

BAB III

LANDASAN TEORI

3.1 *Company Profile*

Company profile menurut Rachmat Kriyantono (2008), adalah produk tulisan praktisi PR (*Public Relation*) yang berisi gambaran umum perusahaan. Gambaran ini tidak sepenuhnya lengkap, detail dan mendalam. Perusahaan bisa memilih poin-poin apa saja yang ingin disampaikan secara terbuka kepada publiknya. Bahkan ada perusahaan yang memilih membuat *company profile* berdasarkan kepentingan publik sasaran. Ada *company profile* yang dibuat khusus untuk konsumen (pelanggan), untuk bank, untuk pemerintah, pemasok dan sebagainya. Biasanya hal ini dilakukan oleh perusahaan besar yang mempunyai bidang usaha luas dan publik yang berbeda-beda.

3.2 **Konsep Dasar Sistem Informasi**

3.2.1 **Sistem**

Menurut Mulyadi (2008), sistem adalah jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan. Sedangkan pengertian prosedur adalah suatu urutan kegiatan klerikal, biasanya melibatkan beberapa orang dalam satu departemen atau lebih, yang dibuat untuk menjamin penanganan secara transaksi perusahaan yang terjadi berulang-ulang.

3.2.2 **Informasi**

Menurut Kusriani (2008), informasi merupakan data yang sudah diolah sedemikian rupa sehingga sesuai dengan yang dibutuhkan oleh

penggunanya. Untuk memperoleh informasi yang berguna, pertama kali yang harus dilakukan adalah pengumpulan data, lalu diolah sehingga menjadi informasi. Ketika data telah menjadi suatu informasi, maka informasi tersebut akan menjadi terarah dan penting. Hal ini dikarenakan telah dilaluinya berbagai tahapan dalam pengolahannya, yaitu pengumpulan data, data apa saja yang terkumpul, dan penemuan informasi yang diperlukan pengguna.

3.2.3 Sistem Informasi

Menurut Robert A. Leitch dan K. Roscoe Davis (Jogiyanto, 2001:11)

sistem informasi merupakan suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan untuk proses pengambilan keputusan. Jadi dapat ditarik kesimpulan bahwa sistem informasi adalah suatu sistem yang terintegrasi dengan berbagai elemen pendukungnya untuk menyediakan suatu informasi dari data-data yang ada bagi penggunanya.

3.3 Analisis Sistem

Menganalisis sistem merupakan tahapan dalam menganalisis kebutuhan-kebutuhan sistem. Menurut Kendall & Kendall (2003:13), perangkat atau teknik untuk menentukan kebutuhan sistem adalah dengan menggunakan diagram aliran data untuk menyusun daftar input, proses, dan output fungsi bisnis dalam bentuk grafik terstruktur. Dari diagram aliran data, dikembangkan suatu kamus data berisikan daftar seluruh item data yang digunakan dalam sistem beserta spesifikasinya berupa tipe.

Menganalisis kebutuhan sistem dapat pula dilakukan dengan melakukan teknik wawancara guna mendapatkan informasi penting lainnya seperti tujuan mendatang. Jenis informasi berupa perilaku, atau sikap-sikap, keyakinan dan karakteristik beberapa orang utama dalam organisasi yang bisa terpengaruh oleh sistem yang diajukan atau dari yang sudah ada, bisa didapatkan melalui penggunaan kuesioner (Kendall & Kendall, 2003:167). Dengan menggunakan kuesioner, dapat mengukur apa yang ditemukan dalam wawancara dan untuk menentukan seberapa luas atau terbatasnya sentimen yang diekspresikan dalam suatu wawancara.

3.4 Desain Sistem

Menurut Jogiyanto (2005:196) desain sistem dapat didefinisikan sebagai penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi.

Desain sistem menentukan bagaimana suatu sistem akan menyelesaikan tahap ini menyangkut mengkonfigurasi dari komponen perangkat lunak dan perangkat keras suatu sistem sehingga setelah instalasi dari sistem akan memuaskan rancang bangun yang telah ditetapkan pada akhir tahapan analisis (Jogiyanto, 2005:196). Berdasarkan beberapa definisi di atas maka desain sistem dapat diartikan sebagai berikut:



- a. Tahap setelah analisis dari siklus pengembangan sistem.
- b. Pendefinisian dari kebutuhan-kebutuhan fungsional.
- c. Persiapan untuk rancang bangun untuk implementasi.
- d. Menggambarkan bagaimana suatu sistem dibentuk.




- e. Yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi.

3.5 Diagram Alir Dokumen (*Document Flowchart*)

Menurut Jogiyanto (2005), diagram alir dokumen atau *paperwork flowchart* merupakan diagram alir yang menunjukkan arus laporan dan formulir beserta tembusannya. Berdasarkan pengertian di atas dapat disimpulkan bahwa diagram alir dokumen adalah diagram yang menggambarkan aliran seluruh dokumen. Diagram alir dokumen ini menggunakan simbol-simbol yang sama dengan diagram alir sistem. Diagram alir dokumen digambar dengan menggunakan simbol-simbol yang ada pada tabel 3.1 (Jogiyanto, 2005).

Tabel 3.1 Simbol - Simbol Diagram Alir Dokumen

No	Nama Simbol	Simbol	Fungsi
1.	Terminator		Simbol ini digunakan untuk menunjukkan awal dan akhir suatu proses dokumen.
2.	Document		Simbol ini digunakan sebagai <i>input</i> dan <i>output</i> baik secara manual ataupun dengan menggunakan computer.

No	Nama Simbol	Simbol	Fungsi
3.	Manual Input		Simbol ini berfungsi untuk memasukkan data dengan menggunakan <i>online keyboard</i> .
4.	Manual Process		Simbol ini menunjukkan kegiatan manual.
5.	Offline Storage		Simbol ini merupakan dokumen yang diarsip dan diurutkan berdasarkan N (<i>numeric</i>), A (<i>alphabet</i>), C (<i>chronological</i>)
6.	Flow		Simbol ini digunakan sebagai arah aliran dokumen.

3.6 Diagram Alir Sistem (*System Flowchart*)

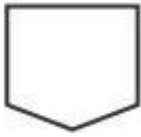
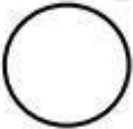
Diagram alir sistem merupakan diagram alir yang menggambarkan suatu sistem peralatan komputer yang digunakan untuk mengolah data dan menghubungkan antar peralatan tersebut (Oetomo, 2002). Diagram alir sistem ini tidak digunakan untuk menggambarkan langkah-langkah dalam memecahkan masalah tetapi hanya menggambarkan prosedur pada sistem yang dibentuk.


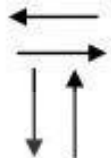
Berikut adalah penjelasan simbol dari diagram alir sistem:

1. *Flow Direction Symbols*

Flow direction symbols digunakan untuk menghubungkan antara satu simbol dengan simbol lainnya (Ladjamudin, 2005). Simbol ini disebut *connecting line*. Simbol-simbol tersebut dijelaskan pada tabel 3.2.

Tabel 3.2 Simbol - Simbol *Flow Direction*

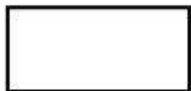

No	Nama Simbol	Simbol	Fungsi
1.	<i>Offline Connector</i>		Fungsi dari simbol ini adalah menyambungkan antara suatu proses dengan proses lainnya di halaman yang berbeda.
2.	<i>Connector</i>		Fungsi dari simbol ini adalah menyambungkan antara, suatu proses dengan proses lainnya di halaman yang sama.

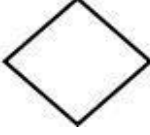
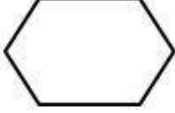


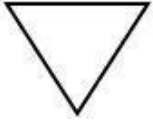
3.	Communication Link		Fungsi dari simbol ini adalah mentransisi suatu data atau informasi dari setiap lokasi.
4.	Flow		Fungsi dari simbol ini adalah menyatakan jalannya arus suatu proses.


2. Processing Symbols

Processing symbols merupakan simbol yang menunjukkan jenis operasi pengolahan data dalam suatu proses (Ladjamudin, 2005). Simbol-simbol tersebut dijelaskan pada tabel 3.3.

Tabel 3.3 Simbol - Simbol *Processing*

No.	Nama Simbol	Simbol	Fungsi
1.	Offline Conector		Simbol ini berfungsi untuk menyambungkan satu proses dengan proses lainnya di halaman yang berbeda.
2.	Manual Process		Simbol ini berfungsi untuk melakukan prosedur atau

			proses tanpa menggunakan komputer.
3.	<i>Decision</i>		Simbol ini berfungsi untuk melakukan pengecekan.
4.	<i>Predefined Process</i>		Simbol ini berfungsi sebagai tempat penyimpanan nilai awal.
5.	<i>Terminal</i>		Simbol ini berfungsi untuk menyatakan permulaan atau penghentian suatu program.
6.	<i>Key Operation</i>		Simbol ini berfungsi untuk menyatakan suatu jenis operasi yang diproses dengan menggunakan mesin yang memiliki <i>keyboard</i> .
7.	<i>Offline Storage</i>		Simbol ini digunakan untuk menyimpan data ke suatu media tertentu.

8.	<i>Manual Input</i>		<p>Simbol ini berfungsi untuk memasukkan data dengan menggunakan <i>online keyboard</i>.</p>
----	----------------------------	---	--


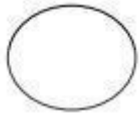

3.7 Diagram Konteks (*Context Diagram*)

Diagram konteks merupakan sebuah model proses yang digunakan untuk mendokumentasikan ruang lingkup dari sebuah sistem (Whitten, 2004). Menurut Oetomo (2002), terdapat beberapa hal yang perlu diperhatikan dalam membuat diagram konteks, diantaranya:

1. Kelompok pemakai, baik internal maupun eksternal perusahaan.
2. Identifikasi kejadian-kejadian yang mungkin terjadi dalam penggunaan sistem.
3. Arah anak panah yang menunjukkan aliran data.
4. Setiap kejadian digambarkan dalam bentuk yang sederhana dan mudah dipahami oleh pembuat sistem.

Suatu diagram konteks hanya mengandung satu proses saja, biasanya diberi nomor proses 0. Proses ini mewakili proses dari seluruh sistem dengan dunia luarnya. Simbol-simbol yang digunakan dalam membuat diagram konteks digambarkan pada tabel 3.4.

Tabel 3.4 Simbol - Simbol *Context Diagram*

No	Nama Simbol	Simbol	Fungsi
1.	<i>Terminator</i>		Simbol ini digunakan untuk berkomunikasi dengan sistem aliran data.
2.	<i>Process</i>		Simbol ini berfungsi untuk mewakili suatu aktifitas yang ada pada sistem.
No	Nama Simbol	Simbol	Fungsi
3.	<i>Flow (Aliran data)</i>		Simbol ini digunakan untuk menunjukkan arah dari aliran

3.8 *Data Flow Diagram (DFD)*

Menurut Whitten (2004), *Data Flow Diagram (DFD)* merupakan alat yang menggambarkan aliran data melalui sistem. Dalam pembuatan DFD, terdapat beberapa tingkatan yang bertujuan untuk menghindari aliran data yang rumit. Tingkatan tersebut dimulai dari tingkatan tertinggi ke bentuk yang lebih rinci. Tingkatan DFD terdiri atas:

1. **Diagram Konteks (*Context Diagram*)**

Diagram konteks merupakan sebuah model proses yang digunakan untuk mendokumentasikan ruang lingkup dari sebuah sistem (Whitten, 2004).


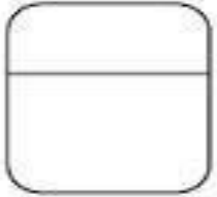
2. Diagram Rinci


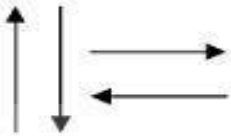
Diagram rinci menggambarkan rincian dari proses yang ada pada tingkatan sebelumnya. Diagram ini merupakan diagram dengan tingkatan paling rendah dan tidak dapat diuraikan lagi.

3. Diagram Level 0

Diagram level 0 merupakan diagram aliran data yang menggambarkan sebuah *event* konteks. Diagram ini menunjukkan interaksi antara *input*, *output*, dan *data store* pada setiap proses yang ada.

Tabel 3.5 Simbol - Simbol *DFD*

Nama Simbol	Simbol	Keterangan
<i>External Entity</i>		<p><i>External entity</i> merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lainnya yang akan memberikan <i>input</i> ataupun menerima <i>output</i>.</p>
<i>Process</i>		<p>Proses adalah kegiatan yang dilakukan oleh orang atau komputer dari arus data yang masuk untuk menghasilkan arus data yang keluar.</p>

Data Store		<i>Data store</i> merupakan tempat penyimpanan data yang berupa <i>file</i> maupun <i>database</i> di dalam sistem komputer
Data Flow		<i>Data flow</i> atau aliran data yang mengalir diantara proses. Aliran data dapat digambarkan dari bawah ke atas, kiri ke kanan, maupun sebaliknya.

3.9 Entity Relationship Diagram (ERD)

Pengertian *Entity Relation Diagram* (ERD) menurut Jogiyanto (2001) adalah suatu komponen himpunan entitas dan relasi yang dilengkapi dengan atribut yang mempresentasikan seluruh fakta. ERD digunakan untuk menggambarkan model hubungan data dalam sistem yang di dalamnya terdapat hubungan entitas berserta atribut relasinya serta mendokumentasikan kebutuhan sistem untuk pemrosesan data. ERD memiliki 4 jenis objek, antara lain:

1. Entity

Menurut Connolly dan Begg (Whitten, 2004), Entitas adalah kelompok orang, tempat, objek, kejadian atau konsep tentang apa yang diperlukan untuk menyimpan data. Setiap entitas yang dibuat memiliki tipe untuk mengidentifikasi apakah entitas tersebut bergantung dengan entitas lainnya atau tidak. Tipe entitas merupakan kumpulan objek yang memiliki kesamaan properti yang teridentifikasi

oleh perusahaan dan memiliki keberadaan yang independen. Tipe entitas terdiri atas dua jenis, yaitu:

a. *Strong Entity*

Strong entity adalah tipe entitas yang tidak bergantung pada keberadaan jenis entitas lainnya. Suatu entitas dikatakan kuat apabila tidak tergantung pada entitas lainnya.

b. *Weak Entity*

Weak Entity adalah tipe entitas yang bergantung pada keberadaan jenis entitas lain yang saling berhubungan. Karakteristik *weak entity* terletak pada entitas *occurrence* yang tidak dapat teridentifikasi secara unik. Entitas *occurrence* adalah sebuah objek yang secara unik dapat teridentifikasi dengan tipe entitas

2. *Attribute*

Menurut Connolly dan Carolyn (2002) atribut adalah deskripsi data yang mengidentifikasi dan membedakan suatu entitas dengan entitas lainnya. Setiap atribut memiliki domain untuk mendefinisikan nilai-nilai potensial yang dapat menguatkan atribut. Atribut domain adalah kumpulan nilai-nilai yang diperbolehkan untuk satu atau lebih atribut. Atribut dapat dibedakan menjadi 5 jenis, yaitu:

a. *Simple Attribute*

Simple Attribute adalah atribut yang terdiri dari komponen tunggal. Sempel atribut tidak dapat dibagi menjadi komponen yang lebih kecil.

b. *Composite Attribute*

Composite Attribute adalah atribut yang terdiri dari beberapa komponen yang bersifat independen.

c. *Single-value Attribute*

Single-value Attribute adalah atribut yang memegang nilai tunggal dari suatu entitas.

d. *Multi-value Attribute*

Muti-value Attribute adalah atribut yang dapat memegang nilai lebih dari suatu entitas.

e. *Derived Attribute*

Derived Attribute adalah atribut yang mewakili turunan nilai sebuah atribut yang saling berkaitan dan belum tentu dalam tipe entitas yang sama.

3. *Keys*

Menurut Connolly dan Carolyn (2002) *keys* terdiri atas beberapa jenis, yaitu:

a. *Candidate Key*

Candidate key merupakan *set* minimal dari suatu atribut yang secara unik mengidentifikasi setiap *occurrence* dari tipe entitas. *Candidate key* tidak boleh *null* (kosong).

b. *Primary Key*

Sebuah *candidate key* yang dipilih untuk mengidentifikasi secara unik tiap kejadian pada suatu entitas. *Primary key* harus bernilai *unique* dan tidak boleh *null* (kosong).

c. Composite Key

Sebuah *candidate key* yang mempunyai dua atribut atau lebih. Suatu atribut yang membentuk *composite key* bukanlah kunci sederhana karena *composite key* tidak membentuk kunci senyawa.

d. Alternate Key

Sebuah *candidate key* yang tidak menjadi *primary key*. *Key* ini biasa disebut dengan *secondary key*.

e. Foreign Key

Himpunan atribut dalam suatu relasi yang cocok dengan *candidate key* dari beberapa relasi lainnya. *Foreign key* mengacu pada *primary key* suatu tabel. Nilai *foreign key* harus sesuai dengan nilai *primary key* yang diacunya.

4. Relationship

Menurut Whitten (2004) *relationship* adalah asosiasi bisnis alami antara satu entitas atau lebih. Dalam suatu relasi, entitas yang saling berelasi memiliki kata kerja aktif yang menunjukkan bahwa keduanya saling berelasi satu sama lain.

Relasi terdiri atas enam tipe, yaitu:

a. Relasi one to many

Relasi *one to many* berarti suatu entitas himpunan A dapat berhubungan dengan banyak entitas pada entitas himpunan B, namun tidak sebaliknya.

b. Relasi one to one

Relasi *one to one* berarti setiap entitas himpunan A hanya berhubungan dengan satu entitas himpunan B, begitu juga sebaliknya.

c. Relasi rekursif one to one

Relasi rekursif *one to one* adalah sebuah tipe relasi yang dimana entitasnya berpartisipasi lebih dari satu peran.

d. Relasi *superclass/subclass*

Entitas *superclass* diidentifikasi sebagai entitas induk dan entitas *subclass* sebagai anggotanya.

e. Relasi *many to many*

Relasi *many to many* berarti setiap entitas himpunan A dapat berhubungan dengan entitas pada himpunan B, begitu juga sebaliknya.

f. Relasi kompleks

Relasi kompleks adalah tipe relasi yang dimana satu entitas berhubungan dengan entitas lainnya yang dapat membentuk sirkulasi dalam relasi tersebut.

3.10 *Internet*

Menurut Jack Febrian (2007:15) *Internet* merupakan tempat terhubungnya berbagai mesin komputer yang mengolah informasi di dunia ini, baik berupa server, komputer pribadi, *handphone*, komputer genggam, PDA, dan lain sebagainya. Masing-masing ini bekerja sesuai dengan fungsinya, baik sebagai penyedia layanan yang biasa disebut dengan *server* maupun sebagai pengguna layanan yang biasa disebut dengan *client*. Berbagai jenis komputer yang jumlahnya mencapai jutaan, terhubung melalui jaringan yang disebut dengan *internet* ini.

Internet terhubung baik melalui kabel, saluran telepon, saluran *handphone*, satelit, *fiber optik*, gelombang listrik, cahaya, serta media apa saja yang mungkin dialiri oleh data. *Internet* merupakan salah satu solusi luar biasa yang pernah diciptakan oleh manusia, informasi apapun dan dari manapun memungkinkan untuk didapatkan melalui teknologi ini.

3.11 Website

Menurut Yuhefizar (2009), *website* adalah keseluruhan halaman-halaman *website* yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah *website* biasanya dibangun atas banyak halaman *website* yang saling berhubungan. Selain itu, *website* dapat juga digunakan sebagai alat promosi, tetapi bukan sebagai alat promosi pertama.

Kelebihan *website* dibandingkan dengan media cetak maupun elektronik adalah kelengkapan informasi yang disajikan dengan biaya yang relatif murah. Kekurangannya adalah produk yang ditampilkan serta pasar yang dituju lebih *segmented* (terpusat pada kalangan/kelompok konsumen tertentu). Oleh karena itu, harus memanfaatkan kekurangannya menjadi *Strong Point* dalam pemasaran.

3.12 HTML (*Hypertext Markup Language*)

Menurut Sutarman (2003), HTML (*Hypertext Markup Language*) adalah suatu bahasa yang digunakan untuk menulis halaman web, HTML dirancang untuk digunakan tanpa bergantung pada suatu *platform* tertentu. Dokumen HTML adalah suatu dokumen text biasa, dan disebut sebagai *markup language* karena mengandung tanda-tanda (*tag*) tertentu yang digunakan untuk menentukan tampilan suatu teks dan tingkat kepentingan dari teks tersebut dalam suatu dokumen. Pada dokumen HTML yang termasuk sistem hypertext yaitu tidak diwajibkan untuk membaca dokumen tersebut secara urut dari atas ke bawah atau sebaliknya, tetapi dapat menuju pada topik tertentu secara langsung dengan menggunakan teks penghubung yang akan membawa ke suatu topik atau dokumen lain secara langsung.

HTML adalah pengembangan dari standart pemformatan dokumen teks yaitu SGML (*Standard Generalized Markup Language*).

3.13 CSS (*Cascading Style Sheet*)

Menurut Saputra & Agustin (2011), CSS atau yang memiliki kepanjangan *Cascading Style Sheet* merupakan suatu bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur, dan seragam.

CSS merupakan pemrograman wajib yang harus dikuasai oleh setiap pembuat program (*Web Programmer*), terlebih lagi itu adalah pendesain web (*web designer*). Ada dua sifat CSS, yaitu:

- a. Internal, Jika kode CSS yang akan dibuat tersebut dimasukkan atau disisipkan ke dalam file kode HTML.
- b. Eksternal, pembuatan kode CSS dan HTML terpisah. Artinya, kita membuatkan satu file CSS untuk kemudian file CSS tersebut dapat dipanggil berulang-ulang guna dihubungkan dengan file HTML (melalui Linked).

3.14 PHP (*Hypertext Preprocessor*)

PHP Hypertext Preprocessor (PHP) adalah bahasa pemrograman *scripting* sisi server (*server-side*), bahasa pemrograman yang digunakan oleh server web untuk menghasilkan dokumen *Hypertext Markup Language* (HTML) *on-the-fly*.

Menurut Kadir (2008), PHP dirancang untuk membentuk aplikasi web dinamis. Artinya, ia dapat membentuk suatu penampilan berdasarkan permintaan terkini. Misalnya, bisa menampilkan *database* ke halaman web. Pada prinsip PHP mempunyai fungsi yang sama dengan skrip-skrip seperti *Active Server Page* (ASP),

Cold Fusion , atau perl. Namun, perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *command line*. Artinya, Skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

3.15 MySQL

MySQL adalah *database server* relasional yang gratis di bawah lisensi *General Public License*. Dengan sifatnya yang *open source*, memungkinkan *user* untuk melakukan modifikasi pada *source code* untuk memenuhi kebutuhan spesifik mereka sendiri. MySQL merupakan *database server multi-user* dan *multi-threaded* yang tangguh (*robust*) yang memungkinkan *backend* yang berbeda, sejumlah *program client* dan *library* yang berbeda, *tool administrative*, dan beberapa antar muka pemrograman. MySQL juga tersedia sebagai *library* yang bisa digabungkan ke aplikasi. Dengan memiliki banyak fitur, MySQL bisa bersaing dengan *database* komersial sekalipun (Utdirartatmo, 2002:1).

Menurut Prasetyo (2003:3) mengemukakan beberapa keistimewaan yang dimiliki oleh MySQL, diantaranya adalah:

1. *Portability*: dapat berjalan stabil pada berbagai sistem operasi (Windows, Linux, Mac OS dan lain-lain).
2. *Open Source*: didistribusikan secara gratis, di bawah lisensi GPL sehingga dapat digunakan secara Cuma-Cuma tanpa dipungut biaya.
3. *Multi-user*: dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance Tuning*: memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana.

5. *Column Types*: memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year, set*, serta *enum*.
6. *Commands and Functions*: memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam *query*.
7. *Security*: memiliki beberapa lapisan sekuritas seperti *level subnetmark*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetil serta *password* terenkripsi.
8. *Scalability dan Limit*: mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. *Connectivity*: dapat melakukan koneksi dengan *client* menggunakan protocol TCP/IP, Unix soket (Unix), atau *Named Pipes*.
10. *Localization*: dapat mendeteksi pesan kesalahan (*error code*) pada *client* dengan menggunakan lebih dari dua puluh bahasa.
11. *Interface*: memiliki *interface* terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. *Lients and Tools*: dilengkapi dengan berbagai *tool* atau perangkat yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.

13. Struktur Tabel: memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan *database* lainnya semacam PostgreSQL dan Oracle.

