Nomor Telepon dinamakan satu *record*. Dan setiap *record* diberi nomor urut yang disebut nomor *record* (*Record Number*). Ukuran suatu *file database* ditentukan oleh jumlah *record* yang tersimpan di dalamnya.

3. *Field* adalah sub bagian dari *Record*. Dari contoh isi *record* diatas maka terdiri dari 3 *field*, yaitu *field* Nama, *field* Alamat dan *field* Nomor Telepon.

2.2.1.2. Perbedaan, Keuntungan dan Kelemahan File Manajemen

Ada dua tipe file manajemen, file manajemen tradisional dan file manajemen data base. Berikut perbedaan dan kelemahan file manajemen seperti yang ditunjukan oleh tabel 2.2.

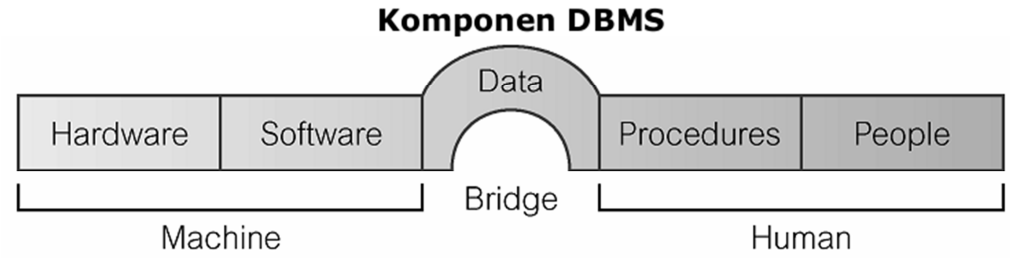
Tabel 2.2. Perbedaan dan Kelemahan File Manajemen

Perbedaan	
File manajemen tradisional	File manajemen data base
1. Program Oriented 2. Kaku 3. Kerangkapan data	1. Data Oriented 2. Luwes 3. Tidak terjadi kerangkapan data
Kelemahan	
File manajemen tradisional	File manajemen data base
1. Timbulnya data rangkap dan ketidak konsistenan data 2. Data tidak dapat digunakan secara bersama-sama 3. Kesukaran dalam mengakses data 4. Tidak fleksibel 5. Data tidak standar	1. Storage yang dibutuhkan besar 2. Dibutuhkan tenaga spesialis 3. Software mahal 4. Kerusakan pada data base dapat mempengaruhi departemen lain yang terkait kerangkapan data

Beberapa keuntungan file manajemen basis data, tidak terjadi kerangkapan data, data lebih konsisten, data dapat digunakan bersama-sama, data dapat distandarisasi, keamanan data dapat terjamin, integritas data terpelihara dan data independen.

2.2.1.3. Komponen DBMS

Komponen DBMS terdiri dari *hardware*, *software*, data, *procedures* dan *people*. Berikut komponen DBMS seperti yang ditunjukan oleh gambar 2.3.



Gambar 2.3. Komponen DBMS

Keterangan gambar :

1. *Hardware*

1. Meliputi PC sampai dengan jaringan komputer.
2. Tempat penyimpanan *secondary* (*manegtic disk*), I/O *device* ex : *disk drives*), *device Controller*, I/O *Channels*, dan lainnya.
3. *Hardware processor* dan *main memory*, digunakan untuk mendukung saat eksekusi *system software database*.

2. *Software*

DBMS, *operating system*, *network software* (jika diperlukan) dan program aplikasi pendukung lainnya.

3. Data

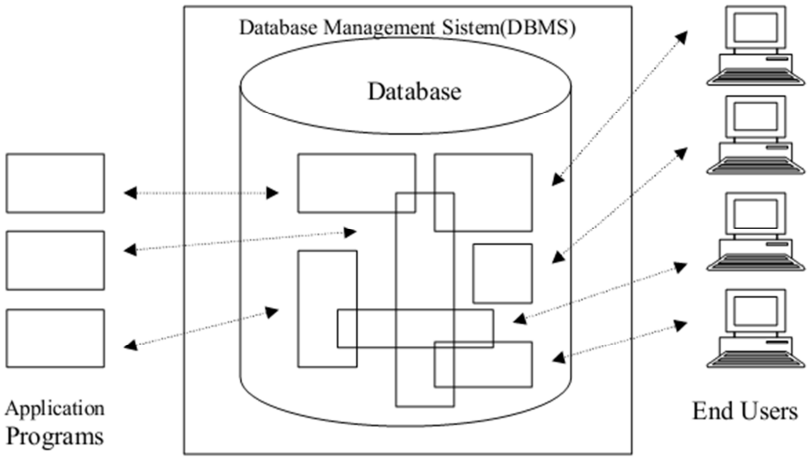
1. Data pada sebuah *system database* baik itu *single-user system* maupun *multi-user system* harus terintegrasi dan dapat digunakan bersama (*Integrated and Shared*).
2. Digunakan oleh organisasi dan deskripsi dari data disebut *schema*.

4. *Procedures*

Instruksi dan aturan yang harus disertakan dalam mendesain dan menggunakan *database* dan DBMS.

5. *People*

1. DA (*Data Administrator*), seseorang yang berwenang untuk membuat keputusan stategis dan kebijakan mengenai data yang ada
2. DBA (*DataBase Administrator*), menyediakan dukungan teknis untuk implementasi keputusan tersebut, dan bertanggung jawab atas keseluruhan kontrol sistem pada level teknis.



Gambar 2.4. Sistem basis data

3. *Database Designer* (*Logical and Physical*)
4. *Application Programmers*, bertanggung jawab untuk membuat aplikasi *database* dengan menggunakan bahasa pemrograman yang ada, seperti : C#, C++, Java, dan lainnya.
5. *End Users*, siapapun yang berinteraksi dengan sistem secara online melalui *workstation*/terminal

2.2.1.4. Kerugian DBMS

Kerugian DBMS antara lain :

1. Rumit (*Complexity*)

Karena penetapan fungsi dari DBMS yang baik, menyebabkan DBMS menjadi *software* yang cukup rumit. Seluruh user harus mengetahui fungsi-fungsi yang ada dengan baik, sehingga dapat memperoleh manfaatnya.

2. Ukuran (*Size*)

Kerumitan dan banyaknya fungsi yang ada menyebabkan DBMS memerlukan banyak *software* pendukung yang mengakibatkan penambahan tempat penyimpanan dan *memory*.

3. Biaya DBMS (*Cost of DBMS*)
4. Biaya Tambahan Perangkat keras (*Additional hardware costs*)
5. Biaya Konversi (*Cost of conversion*)

6. *Performance*

Pada dasarnya DBMS dibuat untuk menyediakan banyak aplikasi, akibatnya mungkin beberapa aplikasi akan berjalan tidak seperti biasanya.

7. *Higher impact of a failure*

Karena sistem yang terpusat, jika seluruh pengguna dan aplikasi terakses dari DBMS maka kerusakan pada bagian manapun dari sistem, akan menyebabkan operasi terhenti.

2.2.1.5. Fungsi-fungsi DBMS

Fungsi-fungsi DBMS antara lain :

1. *Data definition*

DBMS harus dapat mengolah pendefinisian data.

2. *Data manipulation*

DBMS harus dapat menangani permintaan-permintaan dari pemakai untuk mengakses data.

3. *Data security and integrity*

DBMS harus dapat memeriksa keamanan dan integriti data yang didefinisikan oleh DBA.

4. *Data recovery and concurrency*

DBMS harus dapat menangani kegagalan pengaksesan *database* yang disebabkan oleh kesalahan sistem, kerusakan disk dan sebagainya. DBMS harus dapat memantau pengaksesan data yang konkuren yaitu bila satu data diakses secara bersama-sama oleh lebih dari satu pemakai pada saat bersamaan.

5. *Data dictionary*

Tempat penyimpanan informasi yang menggambarkan data dalam *database*. *Data dictionary* disebut juga *metadata* (data mengenai data) berisi tentang :

a. Nama-nama *user* yang mempunyai wewenang untuk penggunaan DBMS
b. Nama-nama *item* data
c. Jenis-jenis dan ukuran *item* data
d. Batasan untuk masing-masing *item* data

6. *Performance*

DBMS harus dapat menangani unjuk kerja dari semua fungsi se-efisien mungkin.

2.2.1.6. Model Data

Kumpulan konsep-konsep yang terintegrasi untuk menggambarkan data, *relationships* antar data, dan batasan-batasan data dalam organisasi.

1. Data Model terdiri dari :
   a. Bagian struktural, berisikan sekumpulan aturan berdasarkan *database* yang dapat dibuat.
   b. Bagian manipulasi, mendefinisikan tipe operasi yang boleh dilakukan.
   c. Aturan-aturan Integritas.
2. Kegunaan model data antara lain :
   a. Merepresentasikan data kedalam bentuk yang lebih mudah untuk dipahami.
   b. Untuk menetapkan konsistensi dalam memandang, mengorganisir, menginterpretasikan dan memperlakukan *database*.

1. Jenis-jenis Model data

1. *Object-Based Data Models, Entity-Relationship, Semantic, Functional, Object-Oriented*.
2. *Record-Based Data Models, Relational Data Model, Network Data Model, Hierarchical Data Model*.
3. *Physical Data Models* : Menerangkan bagaimana data disimpan dalam komputer, merepresentasikan informasi seperti ; struktur *record*, permintaan *record*, dan jalur akses.
4. *Conceptual Modelling*
   a. Skema konseptual merupakan bagian utama dari sistem yang menampilkan *view* seluruh *user*.
   b. Merupakan representasi yang akurat dan lengkap dari kebutuhan data pada organisasi.
   c. Merupakan proses pembentukan suatu model informasi yang digunakan dalam organisasi yang terlepas dari detail implementasi.

d. Hasilnya merupakan model data konseptual.

2. Basis Data Relasional

Basis data relasional (*relational database*) ialah model data yang berbentuk tabel dua dimensi yang terdiri atas baris dan kolom. Keuntungan basis data relasional adalah sebagai berikut :

1. bentuknya sederhana
2. mudah melakukan berbagai operasi data

Istilah dalam basis data relasional diantaranya :

1. relasi : sebuah tabel yang terdiri dari beberapa kolom dan beberapa baris
2. *atribut/field* : kolom pada sebuah relasi
3. *tupel/record* : baris pada sebuah relasi
4. domain : kumpulan nilai yang *valid* untuk satu atau lebih atribut
5. *degree* (derajat) : jumlah atribut dalam sebuah relasi
6. *cardinality* : jumlah *tupelo* dalam sebuah relasi

3. Relational key

Relational key dibagi kedalam lima bagian antara lain *super key*, *candidate key*, *primary key*, *alternate key* dan *foreign key* :

1. *Super key*

   Satu atau kumpulan atribut yang secara unik mengidetifikasi sebuah tupel didalam relasi.
2. *Candidate key*

   Atribut didalam relasi yang biasanya mempunyai nilai unik.
3. *Primary key*

   Candidate key yang dipilih untuk mengidentifikasikan tupel secara unik dalam relasi.
4. *Alternate key*

   Candidate key yang tidak terpilih sebagai *primary key*.
5. *Foreign key*

   Merupakan sembarang *atribut* yang menunjuk kepada *primary key* pada tabel lain.

4. Atribut

Atribut dengan domain yang sama yang menjadi kunci utama pada sebuah relasi tetapi pada relasi lain *atribut* tersebut hanya menjadi *atribut* biasa. *Relational integrity role* dibagi kedalam tiga bagian antara lain *null, entity integrity* dan *referential integrity* :

1. *Null*

Nilai suatu atribut yang tidak diketahui dan tidak cocok untuk tupel tersebut.

2. *Entity Integrity*

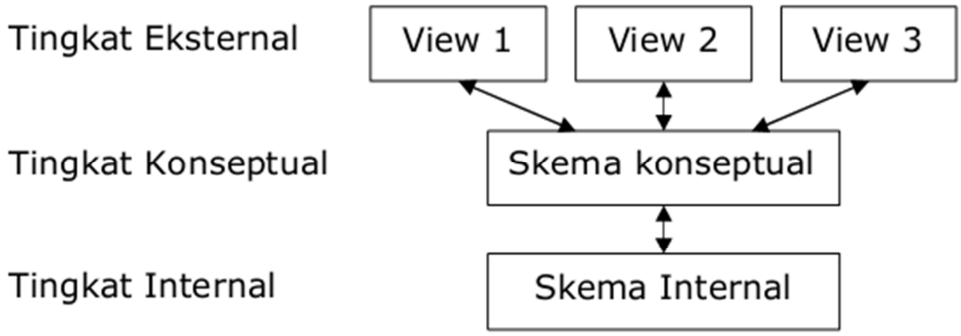
Tidak ada satu komponen *primary key* yang bernilai *null*.

3. *Referential Integrity*

Suatu domain dapat dipakai sebagai kunci utama bila merupakan *atribut* tunggal pada domain yang bersangkutan.

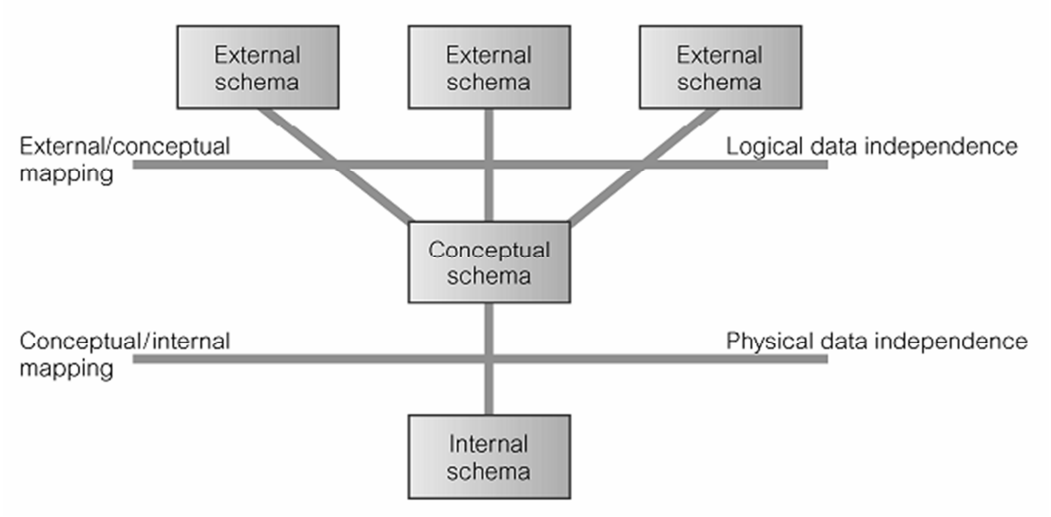
2.2.1.7. Lingkungan basis data

Lingkungan basis data satu seperti yang ditunjukan oleh gambar 2.5.



Gambar 2.5. Lingkungan basis data (1)

Tingkat *eksternal* yaitu menerangkan view *data base* dari sekelompok pemakai. Tingkat konseptual yaitu menerangkan informasi *database* secara menyeluruh dengan menyembunyikan data secara fisik. Tingkat *internal* yaitu menerangkan struktur penyimpanan database secara fisik. Lingkungan basis data dua seperti yang ditunjukan oleh gambar 2.6.



Gambar 2.6. Lingkungan basis data (2)

Data independen : kapasitas untuk mengubah skema pada satu level sistem basis data tanpa mengubah skema pada level lain.

1. *Physical Data Independent*, perubahan pada skema internal tidak mempengaruhi skema lain.
2. *Logical Data Independent*, perubahan pada skema konseptual tidak mempengaruhi skema lain.

Mapping (*Transformasi*), proses pendefinisian informasi dari satu level ke level lainnya.

1. Konseptual /*internal mapping*, pendefinisian hubungan antara view konseptual dengan *database* dilevel internal.
2. *Eksternal* /konseptual *mapping*, pendefinisian hubungan antara *view* konseptual dengan *database* dilevel *eksternal*.

2.2.2. Sistem Informasi

Sistem informasi adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algoritmik, data, dan

teknologi. Sistem informasi yang akan dibahas pada subbab ini meliputi pengertian sistem, konsep dasar informasi, konsep dasar sistem informasi [8].

2.2.2.1. Pengertian Sistem

Sistem berasal dari bahasa Yunani *Systema* yang memiliki pengertian yang luas, tidak ada suatu definisi yang pasti, terdapat dua kelompok pendekatan di dalam pendefinisian sistem, yaitu yang menekankan pada prosedur dan menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur, mendefinisikan sistem sebagai berikut [8]:

Sistem adalah kumpulan/group dari subsistem/bagian/komponen apapun baik fisik maupun *non*-fisik yang saling berhubungan antara satu sama lain dan bekerja sama secara harmonis untuk mencapai suatu tujuan tertentu.

1. Konsep Dasar Sistem

Suatu sistem adalah jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul, bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Komponen–komponen atau subsistem-subsistem dalam suatu sistem tidak dapat berdiri lepas sendiri-sendiri. Komponen–komponen atau subsistem-subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai.

Suatu sistem mempunyai maksud tertentu, ada yang menyebut maksud dari suatu sistem adalah untuk mencapai suatu tujuan (*goal*) dan ada yang menyebutkan untuk mencapai sasaran (*objective*). Suatu sistem pada dasarnya merupakan suatu susunan yang teratur dari kegiatan yang berhubungan satu sama lainnya dan prosedur-prosedur yang berkaitan yang melaksanakan dan memudahkan pelaksanaan kegiatan utama dari suatu organisasi.

2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu.

Karakteristik sistem adalah sebagai berikut :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem, atau bagian-bagian dari sistem.

2. Batas Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menujukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan menggangu kelangsungan hidup dari sistem.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem

Masukan adalah data yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem

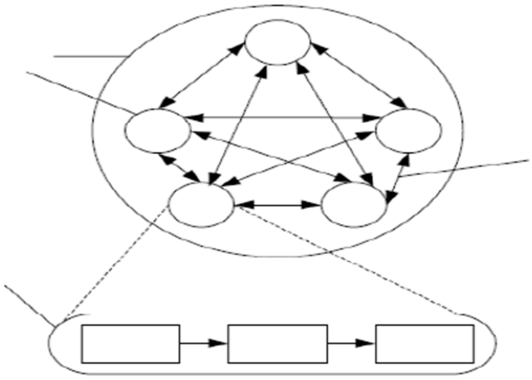
Keluaran adalah hasil dari data yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari system sangat menetukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Karakteristik sistem dapat dilihat pada gambar 2.7.



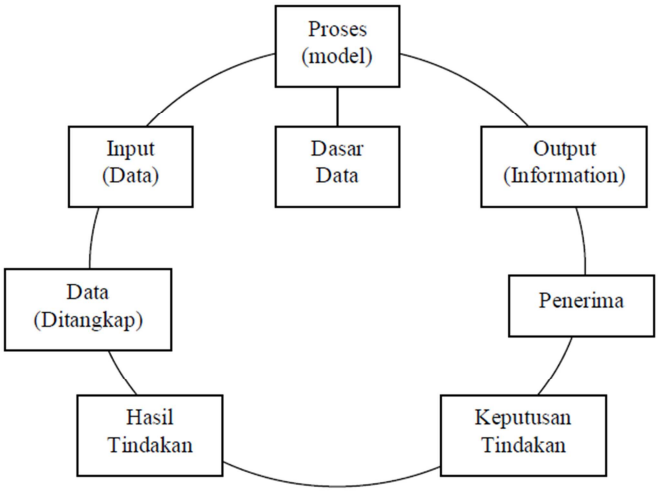
Gambar 2.7. Karateristik Sistem

2.2.2.2. Konsep Dasar Informasi

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Informasi merupakan data yang telah diklasifikasikan atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

1. Siklus Informasi

Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi. Data yang diolah melalui model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus informasi dapat dilihat pada gambar 2.8.



Gambar 2.8. Siklus Informasi

2.2.2.3. Konsep Dasar Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan.

Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah,

mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya [8].

1. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari komponen *input*, komponen model, komponen *output*, komponen teknologi, komponen *hardware*, komponen *software*, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

1. Komponen *input*

Input mewakili data yang masuk kedalam sistem informasi. *Input* disini termasuk metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumen dasar.

2. Komponen model

Komponen ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen *output*

Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen teknologi

Teknologi merupakan *toolbox* dalam sistem informasi, teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

5. Komponen *hardware*

Hardware berperan penting sebagai suatu penyimpanan vital bagi sistem informasi. Yang berfungsi sebagai tempat untuk menampung *database* atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

6. Komponen *software*

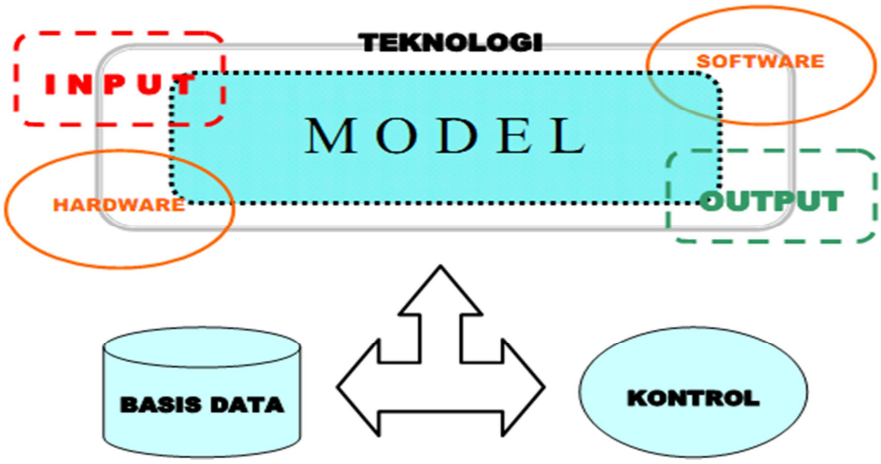
Software berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari hardware untuk menciptakan suatu informasi.

7. Komponen basis data

Basis data merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*).

8. Komponen kontrol

Banyak hal yang dapat merusak informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan, kegagalan sistem itu sendiri, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi. Komponen sistem informasi dapat dilihat pada gambar 2.9.



Gambar 2.9. Komponen Sistem Informasi

2. Elemen sistem informasi

Sistem informasi terdiri dari elemen-elemen yang terdiri dari orang, prosedur, perangkat keras, perangkat lunak, basis data, jaringan komputer dan komunikasi data. Semua elemen ini merupakan komponen fisik.

1. Orang

Orang atau personil yang dimaksudkan yaitu operator komputer, analis sistem, programmer, personil data *entry*, dan manajer sistem informasi.

2. Prosedur

Prosedur merupakan elemen fisik. Hal ini disebabkan karena prosedur disediakan dalam bentuk fisik seperti buku panduan dan instruksi. Ada 3 jenis prosedur yang dibutuhkan, yaitu instruksi untuk pemakai, instruksi untuk penyiapan masukan, instruksi pengoperasian untuk karyawan pusat komputer.

3. Perangkat keras

Perangkat keras bagi suatu sistem informasi terdiri atas komputer(pusat pengolah, unit masukan/keluaran), peralatan penyiapan data dan terminal masukan/keluaran.

4. Perangkat lunak

Perangkat lunak dapat dibagi dalam tiga jenis utama:

a. Sistem perangkat lunak umum, seperti sistem pengoperasian dan sistem manajemen data yang memungkinkan pengoperasian sistem komputer.
b. Aplikasi perangkat lunak umum, seperti model analisis dan keputusan.
c. Aplikasi perangkat lunak yang terdiri atas program yang secara spesifik dibuat untuk setiap aplikasi.

5. Basis data

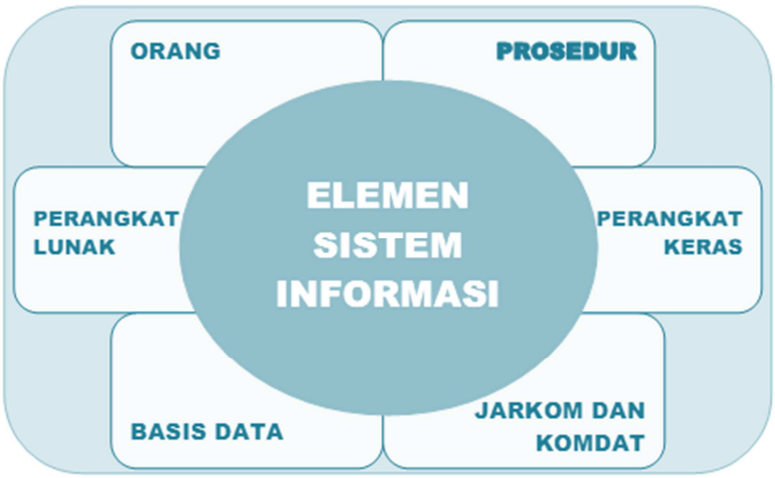
File yang berisi program dan data dibuktikan dengan adanya media penyimpanan secara fisik seperti *hard disk, magnetic tape* dan sebagainya. File juga meliputi keluaran tercetak dan catatan lain diatas kertas, mikro film dan lain sebagainya.

6. Jaringan komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data.

7. Komunikasi data

Komunikasi data adalah merupakan bagian dari telekomunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer-komputer dan piranti-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu sistem informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer-komputer dapat berkomunikasi satu sama lain. Elemen sistem informasi dapat dilihat pada gambar 2.10.



Gambar 2.10. Elemen Sistem Informasi

3. Arsitektur Sistem Informasi

Sistem informasi dapat dibentuk sesuai kebutuhan organisasi masing-masing. Oleh karena itu, untuk dapat menerapkan sistem yang efektif dan efisien diperlukan perencanaan, pelaksanaan, pengaturan dan evaluasi sesuai keinginan masing-masing organisasi. Guna dari sistem yang efektif dan efisien tidak lain

untuk mendapatkan keunggulan dalam berkompetisi. Semua orang dapat menggunakan sistem informasi dalam organisasi, tetapi faktor efisiensi setiap sistem adalah berbeda. Perlu diketahui, perubahan sistem, baik besar maupun kecil, selalu akan melalui tingkatan-tingkatan sebagai berikut:

Tingkat I : Ide, mengetahui perlu adanya perubahan.
Tingkat II : Desain, merancang cara pemecahannya.
Tingkat III : Pelaksanaan, menerapkan design kedalam sistem.
Tingkat IV : Kontrol, memeriksa tingkat pelaksanaan dijalankan sesuai dengan desain.
Tingkat V : Evaluasi, memeriksa apakah perubahan yang terjadi sesuai dengan tujuan semula.
Tingkat VI : Tindak lanjut, melaksanakan perubahan sesuai dengan hasil evaluasi yang ada.

Adapun tingkatan kunci yang digunakan untuk memecahkan bagian masalah baik itu secara menyeluruh maupun per bagian dapat dilihat pada gambar 2.11.



Gambar 2.11. Tingkatan kunci untuk memecahkan masalah

4. Klasifikasi Sistem Informasi

Sistem informasi merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya :

1. Sistem abstrak atau sistem fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem *teologia*, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan tuhan.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi. Sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin, yang disebut *human machine system*. Sistem informasi berbasis internet merupakan contoh *human machine system* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem *deterministik* dan sistem *probabilistik*

Sistem *deteministik* adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksi. Sistem *probabilistik* adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur *probabilistik*.

4. Sistem terbuka dan sistem tertutup

Sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya. Sistem tertutup adalah sistem yang tidak terhubung dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar.

2.2.3. Sistem Pendukung Keputusan

Sistem pendukung keputusan (SPK) menurut ken dan scott-marton merupakan "pasangan dari intelektual sumber daya manusia dengan kemampuan dari komputer untuk memperbaiki kualitas dari keputusan,yaitu sistem pendukung keputusan yang terkomputerisasi bagi pembuat keputusan manajemen yang menghadapi masalah semi terstruktur."

Definisi dari sistem pedukung keputusan adalah sistem yang menyediakan sarana bagi para manajer untuk mengembangkan informasi sesuai dengan keputusan yang akan dibuat. Dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah semi terstruktur yang spesifik. Sistem pendukung keputusan yang akan dibahas pada subbab ini meliputi fase proses pengambilan keputusan, tingkat teknologi SPK,

karateristik SPK, komponen–komponen SPK, pemodelan analisis, tahapan pemodelan, metode keputusan [3].

2.2.3.1. Fase Proses Pengambilan Keputusan

Pada umumnya para ahli sependapat bahwa keputusan (*decision*) berarti pilihan (*choice*), yaitu pilihan dari dua atau lebih kemungkinan. Selain itu, keputusan dapat dilihat pada kaitannya dengan proses lebih dinamis yang diberi label pengambilan keputusan. Keputusan di pandang sebagai proses karena terdiri atas satu aktivitas yang berkaitan dan tidak hanya dianggap sebagai tindakan bijaksana. Dengan kata lain, keputusan merupakan sebuah kesimpulan yang di capai setelah dilakukan pertimbangan, yang terjadi setelah salah satu kemungkinan di pilih, sementara yang lain di kesampingkan.dalam hal ini yang dimaksuk dengan pertimbangan ialah menganalisis beberapa kemungkinan atau aternatif satu diantaranya.

1. Penelusuran (*intellegence*)

   Tahap ini merupakan proses penelusuran dan pendektesian dari lingkup problematika serta pengenalan masalah. Data masukan diperoleh, diproses dan di uji dalam rangka mengindentifikasi masalah.

2. Perancangan (*Design*)

   Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif yang bisa dilakukan.tahap ini meliputi proses menganalisis masalah, menurunkan solusi dan menguji kelayakan solusi.

3. Pemilihan (*choice*)

   Dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin di Jalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilann keputusan.

4. Implementasi (*implementation*)

   Tahap sebenarnya adalah dari tiga tahap yang merupakan pelaksanaan dari keputusan yang diambil.

2.2.3.2. Tingkat Teknologi SPK

Pembangunan SPK berdasarkan perangkatnya, mencakup tiga tingkat perangkat keras atau lunak. Ketiganya digunakan berdasarkan perbedaan kemampuan teknik, dan perbedaan tugas yang akan dikerjakan. Ketiga tingkatan yang dimaksud adalah:

1. Sistem pendukung keputusan khusus (*Spesific DSS*)

   SPK khusus adalah perangkat keras atau lunak yang memungkinkan pembuat keputusan menyelesaikan sekumpulan masalah yang saling berhubungan.

2. Pembangkit sistem pendukung keputusan (*DSS Generator*)

   Pembangkit SPK adalah suatu paket perangkat keras dan lunak yang mempunyai kemampuan untuk mengembangkan SPK khusus secara cepat dan mudah. *DSS Generator* diantaranya meliputi fasilitas penyiapan laporan, tampilan grafik dan sebagainya.

3. Perlengkapan sistem pendukung keputusan (*DSS Tools*)

   Peralatan SPK merupakan tingkat teknologi yang paling mendasar dalam pengembangan SPK. peranan SPK adalah elemen-elemen perangkat keras dan lunak yang digunakan untuk mengembangkan SPK spesifik maupun pembangkit SPK. Yang termasuk dalam kategori-kategori teknologi ini anatara lain bahasa pemograman, sistem operasi komputer, perangakat lunak pengakses data, dan sebagainnya.

2.2.3.3. Karateristik SPK

Peranan SPK dalam konteks keseluruhan sistem informasi ditunjukan untuk memperbaiki kinerja melalui aplikasi teknologi informasi. Terdapat tujuh karateristik dasar SPK yang efektif yaitu:

1. Mendukung proses pengambilan keputusan, menitikberatkan pada *management by perception*.
2. Adanya *interface* manusia /mesin dimana manusia (*user*) tetap mengontrol proses pengambilan keputusan.

3. Mendukung pengambilan keputusan untuk masalah-masalah terstruktur, semi terstruktur dan tidak terstruktur.
4. Memiliki kapabilitas dialog untuk memperoleh informasi sesuai dengan kebutuhan model interaktif.
5. *Output* ditunjukan untuk personil organisasi dalam semua tingkatan.memiliki subsistem-subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.
6. Pendekatan *easy to use*. Ciri suatu SPK yang efektif adalah kemudahannnya untuk digunakan, dan memungkinkan keleluasaan pemakai untuk memilih atau mengembangkan pendekatan-pendekatan baru dalam membahas masalah yang dihadapi.
7. Kemampuan sistem beradaptasi secara cepat, dimana pengambilan keputusan dapat menghadapi masalah-masalah baru, dan pada saat yang sama dapat menanganinya dengan cara mengadaptasi sistem terhadap kondisi-kondisi perubahan yang terjadi.

2.2.3.4. Komponen–komponen SPK

Suatu SPK memiliki tiga subsistem utama yang menentukan kapabilitas teknis SPK tersebut yaitu :

1. Manajemen Basis Data (*Data Base Manajemen System*)

Sub sistem data merupakan komponen SPK penyedia data bagi sistem.data ini disimpan dalam *database* yang diorganisasikan oleh DBMS (*Data Base Manajemen System*).

2. Manajemen Bisnis Model (*Model base*)

Model adalah peniruan dari alam nyata. Model ini dikelola oleh model base.

3. Manajemen Penyelenggara Dialog (*User System Interface*)

Melalui sistem dialog inilah sistem diimplementasikan sehingga pengguna atau pemakai dapat berkomunikasi dengan sistem yang dirancang.

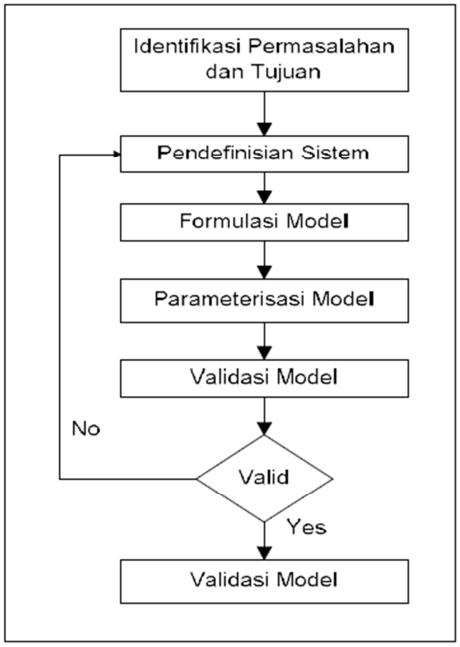
2.2.3.5. Pemodelan Analisis

Pada tingkat teknik, rekayasa perangkat lunak dimulai dengan serangkaian tugas pemodelan yang membawannya kepada suatu spesifikasi lengkap dari persyaratan reprensi dan reprensi desian yang konprehensif bagi perangkat lunak yang akan dibangun. Model analisis sebenarnya serangkaian model, yang merupakan reprensi teknis yang pertama dari sistem.

2.2.3.6. Tahapan Pemodelan

Pemodelan pada dasarnya merupakan proses membangun atau membebtuk sebuah model, dalam bahasa formal tertentu, dari suatu sistem nyata berdasarsakan sudut pandang tertentu menurut.sistem nyata akan dilihat dan dibaca oleh pemodelan dan membentuk citra atau gambaran tertentu di salam pikirannya.

Pemodelan dilakukan menurut beberapa tahapan seperti yang ditunjukan oleh gambar 2.12. Tahapan pemodelan ini menjadi arah bagi pemodel untuk membuat model yang memiliki karakter dengan tingkat generelisasi tinggi, mekanisme transparan, berpotensi untuk dikembangkan peneliti lain, dan peka terhadap perubahan asumsi.



Gambar 2.12. Tahapan Pemodelan

2.2.3.7. Metode Keputusan

Model keputusan relevan dengan model secara umum, model didefinisikan sebagai reprensi sederhana dari suatu keadaan nyata.

1. Pengambilan keputusan kriteria majemuk

Pengambilan keputusan kriteria majemuk pada prinsipnya menurut ramdhani adalah sebagai berikut :

"*Model pengambilan keputusan untuk penentuan prioritas alternatife dengan menggunakan dua atau lebih kriteria atau atribut, yang satu sama lain terkadang memiliki konflik dan kriteria yang tidak sepadan untuk beberapa kepentingan kelompok*".

Lebih lanjut lagi, menurut ramdhani menyatakan penggunaan model untuk pengambilan keputusan kriteria majemuk untuk suatu keputusan tertentu tergantung pada saat pemilihan kriteria, pengambilan keputusan harus mencoba untuk menggambarkan dalam bentuk kuantifikasi jika hal ini memungkin, karena akan selalu ada faktor yang tidak dapat di kuantifikasikan yang juga tidak dapat diabaikan.

Bila diabaikan maka hal ini dapat mengakibatkan semakin sulitnnya membuat perbandingan, kenyataan bahwa kriteria yang baik tidak bisa dikuantifikasikan itu sukar untuk di perkirakan dan perbandingan hendaknya tidak dapat menyebabkan pengambilan keputusan. Untuk tidak menggunakan kriteria tersebut, karena kriteria ini dapat saja relevan dengan masalah utama di dalam setiap analisis.

Beberapa kriteria yang mungkin sangat penting, tetapi sulit di kualifikasikan adalah seperti faktor–faktor sosial (gangguan lingkungan), estetika, keadilan, faktor–faktor politis, serta kelayakan pelaksanaan, akan tetapi jika suatu kriteria dapat dikualifikasikan tanpa merubah pengertiannya, maka hal ini dapat dilakukan.

2. Penentuan kriteria

Sifat-sifat yang harus diperhatikan dalam memilih kriteria pada setiap persoalan pengambilan keputusan adalah sebagai berikut :

1. Lengkap

Kriteria yang dipilih harus dapat mencakup seluruh aspek penting dalam persoalan tersebut. Suatu set kriteria disebut lengkap apabila set ini dapat menunjukan seberapa jauh seluruh tujuan yang dicapai.

2. Operasional

Kriteria yang baik harus dapat digunakan dalam analisis. Sifat operasional ini mencakup beberapa pengertian, antara lain bahwa set kriteria ini harus mempunyai arti bagi pengambilan keputusan, sehingga ia dapat benar-benar menghayati implikasinya terhadap alternatif yang ada.selain itu, jika tujuan pengambilan keputusan ini harus dapat digunakan sebagai sarana untuk memberikan penjelasan atau untuk berkomunikasi.

3. Tidak berlebihan

Kriteria yang dipilih tidak berlebihan untuk menghindari perhitungan yang berulang. Proses menentukan set kriteria diusahakan menghindari kriteria yang mengandung pengertian yang sama.

4. Minimum

Jumlah kriteria harus minimum dengan tujuan agar lebih mengkonpreherasikan untuk dapat menghayati permasalahan dengan baik, lebih jauh lagi, jumlah perhitungan yang diperlukan dalam analisis akan semakin banyak.

Ada kalanya meskipun kita telah berusaha menjabarkan tujuan menjadi lebih spesifik, kita tetap dapat menemukan kriteria untuk sejumlah tujuan.

3. Jenis Metode Pengambilan Keputusan (PK) Kriteria Majemuk

Ada beberapa metode standar yang umum digunakan untuk pengambilan keputusan kriteria majemuk adalah *Multi Attribute Utilty Theory* (MAUT) (Edward,W.1997), *Simple Multi Attribut Rating Tecnique* (SMART) (Edwar, Wdan Barron,FH,1994) dan *Analytic Hierarcy Process* (AHP) (saaty,TL,1980). Perkembangan ilmu pengambilan keputusan kriteria majemuk juga telah meluas dengan diperkenalkan metode yang lebih kompleks seperti *Fuzzy Analytic Hierarcy Process* (F-AHP), *Analytic Network Process* (ANP) dan yang lainnya.

Penelitian ini mengambil basis metode AHP sebagai metode untuk memecahkan permasaahan yang dihadapi dalam menentukan kelayakan rekrutmen pegawai.

2.2.4. *Analytic Hierarchy Proses* (AHP)

Proses pengambilan keputusan pada dasarnya adalah memilih suatu alternatif. Peralatan utama AHP adalah sebuah hirarki fungsional dengan input utamannya persepsi manusia dengan hirarki, suatu masalah kompleks dan tidak terstruktur dipecahkan ke dalam kelompok-kelompoknya. Kemudian kelompok-kelompok tersebut diatur menjadi suatu bentuk hirarki.

AHP adalah suatu model yang memecah-mecah suatu situasi yang kompleks, tidak terstruktur, kedalam bagian-bagian komponennya. Menata bagian atau variabel kedalam susunan hirarki, memberi nilai numerik pada pertimbangan subjektif tentang relatif pentingnya setiap variabel, dan mensintesis berbagai pertimbangan untuk menetapkan variabel mana yang memiliki prioritas paling tinggi dan bertindak untuk mempengaruhi hasil pada situasi tersebut [3].

AHP sering digunakan sebagai metode pemecahan masalah dibanding dengan metode yang lain karena alasan-alasan sebagai berikut [3]:

1. Struktur yang berhirarki, sebagai konsekuesi dari kriteria yang dipilih, sampai pada subkriteria yang paling dalam.
2. Memperhitungkan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh pengambil keputusan.
3. Memperhitungkan daya tahan output analisis sensitivitas pengambilan keputusan.

Analytic hierarchy proses (AHP) yang akan dibahas pada subbab ini meliputi kelebihan dan kelemahan AHP, tahapan AHP, prinsip dasar dan aksioma AHP, menghitung *Analytic Hierarchy Proses* (AHP).

2.2.4.1. Kelebihan dan Kelemahan AHP

Layaknya sebuah metode analisis, AHP pun memiliki kelebihan dan kelemahan dalam sistem analisisnya. Kelebihan metode AHP adalah sebagai berikut [3]:

1. Kesatuan (*Unity*)

   AHP membuat permasalahan yang luas dan tidak terstruktur menjadi suatu model yang fleksibel dan mudah dipahami.

2. Kompleksitas (*Complexity*)

   AHP memecahkan permasalahan yang kompleks melalui pendekatan sistem dan pengintegrasian secara deduktif.

3. Saling ketergantungan (*Inter Dependence*)

   AHP dapat digunakan pada elemen-elemen sistem yang saling bebas dan tidak memerlukan hubungan linier.

4. Struktur Hirarki (*Hierarchy Structuring*)

   AHP mewakili pemikiran alamiah yang cenderung mengelompokkan elemen sistem ke level-level yang berbeda dari masing-masing level berisi elemen yang serupa.

5. Pengukuran (*Measurement*)

   AHP menyediakan skala pengukuran dan metode untuk mendapatkan prioritas.

6. Konsistensi (*Consistency*)

   AHP mempertimbangkan konsistensi logis dalam penilaian yang digunakan untuk menentukan prioritas.

7. Sintesis (*Synthesis*)

   AHP mengarah pada perkiraan keseluruhan mengenai seberapa diinginkannya masing-masing alternatif.

8. Trade Off

   AHP mempertimbangkan prioritas relatif faktor-faktor pada sistem sehingga orang mampu memilih altenatif terbaik berdasarkan tujuan mereka.

9. Penilaian dan Konsensus (*Judgement and Consensus*)

AHP tidak mengharuskan adanya suatu konsensus, tapi menggabungkan hasil penilaian yang berbeda.

10. Pengulangan Proses (*Process Repetition*)

AHP mampu membuat orang menyaring definisi dari suatu permasalahan dan mengembangkan penilaian serta pengertian mereka melalui proses pengulangan.

Sedangkan kelemahan metode AHP adalah sebagai berikut :

1. Ketergantungan model AHP pada input utamanya. Input utama ini berupa persepsi seorang ahli sehingga dalam hal ini melibatkan subyektifitas sang ahli selain itu juga model menjadi tidak berarti jika ahli tersebut memberikan penilaian yang keliru.
2. Metode AHP ini hanya metode matematis tanpa ada pengujian secara statistik sehingga tidak ada batas kepercayaan dari kebenaran model yang terbentuk.

2.2.4.2. Tahapan Metode AHP

Tahapan metode AHP dapat dilihat pada langkah-langkah sebagai berikut [3] :

1. Membuat struktur hierarki yang diawali dengan tujuan utama.

Setelah menyusun tujuan utama sebagai level teratas akan disusun level hirarki yang berada di bawahnya yaitu kriteria-kriteria yang cocok untuk mempertimbangkan. Tiap kriteria mempunyai intensitas yang berbeda-beda. Hirarki dilanjutkan dengan subkriteria (jika mungkin diperlukan).

2. Menentukan skala penilaian perbandingan berpasangan terhadap tiap-tiap kriteria

Skala penilaian perbandingan berpasangan dan maknanya yang diperkenalkan oleh Saaty bisa dilihat di tabel 2.3. Skala Penilaian Perbandingan Berpasangan.

Tabel 2.3. Skala Penilaian Perbandingan Berpasangan

Intensitas Kepentingan	Keterangan	Penjelasan
1	Kedua elemen sama pentingnya	Dua elemen mempunyai pengaruh yang sama besar terhadap tujuan
3	Elemen yang satu sedikit lebih penting dari pada elemen lain	Pengalaman & penilaian sedikit menyokong satu elemen dibandingkan elemen lainnya
5	Elemen yang satu lebih penting dari yang lainnya	Pengalaman & penilaian sangat kuat menyokong satu elemen dibandingkan elemen lainnya
7	Satu elemen jelas lebih mutlak penting dari elemen yang lainnya	Satu elemen yang kuat disokong dibawa terlihat pendek
9	Satu elemen mutlak penting dari elemen yang lainnya	Bukti yang mendukung elemen yang satu terhadap elemen lain memiliki tingkat penerangan perhitungan yang mungkin mengembalikan
2, 4, 6, & 8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan	Nilai yang di berikan bila ada dua kompromi diantara dua pilihan
Kebalikan	Jika untuk aktivitas I mendapat satu angka di bandingkan dengan aktivitas J, maka J mempunyai nilai kebelikannya dibandingkan dengan I	

3. Membuat matrik perbandingan berpasangan sebanyak n x $\left[\frac{(n-1)}{2}\right]$ buah, dengan n adalah banyaknya kriteria yang dibandingkan.

Matriks yang digunakan bersifat sederhana, memiliki kedudukan kuat untuk kerangka konsistensi, mendapatkan informasi lain yang mungkin dibutuhkan dengan semua perbandingan yang mungkin dan mampu menganalisis kepekaan prioritas secara keseluruhan untuk perubahan pertimbangan. Pendekatan dengan matriks mencerminkan aspek ganda dalam prioritas yaitu mendominasi dan didominasi. Perbandingan dilakukan berdasarkan *judgment* dari pengambil keputusan dengan menilai tingkat kepentingan suatu kriteria dibandingkan kriteria lainnya. Untuk memulai proses perbandingan berpasangan dipilih sebuah kriteria dari level paling atas hirarki. Jika suatu subsistem operasi terdapat n kriteria, yaitu kriteria-kriteria operasi $A_1, A_{2,} A_3$....., A_{n}, maka hasil perbandingan secara

berpasangan kriteria-kriteria operasi tersebut akan membentuk matriks perbandingan. Perbandingan berpasangan dimulai dari tingkat hirarki paling tinggi, dimana suatu kriteria digunakan sebagai dasar pembuatan perbandingan. Kriteria yang akan di bandingkan ditunjukan pada tabel 2.4. Matriks Perbandingan Berpasangan.

Tabel 2.4. Matriks Perbandingan Berpasangan

C	A_1	A_2	A_3	…..	A_n
A_1	1	S_{12}	S_{13}	…..	S_{1n}
A_2	$\frac{1}{S_{12}}$	1	n_{23}	…..	S_{2n}
A_3	$\frac{1}{S_{13}}$	$\frac{1}{S_{23}}$	1	…..	S_{3n}
…..	…..	…..	…..	…..	…..
A_n	$\frac{1}{S_{1n}}$	$\frac{1}{S_{2n}}$	$\frac{1}{S_{3n}}$	…..	1

4. Menghitung normalisasi matriks perbandingan berpasangan

   Bagilah setiap angka didalam sebuah kolom dari matriks perbandingan berpasangan dengan jumlah kolomnya.

5. Menghitung vektor prioritas

   Hitung rata-rata setiap baris dalam normalisasi matriks. Hal ini membentuk vektor prioritas dari pilihan alternatif dikaitkan dengan kriteria tertentu, jumlah vektor adalah 1.

6. Memeriksa konsistensi hirarki.
   a. Kalikan setiap kolom dari matrik perbandingan dengan vektor prioritas,
   b. Hasil perkalian matrik setiap kolom dibagi dengan vektor prioritas lalu dijumlahkan,
   c. Menghitung rata-rata maximal (λ_{max}) dengan membagi hasil penjumlahan dari tahap b dibagi dengan banyaknya kriteria,

d. Menghitung *Consistence Index*

$CI = (\lambda_{max} - n)/(n - 1)$ ..... (1)

Keterangan :

CI = *Consistence Index*

λ_{max} = Nilai rata-rata dari kriteria/subkriteria

n = Banyaknya kriteria/subkriteria

e. Menghitung *Consistency Ratio*

CR = CI / RI ..... (2)

Keterangan :

CR = *Consistency Ratio*

CI = *Consistence Index*

RI = *Random Index* (dapat dilihat pada tabel 2.5.)

Konsistensi yang diharapkan adalah yang mendekati sempurna agar menghasilkan keputusan yang mendekati valid. Walaupun sulit untuk mencapai yang sempurna, *Consistency Ratio* diharapkan kurang dari atau sama dengan 10 %.

Tabel 2.5. Indeks Random

Ukuran Matriks	Indeks Random (Inkonsistensi)
1.2	0.00
3.0	0.58
4.0	0.90
5.0	1.12
6.0	1.24
7.0	1.32
8.0	1.41
9.0	1.45
10.0	1.49
11.0	1.51
12.0	1.48
13.0	1.56
14.0	1.57
15.0	1.59

2.2.4.3. Prinsip Dasar dan Aksioma AHP

AHP didasarkan atas tiga prinsip dasar yaitu [3]:

1. Dekomposisi

Dengan prinsip ini struktur masalah yang kompleks dibagi menjadi bagian-bagian secara hierarki. Tujuan didefinisikan dari yang umum sampai khusus. Dalam bentuk yang paling sederhana struktur akan dibandingkan tujuan, kriteria dan level alternatif. Tiap himpunan alternatif mungkin akan dibagi lebih jauh menjadi tingkatan yang lebih detail, mencakup lebih banyak kriteria yang lain. Level paling atas dari hirarki merupakan tujuan yang terdiri atas satu elemen. Level berikutnya mungkin mengandung beberapa elemen, di mana elemen-elemen tersebut bisa dibandingkan, memiliki kepentingan yang hampir sama dan tidak memiliki perbedaan yang terlalu mencolok. Jika perbedaan terlalu besar harus dibuatkan level yang baru.

2. Perbandingan penilaian/pertimbangan (*comparative judgments*).

Dengan prinsip ini akan dibangun perbandingan berpasangan dari semua elemen yang ada dengan tujuan menghasilkan skala kepentingan relatif dari elemen. Penilaian menghasilkan skala penilaian yang berupa angka. Perbandingan berpasangan dalam bentuk matriks jika dikombinasikan akan menghasilkan prioritas.

3. Sintesa Prioritas

Sintesa prioritas dilakukan dengan mengalikan prioritas lokal dengan prioritas dari kriteria bersangkutan di level atasnya dan menambahkannya ke tiap elemen dalam level yang dipengaruhi kriteria. Hasilnya berupa gabungan atau dikenal dengan prioritas global yang kemudian digunakan untuk memboboti prioritas lokal dari elemen di level terendah sesuai dengan kriterianya.

AHP didasarkan atas tiga aksioma utama yaitu :

1. Aksioma Resiprokal

Aksioma ini menyatakan jika PC (EA,EB) adalah sebuah perbandingan berpasangan antara elemen A dan elemen B, dengan memperhitungkan C sebagai elemen parent, menunjukkan berapa kali lebih banyak properti yang dimiliki

elemen A terhadap B, maka PC (EB,EA)= 1/ PC (EA,EB). Misalnya jika A 5 kali lebih besar daripada B, maka B=1/5 A.

2. Aksioma Homogenitas

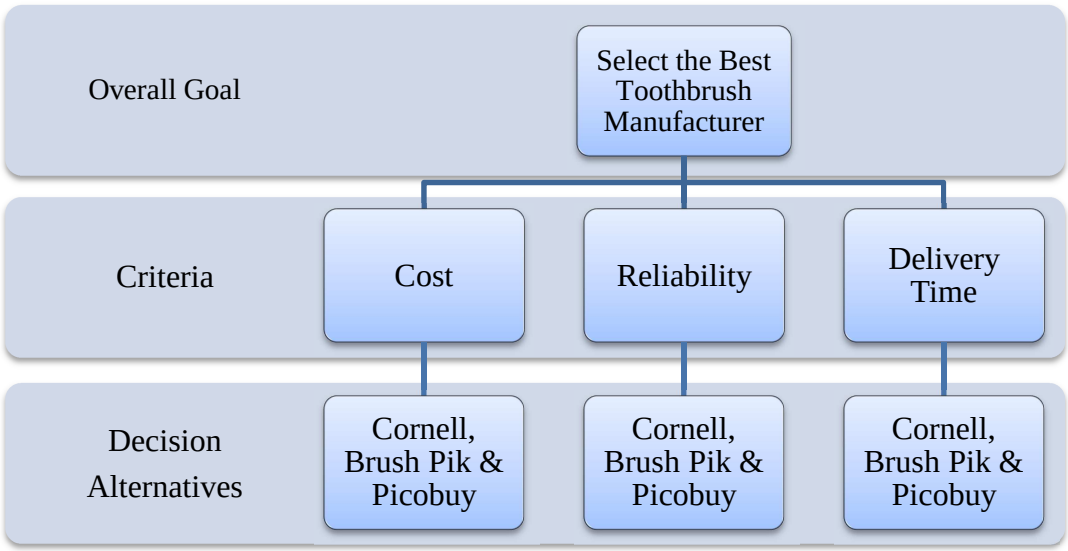
Aksioma ini menyatakan bahwa elemen yang dibandingkan tidak berbeda terlalu jauh. Jika perbedaan terlalu besar, hasil yang didapatkan mengandung nilai kesalahan yang tinggi. Ketika hirarki dibangun, kita harus berusaha mengatur elemen-elemen agar elemen tersebut tidak menghasilkan hasil dengan akurasi rendah dan inkonsistensi tinggi.

3. Aksioma Ketergantungan

Aksioma ini menyatakan bahwa prioritas elemen dalam hirarki tidak bergantung pada elemen level di bawahnya. Aksioma ini membuat kita bisa menerapkan prinsip komposisi hirarki.

2.2.4.4. Menghitung *Analytic Hierarchy Proses* (AHP)

Menghitung *Analytic Hierarchy Proses* (AHP) dalam memilih produsen sikat gigi terbaik dengan kriteria dan alternatif/elemen yang telah ditentukan [4]. Struktur hirarki dalam memilih produsen sikat gigi terbaik dapat dilihat pada gambar 2.13.



Gambar 2.13. Struktur Hirarki

Karena banyaknya alternatif/elemen tiga buah maka n = 3, maka diketahui jumlah matrik perbandingan :

$$n \times \left[\frac{(n-1)}{2}\right] = 3 \times \left[\frac{(3-1)}{2}\right] = 3 \times \left[\frac{2}{2}\right] = 3 \times 1 = 3 \text{ Buah}$$

Jadi, jumlah matrik perbandingan (n x n) = (3 x 3).

1. Menghitung vektor prioritas untuk kriteria *cost* :

a. Cornell 1/3 dari Brush Pik dan 6 kali dari Picobuy.

b. Brush Pik 3 kali dari Cornell dan 7 kali dari Picobuy.

c. Picobuy 1/6 dari Cornell dan 1/7 dari Brush Pik.

Matrik perbandingan untuk kriteria *cost* dapat dilihat pada tabel 2.6.

Tabel 2.6. Matrik perbandingan untuk kriteria *cost*

	Cornell	Brush Pik	Picobuy
Cornell	1.0000	0.3333	6.0000
Brush Pik	3.0000	1.0000	7.0000
Picobuy	0.1667	0.1429	1.0000
Jumlah	4.1667	1.4762	14.0000

Menghitung normalisasi matrik perbandingan untuk kriteria *cost*, contoh untuk menghitung kolom cornell = $\frac{1.0000}{4.1667}$ = 0.2400, $\frac{3.0000}{4.1667}$ = 0.7200, $\frac{0.1667}{4.1667}$ = 0.0400. Normalisai matrik untuk kriteria *cost* dapat dilihat pada tabel 2.7.

Tabel 2.7. Normalisai matrik untuk kriteria *cost*

	Cornell	Brush Pik	Picobuy	Jumlah
Cornell	0.2400	0.2258	0.4286	0.8944
Brush Pik	0.7200	0.6774	0.5000	1.8974
Picobuy	0.0400	0.0968	0.0714	0.2082

Menghitung vektor prioritas untuk kriteria *cost*, contoh untuk menghitung baris cornell = $\frac{0.8994}{3}$ = 0.2981, dimana 3 adalah banyaknya alternatif/elemen. vektor prioritas untuk kriteria *cost* dapat dilihat pada tabel 2.8.

Tabel 2.8. Vektor prioritas untuk kriteria *cost*

	Vektor prioritas
Cornell	0.2981
Brush Pik	0.6325
Picobuy	0.0694

Memeriksa konsistensi untuk kriteria *cost* :

a. Kalikan setiap kolom dari matrik perbandingan dengan vektor prioritas,

$$0.2981\begin{bmatrix}1.0000\\3.0000\\0.1667\end{bmatrix} + 0.6325\begin{bmatrix}0.3333\\1.0000\\0.1429\end{bmatrix} + 0.0694\begin{bmatrix}6.0000\\7.0000\\1.0000\end{bmatrix} = \begin{bmatrix}0.9254\\2.0127\\0.2094\end{bmatrix}$$

b. Bagi setiap kolom dengan vektor prioritas,

0.9254/0.2981 =	3.1039
2.0127/0.6325 =	3.1822
0.2094/0.0694 =	3.0179
Jumlah	9.3040

Menghitung rata-rata maximal untuk kriteria *cost* :

$$\lambda_{max} = \left[\frac{9.3040}{3}\right] = 3.1013$$

Menghitung *Consistency Index* untuk kriteria *cost* :

$$\text{CI} = \left[\frac{\lambda_{max} - \text{n}}{\text{n}-1}\right] = \left[\frac{3.1013-3}{3-1}\right] = 0.0507$$

Menghitung *Consistency Ratio* untuk kriteria *cost*, dimana RI = 0.58 :

$$\text{CR} = \left[\frac{\text{CI}}{\text{RI}}\right] = \left[\frac{0.0507}{0.58}\right] = 0.0873$$

Consistency Ratio untuk kriteria *cost* kurang dari 0.1 atau dibawah 10%, sehingga nilai vektor prioritas untuk kriteria *cost* konsisten.

2. Menghitung vektor prioritas untuk kriteria *reliability* :

a. Cornell 7 kali dari Brush Pik dan 2 kali dari Picobuy.
b. Brush Pik 1/7 dari Cornell dan 5 kali dari Picobuy.
c. Picobuy 1/2 dari Cornell dan 1/5 dari Brush Pik.

Matrik perbandingan untuk kriteria *reliability* dapat dilihat pada tabel 2.9.

Tabel 2.9. Matrik perbandingan untuk kriteria *reliability*

	Cornell	Brush Pik	Picobuy
Cornell	1.0000	7.0000	2.0000
Brush Pik	0.1429	1.0000	5.0000
Picobuy	0.5000	0.2000	1.0000
Jumlah	1.6429	8.2000	8.0000

Menghitung normalisasi matrik perbandingan untuk kriteria *reliability*, contoh untuk menghitung kolom cornell = $\frac{1.0000}{1.6429}$ = 0.6087, $\frac{0.1429}{1.6429}$ = 0.0870, $\frac{0.5000}{1.6429}$ = 0.3043. Normalisai matrik untuk kriteria *reliability* dapat dilihat pada tabel 2.10.

Tabel 2.10. Normalisai matrik untuk kriteria *reliability*

	Cornell	Brush Pik	Picobuy	Jumlah
Cornell	0.6087	0.8537	0.2500	1.7124
Brush Pik	0.0870	0.1220	0.6250	0.8339
Picobuy	0.3043	0.0244	0.1250	0.4537

Menghitung vektor prioritas untuk kriteria *reliability*, contoh untuk menghitung baris cornell = $\frac{1.7124}{3}$ = 0.5708, dimana 3 adalah banyaknya alternatif/elemen. vektor prioritas untuk kriteria *reliability* dapat dilihat pada tabel 2.11.

Tabel 2.11. Vektor prioritas untuk kriteria *reliability*

	Vektor prioritas
Cornell	0.5708
Brush Pik	0.2780
Picobuy	0.1512

3. Menghitung vektor prioritas untuk kriteria *delivery time* :

a. Cornell 8 kali dari Brush Pik dan sebanding dengan Picobuy.
b. Brush Pik 1/8 dari Cornell dan 1/8 dari Picobuy.
c. Picobuy sebanding dengan Cornell dan 8 kali dari Brush Pik.

Matrik perbandingan untuk kriteria *delivery time* dapat dilihat pada tabel 2.12.

Tabel 2.12. Matrik perbandingan untuk kriteria *delivery time*

	Cornell	Brush Pik	Picobuy
Cornell	1.0000	8.0000	1.0000
Brush Pik	0.1250	1.0000	0.1250
Picobuy	1.0000	8.0000	1.0000
Jumlah	2.1250	17.0000	2.1250

Menghitung normalisasi matrik perbandingan untuk kriteria *delivery time* contoh untuk menghitung kolom cornell = $\frac{1.0000}{2.1250}$ = 0.4706, $\frac{0.1250}{2.1250}$ = 0.0588, $\frac{1.0000}{2.1250}$ = 0.4706. Normalisai matrik untuk kriteria *delivery time* dapat dilihat pada tabel 2.13.

Tabel 2.13. Normalisai matrik untuk kriteria *delivery time*

	Cornell	Brush Pik	Picobuy	Jumlah
Cornell	0.4706	0.4706	0.4706	1.4118
Brush Pik	0.0588	0.0588	0.0588	0.1765
Picobuy	0.4706	0.4706	0.4706	1.4118

Menghitung vektor prioritas untuk kriteria *delivery time*, contoh untuk menghitung baris cornell = $\frac{1.4118}{3}$ = 0.4706, dimana 3 adalah banyaknya alternatif/elemen. vektor prioritas untuk kriteria *delivery time* dapat dilihat pada tabel 2.14.

Tabel 2.14. Vektor prioritas untuk kriteria *delivery time*

	Vektor prioritas
Cornell	0.4706
Brush Pik	0.0588
Picobuy	0.4706

4. Menghitung vektor prioritas untuk kriteria :

a. *Cost* 7 kali dari *Reliability* dan 9 kali dari *Delivery Time.*
b. *Reliability* 1/7 dari *Cost* dan 7 kali dari *Delivery Time*.
c. *Delivery Time* 1/9 dari *Cost* dan 1/7 dari *Reliability*.

Matrik perbandingan untuk kriteria dapat dilihat pada tabel 2.15.

Tabel 2.15. Matrik perbandingan untuk kriteria

	Cost	*Reliability*	*Delivery Time*
Cost	1.0000	7.0000	9.0000
Reliability	0.1429	1.0000	7.0000
Delivery Time	0.1111	0.1429	1.0000
Jumlah	1.2540	8.1429	17.0000

Menghitung normalisasi matrik perbandingan untuk kriteria contoh untuk menghitung kolom cornell = $\frac{1.0000}{1.2540}$ = 0.7975, $\frac{0.1429}{1.2540}$ = 0.1139, $\frac{0.1111}{1.2540}$ = 0.0886. Normalisai matrik untuk kriteria dapat dilihat pada tabel 2.16.

Tabel 2.16. Normalisai matrik untuk kriteria

	Cost	*Reliability*	*Delivery Time*	Jumlah
Cost	0.7975	0.8596	0.5294	2.1865
Reliability	0.1139	0.1228	0.4118	0.6485
Delivery Time	0.0886	0.0175	0.0588	0.1650

Menghitung vektor prioritas untuk kriteria, contoh untuk menghitung baris cornell = $\frac{2.1865}{3}$ = 0.7288, dimana 3 adalah banyaknya kriteria. vektor prioritas untuk kriteria dapat dilihat pada tabel 2.17.

Tabel 2.17. Vektor prioritas untuk kriteria

	Vektor prioritas
Cost	0.7288
Reliability	0.2162
Delivery Time	0.0550

5. Menghitung vektor prioritas keseluruhan :

Vektor prioritas keseluruhan ditentukan dengan mengalikan vektor prioritas kriteria dengan vektor prioritas untuk setiap alternatif. Vektor prioritas keseluruhan dapat dilihat pada tabel 2.18.

Tabel 2.18. Vektor prioritas keseluruhan

Kriteria	*Cost*	*Reliability*	*Delivery Time*
	0.7288	0.2162	0.0550

Cornell	0.2981	0.5708	0.4706
Brush Pik	0.6325	0.2780	0.0588
Picobuy	0.0694	0.1512	0.4706

	Cost	*Reliability*	*Delivery Time*	Vektor prioritas keseluruhan
Cornell	(0.7288)(0.2981) = 0.2173	(0.2162)(0.5708) = 0.1234	(0.0550)(0.4706) = 0.0259	0.3665
Brush Pik	(0.7288)(0.6325) = 0.4610	(0.2162)(0.2780) = 0.0601	(0.0550)(0.0588) = 0.0032	0.5243
Picobuy	(0.7288)(0.0694) = 0.0506	(0.2162)(0.1512) = 0.0327	(0.0550)(0.4706) = 0.0259	0.1091

Dari hasil analisa menggunakan metode *Analytic Hierarchy Proses* (AHP), dapat disimpulkan bahwa Brush Pik mempunyai nilai vektor prioritas terbaik dibandingkan Cornell dan Picobuy. Sehingga Brush Pik pantas untuk direkomendasikan menjadi produsen sikat gigi terbaik.

2.2.5. *Unified Modeling Language* (UML)

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*.

Obyek dalam '*software analysis & design*' adalah sesuatu berupa konsep (*concept*), benda (*thing*), dan sesuatu yang membedakannya dengan lingkungannya. Secara sederhana obyek adalah mobil, manusia, alarm dan lain-lainnya. Tapi obyek dapat pula merupakan sesuatu yang abstrak yang hidup didalam sistem seperti tabel, *database*, *event*, *system messages*.

Obyek dikenali dari keadaannya dan juga operasinya. Sebagai contoh sebuah mobil dikenali dari warnanya, bentuknya, sedangkan manusia dari suaranya. Ciri-ciri ini yang akan membedakan obyek tersebut dari obyek lainnya.

Alasan mengapa saat ini pendekatandalam pengembangan *software* dengan *object-oriented*, pertama adalah *scalability* dimana obyek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek. Kedua *dynamic* modeling, adalah dapat dipakai untuk permodelan sistem dinamis dan *real time* [5].

1. Teknik Dasar OOA/D (*Object-Oriented Analysis/Design*)

Dalam dunia pemodelan, metodologi implementasi obyek walaupun terikat kaidah standar, namun teknik pemilihan obyek tidak terlepas pada subyektifitas *software analyst & designer*. Beberapa obyek akan diabaikan dan beberapa obyek menjadi perhatian untuk diimplementasikan di dalam sistem. Hal

ini sah saja karena kenyataan bahwa suatu permasalahan sudah tentu memiliki lebih dari satu solusi. Ada tiga teknik/konsep dasar dalam OOA/D, yaitu pemodulan (*encapsulation*), penurunan (*inheritance*) dan *polymorphism*.

a. Pemodulan (*Encapsulation*)

Pada dunia nyata, seorang ibu rumah tangga menanak nasi dengan menggunakan *rice cooker*, ibu tersebut menggunakannya hanya dengan menekan tombol. Tanpa harus tahu bagaimana proses itu sebenarnya terjadi. Disini terdapat penyembunyian informasi milik *rice cooker*, sehingga tidak perlu diketahui seorang ibu. Dengan demikian menanak nasi oleh si ibu menjadi sesuatu yang menjadi dasar bagi konsep *information hiding*.

b. Penurunan (*Inheritance*)

Obyek memiliki banyak persamaan, namun ada sedikit perbedan. Contoh dengan beberapa buah mobil yang mempunyai kegunaan yang berbeda. Ada mobil bak terbuka seperti truk, baktertutup seperti sedan dan minibus. Walaupun demikian obyek ini memiliki kesamaan yaitu teridentifikasi sebagai obyek mobil, obyek ini dapat dikatakan sebagai obyek induk (*parent*). Sedangkan minibus dikatakan sebagai obyek anak (*child*), hal ini juga berarti semua operasi yang berlaku pada mobil berlaku juga pada minibus.

c. Polymorphism

Pada obyek mobil, walaupun minibus dan truk merupakan jenis obyek mobil yang sama, namun memiliki juga perbedaan. Misalnya suara truk lebih keras dari pada minibus, hal ini juga berlaku pada obyek anak (*child*) melakukan metoda yang sama dengan algoritma berbeda dari obyek induknya. Hal ini yang disebut *polymorphism*, teknik atau konsep dasar lainnya adalah ruang lingkup / pembatasan. Artinya setiap obyek mempunyai ruang lingkup kelas, atribut, dan metoda yang dibatasi.

UML sendiri terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mendefinisikan diagram-diagram sebagai berikut [5]:

1. *Use Case Diagram*

Use case diagram menggambarkan *fungsionalitas* yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah *entitas* manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi *fungsionalitas* dapat dihindari dengan cara menarik keluar *fungsionalitas* yang *common*.

Sebuah *use case* juga dapat meng-*extenduse case* lain dengan *behaviour*-nya sendiri. Sementara hubungan *generalisasi* antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan *desain* berorientasi objek. *Class* menggambarkan keadaan (*atribut*/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

a. Nama (dan *stereotype*)
b. *Atribut*
c. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
c. *Public*, dapat dipanggil oleh siapa saja

3. *Statechart Diagram*

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimulus yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

Dalam UML, *state* digambarkan berbentuk segi empat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

4. *Activity Diagram*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses dan jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih.

Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa object *swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

5. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistententity*.

6. *Collaboration Diagram*

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence*

number, di mana *message* dari level tertinggi memiliki nomor satu. *Messages* dari level yang sama memiliki *prefiks* yang sama.

7. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* atau *package*, tapi dapat juga dari komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

8. *Deployment Diagram*

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal.

Sebuah *node* adalah *server*, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

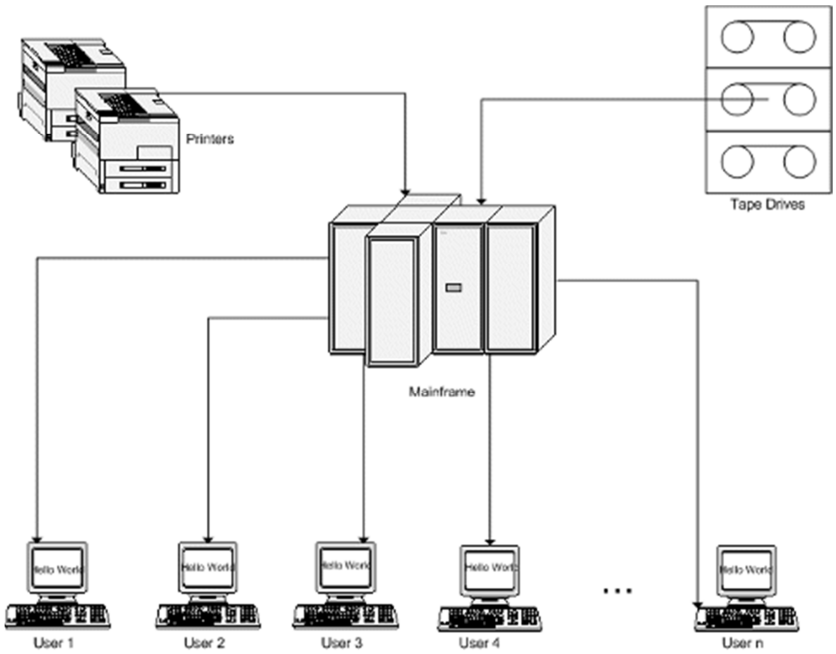
2.2.6. Arsitektur *Client Server*

Istilah arsitektur mengacu pada desain sebuah aplikasi, atau dimana komponen yang membentuk suatu sistem ditempatkan dan bagaimana mereka berkomunikasi. Arsitektur terdistribusi sebuah istilah yang relatif baru untuk menjelaskan arsitektur aplikasi berarti bahwa pemrosesan dari suatu aplikasi terjadi pada lebih dari satu mesin.

Terdapat beberapa macam arsitektur aplikasi, yaitu :

1. *Standalone* (*one-tier*)

Pada arsitektur ini semua pemrosesan dilakukan pada *mainframe*. Kode aplikasi, data dan semua komponen sistem ditempatkan dan dijalankan pada *host*. Seperti terlihat pada gambar 2.14.

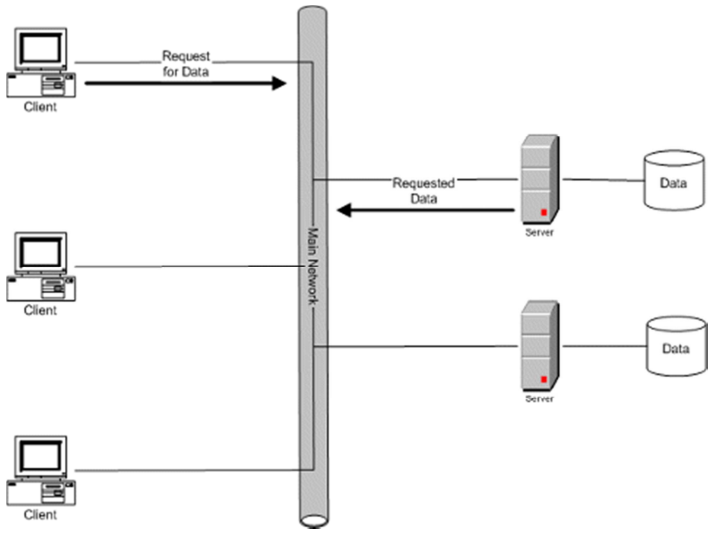


Gambar 2.14. Arsitektur *Standalone (one-tier)*

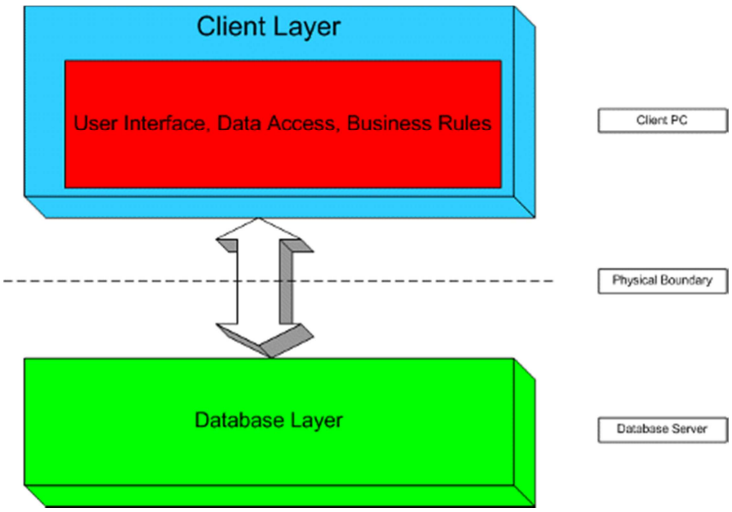
Walaupun komputer *client* dipakai untuk mengakses *mainframe*, tidak ada pemrosesan yang terjadi pada mesin ini, dan karena mereka "*dump-client*" atau "*dump-terminal*". Tipe model ini, dimana semua pemrosesan terjadi secara terpusat, dikenal sebagai berbasis-*host*. Sekilas dapat dilihat kesalahan pada model ini. Ada dua masalah pada komputasi berbasis *host*: Pertama, semua pemrosesan terjadi pada sebuah mesin tunggal, sehingga semakin banyak *user* yang mengakses *host*, semakin kewalahan jadinya. Jika sebuah perusahaan memiliki beberapa kantor pusat, *user* yang dapat mengakses *mainframe* adalah yang berlokasi pada tempat itu, membiarkan kantor lain tanpa akses ke aplikasi yang ada.

2. *Client/Server* (*two-tier*)

Dalam model *client/server*, pemrosesan pada sebuah aplikasi terjadi pada *client* dan *server*. *Client/server* adalah tipikal sebuah aplikasi *two-tier* dengan banyak *client* dan sebuah *server* yang dihubungkan melalui sebuah jaringan, seperti terlihat dalam gambar 2.15. dan gambar 2.16. Aplikasi ditempatkan pada komputer *client* dan mesin *database* dijalankan pada *server* jarak-jauh. Aplikasi *client* mengeluarkan permintaan ke *database* yang mengirimkan kembali data ke *client*-nya.



Gambar 2.15. *Client Server Physical Model*



Gambar 2.16. Arsitektur *Client/Server (two-tier) - Logical View*

Dalam *client/server*, *client* yang cerdas bertanggung jawab untuk bagian dari aplikasi yang berinteraksi dengan *user*, termasuk logika bisnis dan komunikasi dengan *server database*. Tipe-tipe tugas yang terjadi pada *client* adalah :

a. Antarmuka pengguna
b. Interaksi *database*
c. Pengambilan dan *modifikasi* data
d. Sejumlah aturan bisnis
e. Penanganan kesalahan

Server database berisi mesin *database*, termasuk tabel, prosedur tersimpan, dan *trigger* (yang juga berisi aturan bisnis). Dalam sistem *client/server*, sebagian besar logika bisnis biasanya diterapkan dalam *database*. *Server database* manangani :

a. Manajemen data
b. Keamanan
c. *Query, trigger*, prosedur tersimpan
d. Penangan kesalahan

Arsitektur *client/server* merupakan sebuah langkah maju karena mengurangi beban pemrosesan dari komputer sentral ke komputer *client*. Ini berarti semakin banyak user bertambah pada aplikasi *client/server*, kinerja *server file* tidak akan menurun dengan cepat. Dengan *client/server user* dari berbagai lokasi dapat mengakses data yang sama dengan sedikit beban pada sebuah mesin tunggal. Namun masih terdapat kelemahan pada model ini. Selain menjalankan tugas tertentu, kinerja dan skalabilitas merupakan tujuan nyata dari sebagian besar aplikasi. Model *client/server* memiliki sejumlah keterbatasan :

a. Kurangnya skalabilitas
b. Koneksi *database* dijaga
c. Tidak ada keterbaharuan kode
d. Tidak ada tingkat menengah untuk menangani keamanan dan transaksi

Aplikasi-aplikasi berbasis *client/server* memiliki kekurangan pada skalabilitas. Skalabilitas adalah seberapa besar aplikasi bisa menangani suatu

kebutuhan yang meningkat misalnya, 50 *user* tambahan yang mengakses aplikasi tersebut. Walaupun model *client/server* lebih terukur daripada model berbasis *host*, masih banyak pemrosesan yang terjadi pada *server*. Dalam model *client/server* semakin banyak client yang menggunakan suatu aplikasi, semakin banyak beban pada *server*.

Koneksi *database* harus terjaga untuk seluruh *client*. Koneksi menghabiskan sumber daya *server* yang berharga dan seluruh *client* tambahan diterjemahkan ke dalam satu atau beberapa koneksi. Logika kode tidak bisa didaur ulang karena kode aplikasi ada dalam sebuah pelaksanaan *execuTabel monolitik* pada *client*. Ini juga menjadikan modifikasi pada kode sumber sulit. Penyusunan ulang perubahan itu ke semua komputer *client* juga membuat sakit kepala.

Keamanan dan transaksi juga harus dikodekan sebagai pengganti penanganan oleh COM+/MTS. Bukan berarti model *client/server* bukanlah merupakan model yang layak bagi aplikasi-aplikasi. Banyak aplikasi yang lebih kecil dengan jumlah user terbatas bekerja sempurna dengan model ini. Kemudahan pengembangan aplikasi *client/server* turut menjadikannya sebuah solusi menarik bagi perusahaan. Pengembangan umumnya jauh lebih cepat dengan tipe sistem ini. Siklus pengembangan yang lebih cepat ini tidak hanya menjadikan aplikasi meningkat dan berjalan dengan cepat namun juga lebih hemat biaya.

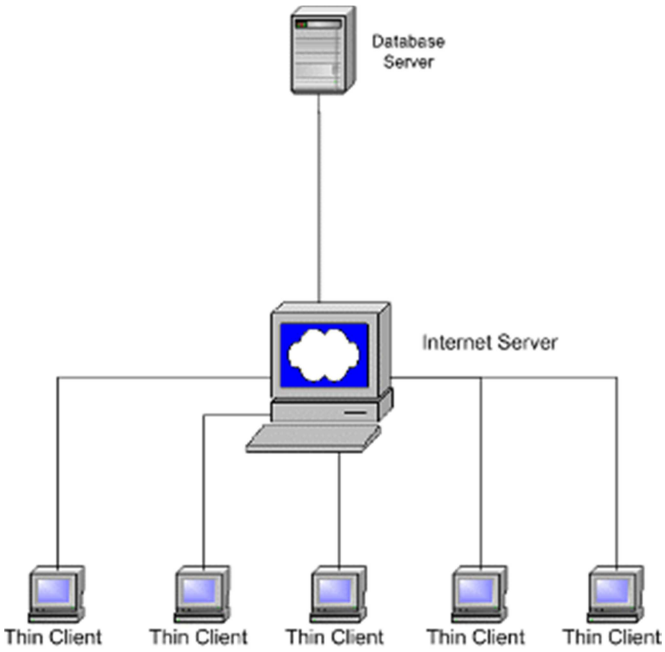
3. *Three-Tier / Multi-Tier*

Model *three-tier* atau *multi-tier* dikembangkan untuk menjawab keterbatasan pada arsitektur *client/server*. Dalam model ini, pemrosesan disebarkan di dalam tiga lapisan (atau lebih jika diterapkan arsitektur *multitier*). Lapisan ketiga dalam arsitektur ini masing-masing menjumlahkan *fungsionalitas* khusus. yaitu :

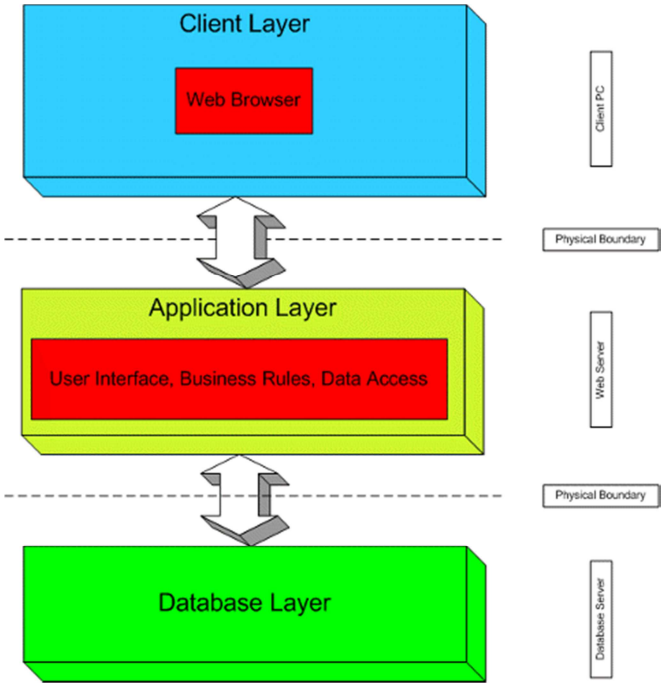
a. Layanan presentasi (tingkat *client*)
b. Layanan bisnis (tingkat menengah)
c. Layanan data (tingkat sumber data)

Layanan presentasi atau logika antarmuka pengguna ditempatkan pada mesin *client*. Logika bisnis dikeluarkan dari kode *client* dan ditempatkan dalam

tingkat menengah. Lapisan layanan data berisi *server database*. Setiap tingkatan dalam model *three-tier* berada pada komputer tersendiri, seperti pada gambar 2.17. dan gambar 2.18.



Gambar 2.17. ***Three-Tier / Multi-Tier***



Gambar 2.18. ***Three-Tier/Multi-Tier - Logical View***

Konsep model *three-tier* adalah model yang membagi *fungsionalitas* ke dalam lapisan *layer*, aplikasi mendapatkan skalabilitas, keterbaharuan, dan keamanan.

2.3. Perangkat Lunak yang Digunakan

Perangkat lunak yang digunakan merupakan beberapa software yang mendukung untuk digunakan dalam pembangunan sistem pendukung keputusan rekrutmen pegawai. Perangkat lunak yang digunakan meliputi C#.NET dan MySQL.

2.3.1. C# .NET

C# merupakan sebuah bahasa pemrograman yang berorientasi objek yangdikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. C# adalah Java versi Microsoft, sebuah bahasa *multi flatform* yang didesain untuk bisaberjalan di berbagai mesin. C# adalah pemrograman berorientasi Object (OOP). C# memiliki kekuatan bahasa C++ dan portabilitas seperti Java. Fitur-fitur yang diambilnya dari bahasa C++ dan Java adalah desain berorientasi objek, seperti *garbage collection*, *reflection*, akar kelas (*root class*), dan juga penyederhanaan terhadap pewarisan jamak (*multiple inheritance*).

Bahasa pemrograman C# dibuat sebagai bahasa pemrograman yang bersifat general-*purpose* (untuk tujuan jamak), berorientasi objek, modern, dan sederhana. C# ditujukan agar cocok digunakan untuk menulis program aplikasi baik dalam system *clien-server* (*hosted system*) maupun sistem *embedded* (*embedded system*), mulai dariprogram aplikasi yang sangat besar yang menggunakan sistem operasi yang canggih hingga kepada program aplikasi yang sangat kecil.

Meskipun aplikasi C# ditujukan agar bersifat 'ekonomis' dalam hal kebutuhan pemrosesan dan memori komputer, bahasa C# tidak ditujukan untuk bersaing secara langsung dengan kinerja dan ukuran program aplikasi yang dibuat dengan menggunakan bahasa pemrograman C [6].

2.3.1.1. Perkembangan C# .NET

C# .NET saat ini telah berkembang hingga versi ke 3, yaitu C# 3.0. Sejak awal kemunculannya, C# .NET telah memberikan sesuatu yang baru di keluarga bahasa pemprograman Microsoft. Dengan struktur bahasanya yang simpel, modern, dan berbasis OOP, C# .NET telah menjaring banyak programmer untuk menggunakannya. Walaupun C# .NET mirip dengan Java, namun dengan didukung oleh IDE yang menarik dan simpel pada Visual Studio serta dukungannya terhadapmetode RAD, C# .NET telah menjadi salah satu bahasa yang fenomenal.

Diantara keluarga bahasa pemprograman Microsoft lainnya, C# .NET merupakan satu-satunya yang benar-benar terintegrasi dengan CLI-nya .NET Framework. Karena C# mempunyai fitur-fitur yang sangat erat kaitannya dengan CLI dan implementasi dari C# seluruhnya ditargetkan pada CLI.

Dengan berkembangnya C# .NET, bahasa ini telah disetujui menjadi ISO Standard dan juga oleh ECMA. Dengan distandarisasinya C# .NET hal ini berarti C#.NET semakin luas jangkauan lingkupnya [6].

2.3.1.2. Fitur-Fitur Utama C# .NET

Fitur-fitur utama C# .NET meliputi kesederhanaannya, *modern*, *object oriented*, *type safe*, *interoperability* dan modular [6].

1. Kesederhanannya

a. *Pointer* telah dihilangkan dari C#.

Hal ini berarti tidak perlu lagi manajemen memori secara manual yang dapat mempersulit programmer dalam membangun suatu aplikasi.

b. Karena berada dalam lingkungan .NET, maka C# memiliki turunan manajemenmemori otomatis dan *Garbage Collection*.

Untuk mengganti *pointer*, sistem manajemen memori otomatis seperti ini digunakan, hal ini berdampak pada manjemen memori yang lebih mudah dan mempermudah pekerjaan programmer.

c. Dukungan tipe data *primitif* yang lebih banyak seperti misalnya *Integer*, *Floats*, dan sebagainya.

Dengan adanya tipe data *primitif* yang lengkap, programmer dapat memilih tipe data yang ingin digunakan sesuai dengan kondisi yang ada secara optimal tanpa dibatasi oleh jumlah tipe variabel yang di*support*.

d. Nilai *integer* 0 dan 1 tidak diterima lagi sebagai alternatif nilai Boolean. Nilai Boolean adalah murni *true* atau *false*.

Dengan adanya peraturan ini berarti C# .NET telah menjalankan konsistensinya dalam penggunaan sintaks pemprograman. Hal ini dapat mengurangi ambiguitas yang sering terjadi ketika membangun suatu aplikasi.

2. *Modern*

a. C# didasarkan pada *trend* yang berkembang saat ini dan sangat ampuh serta mudah digunakan untuk membangun aplikasi yang bersifat *interopable*, skalabilitas, dan cepat.
b. C# menyertakan dukungan yang dapat membuat setiap komponennya diintegrasikan ke dalam *web service* yang dapat dipanggil dari aplikasi apa saja dan *platform* apa saja melalui internet.

3. *Object Oriented*

a. C# mendukung *enkapsulasi*, *inheritance*, *polymorphism*, dan *interfaces* secara penuh.
b. (*int, float, double*) bukanlah merupakan *object* dalam bahasa Java, tetapi C# memperkenalkan struktur yang memungkinkan tipe-tipe *primitif* dapat menjadi *object*.

4. *Type Safe*

a. Pada C#, kita tidak dapat melakukan hal-hal tertentu seperti mengkonversi nilai *variabel* bertipe *double* ke boolean.
b. *Indeks array* selalu dari Nol dan batasannya selalu diperhatikan.
c. Tipe-tipe yang menghasilkan nilai *overflow* dapat dievaluasi.

5. *Interoperability*

a. C# menyertakan dukungan pada COM dan aplikasi berbasis windows.
b. *User* sekarang tidak lagi secara eksplisit mengimplementasikan *inteface* COM yang tidak diketahui, *fitur* seperti itu sudah ada pada C#.

c. C# mengizinkan *user* untuk menggunakan *pointer* sebagai blok kode yang bersifat tidak aman.
d. Komponen yang dibuat oleh VB.NET atau bahasa pemrograman lain yang mendukung .NET dapat langsung digunakan oleh C#.

6. Modular

Kode C# ditulis dengan pembagian masing *classes* yang terdiri dari beberapa *routines* yang disebut sebagai member *methods*. *Classes* dan metode-metode ini dapat digunakan kembali oleh program atau aplikasi lain. Hanya dengan memberikan informasi yang dibutuhkan oleh *Class* dan metode yang dimaksud, maka kita akan dapat membuat suata kode yang dapat digunakan oleh satu atau beberapa aplikasi dan program (*reusable code*).

2.3.1.3. Keunggulan Dan Kekurangan C# .NET

C# diciptakan dengan harapan menjadi bahasa pemrograman yang simpel, modern, dan juga berbasis OOP. C# berusaha mencapai keungulan yang diharapkan dengan menjadikan *developer* inti dari Delphi menjadi *team leader* dari C#. C# menjadi bahasa pemrograman yang unggul karena C# mengadopsi konsep positif dan keunggulan dari bahasa yang lain yang merupakan pendahulu dari C# seperti C, C++, JAVA, dan juga VB. Konsep OOP misalnya, C# berbasiskan bahasa C++ dan sangat kental dengan pengaruh konsep OOP milik Java. Sehingga dengan pengadopsian bahasa pemrograman yang memang bagus dalam hal pembuatan aplikasi berbasis OOP maka C# pun juga menjadi bahasa yang sering dipakai oleh programmer. Selain itu sokongan dari Microsoft dan juga *tool* yang ada, serta tidak lupa IDE yang baik membuat C# dapat bersaing dengan bahasa pemrograman lainya.

Dengan segala keunggulan C# .NET mulai dari manajemen memorinya yang lebih baik, standarisasi oleh ECMA dan ISO, dukungan penuh dari vendor dan lain-lain, C# .NET memiliki kelemahan yang mirip dengan keluarga bahasa pemprograman lainnya yang berada di bawah .NET Framework. C# .NET membutuhkan *platform* yang akan menjalankan programnya sudah ter-*instal*

dengan .NET Framework yang besar filenya semakin lama semakin memberatkan *end-user*.

C# .NET juga mengalami masalah mengenai penggunaan *resource* komputernya yang lebih banyak dari pada bahasa pemprograman lainnya. Hal ini berdampak pada tingginya *requirement* yang harus dipenuhi oleh suatu *platform* agar dapat menajalankan aplikasi C# .NET [6]

2.3.2. MySQL

MySQL ialah sebuah perangkat lunak sistem untuk memanajemen basis data SQL atau sering juga disebut DBMS (*Databases Management System*). MySQL adalah DBMS yang *multithread*, *multiuser* dengan perkiraan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL sebagai perangkat lunak gratis, dibawah lisensi GNU GPL.

Tidak sama hal nya dengan Apache, dimana perangkat lunak dikembangkan oleh komunitas umum. MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya.

MySQL adalah sebuah implementasi dari sistem manajemen basisdata *relasional* (RDBMS) yang di distribusikan secara gratis dibawah lisensi GPL. SQL adalah sebuah konsep pengoperasian basisdata untuk pemilihan atau seleksi dan pemasukan data, yang pengoperasiannya dapat dengan mudah secara otomatis.

Sebagai basis data, MySQL mendukung operasi basis data transaksional maupun *non*-transaksional, pada operasi transaksional MySQL lebih unggul dari kompetitor lainnya. Namun demikian pada modus *non*-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus *non*-transaksional 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 *non*-transaksional. MySQL memiliki beberapa keistimewaan, antara lain [7]:

1. *Portabilitas*, yaitu dapat berjalan melintasi multiplatform dan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka atau open source, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. *Multi-user*, yaitu dapat digunakan oleh beberapa pengguna atau user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. '*Performance tuning*', memiliki kecepatan yang cepat dan stabil dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data, memiliki ragam tipe data yang sangat kaya, seperti *signed*/*unsigned integer*, *float*, *double*, *char*, *text*, *date*, *timestamp*, dan lain-lain.
6. Perintah dan Fungsi, berikut perintah dan fungsi milik MySql pada *query* :
   a. *Create*, yaitu digunakan untuk mmbuat tabel baru.
   b. *Select*, yaitu digunakan untuk menampilkan data sesuai kriteria yang kita tentukan.
   c. *Inser*, yaitu digunakan untuk menyisipkan atau menambah baris pada tabel.
   d. *Update*, yaitu digunakan untuk merubah isi data dalam tabel.
   e. *Delete*, yaitu digunakan untuk menghapus baris atau *record* data dalam tabel.
   f. *Drop*, yaitu digunakan untuk menghapus tabel.
7. Keamanan, MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses user dengan sistem perizinan yang mendetail serta sandi ter-*enkripsi*.
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 klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Lokalisasi, MySQL dapat mendeteksi pesan kesalahan pada klien 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. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABEL, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.