

BAB 2

LANDASAN TEORI

2.1 TEORI UMUM

2.1.1 Analisis sistem

Menurut Whitten, Bentley & Dittman (2004,p186) , analisis sistem adalah suatu teknik pemecahan masalah yang menguraikan suatu sistem menjadi bagian – bagian komponen yang bertujuan untuk mempelajari seberapa baik komponen – komponen tersebut bekerja dan berinteraksi untuk mencapai tujuannya. Terdapat 3 pendekatan analisis, yaitu :

- a. **Analisis terstruktur, yaitu** sebuah teknik model driven dan berpusat pada proses yang digunakan untuk menganalisis sistem yang ada, dan mendefinisikan kebutuhan – kebutuhan bisnis untuk sistem yang baru atau keduanya.
- b. **Teknik informasi, yaitu** merupakan teknik model-driven yang berpusat pada data, tapi sensitif terhadap proses. Teknik ini digunakan untuk merencanakan, menganalisis dan mendesain sistem informasi. Model – model teknik informasi adalah gambar – gambar yang mengilustrasikan dan mensinkronkan data dan proses – proses sistem.

- c. **Analisis berorientasi objek**, yaitu teknik model- driven yang mengintegrasikan data dan proses ke dalam konstruksi yang disebut objek. Model – model analisis berorientasi objek adalah gambar – gambar yang mengilustrasikan objek – objek sistem dari berbagai macam perspektif, seperti struktur, kelakuan, dan interaksi objek – objek.

2.1.2 Perancangan Sistem

Perancangan sistem adalah spesifikasi atau konstruksi dari sebuah teknik, solusi berbasis komputer untuk identifikasi kebutuhan – kebutuhan bisnis dalam sebuah analisis sistem, dapat dilakukan dengan menambahkan, menghapus, mengganti bagian – bagian yang berhubungan dengan sistem original (Witten, Bentley & Dittman (2004, p39)).

2.1.2.1 Pendekatan perancangan sistem

Terdapat beberapa strategi untuk melakukan desain sistem. Strategi dan teknik tersebut meliputi desain struktur modern, teknik informasi, prototyping, desain berorientasi objek, RAD (*Rapid Application Development*).

a. Desain struktur modern

Adalah teknik perancangan berorientasi proses untuk mengubah program besar ke dalam hierarki modul – modul

yang menghasilkan sebuah program komputer yang lebih mudah untuk diimplementasikan dan dipelihara (diubah). Contohnya top – down program design.

b. Teknik Informasi

Teknik perancangan berfokus pada data dan sensitif terhadap proses. Aplikasi yang telah diidentifikasi di dalam teknik informasi menjadi proyek tempat diterapkannya metode perancangan dan analisis sistem lain untuk mengembangkan produksi. Metode – metode tersebut dapat melibatkan beberapa kombinasi analisis terstruktur modern, desain terstruktur modern, prototyping, dan analisis dan desain berorientasi objek.

c. Prototyping

Sebuah pendekatan perancangan berbasis teknik modern, pendekatan prototyping adalah proses iteratif yang melibatkan hubungan kerja yang dekat antara perancang dengan pengguna. Prototipe dapat dibangun untuk output sederhana, dialog komputer, keseluruhan subsistem atau bahkan sistem keseluruhan. Masing – masing sistem prototipe ditinjau ulang oleh pengguna akhir dan manajemen, yaitu pihak yang membuat rekomendasi tentang persyaratan, metode, dan format. Prototipe tersebut

kemudian dikoreksi, ditingkatkan atau diseleksi untuk mencerminkan persyaratan baru. Teknologi prototipe membuat beberapa perbaikan dalam cara yang relatif langsung. Revisi dan peninjauan ulang berlanjut sampai prototipe diterima.

d. Desain berorientasi objek

Teknik perancangan ini merupakan perluasan dari strategi analisis berorientasi objek. Teknologi dan teknik objek merupakan usaha untuk menghilangkan pemisahan yang menyangkut data dan proses. Teknik perancangan berorientasi objek digunakan untuk memperbaiki definisi kebutuhan persyaratan objek yang telah diidentifikasi lebih awal selama analisis dan untuk mengenali objek dengan desain spesifik.

e. RAD (*Rapid Application Development*)

Sebuah pendekatan desain sistem yang menggunakan penggabungan teknik terstruktur, prototipe, dan JAD (*Joint Application Development*). RAD mengharuskan penggunaan teknik terstruktur dan prototipe secara interaktif untuk menentukan persyaratan dan kebutuhan pengguna dan mendesain sistem akhir, yang kemudian dilengkapi dengan JAD (*Joint Application Development*)

yang merupakan teknik melengkapi analisis sistem dan teknik desain lain dengan cara menekankan *participative developement* diantara pemilik sistem, pengguna, perancang dan pengembang.

2.1.3 Data, Informasi dan Database

Menurut O'Brien dan Marakas (1997,p24), data adalah kumpulan fakta atau penelitian mentah yang umumnya berupa fenomena fisik atau transaksi bisnis. Lebih khusus lagi, data adalah pengukuran objektif dari atribut (karakteristik) tentang entitas (contohnya orang, tempat, benda, kejadian), sedangkan menurut Thompson dan Cats-Baril (2003, p205), data merupakan potongan informasi yang tidak terstruktur yang dikumpulkan berdasarkan beberapa entitas, kejadian, atau observasi.

Informasi adalah data yang telah diproses atau diatur kembali menjadi sebuah bentuk yang lebih berarti bagi seseorang (Whitten et al (2004, p27)). Informasi dibentuk dari kombinasi-kombinasi data yang diharapkan memiliki arti bagi penerimanya dan menurut O'Brien dan Marakas (1997, p24), Informasi adalah data yang telah dikonversi menjadi sebuah konteks yang memiliki arti dan berguna bagi *end user* tertentu.

Menurut Connolly dan Begg (2005, p15), *database* adalah sekumpulan data dan deskripsi data yang berhubungan secara logis

yang digunakan bersama, dirancang untuk memenuhi kebutuhan informasi sebuah organisasi.

Database juga merupakan koleksi data yang digunakan untuk merepresentasikan informasi yang diinginkan ke dalam sebuah sistem informasi (Atzeni et al (2003, p2)).

Selain itu terdapat pengertian bahwa *database* adalah sebuah koleksi penyimpanan data yang terhubung yang sering digunakan dan mengurangi pengulangan data menurut skemanya (Inmon, 2002, p3),

2.1.4 DBMS (Database Management System)

Menurut Connolly dan Begg (2005, p16), DBMS adalah sistem *software* yang memungkinkan *user* untuk mendefinisikan, membuat, memelihara, dan mengontrol akses ke database, sedangkan menurut O'Brien dan Marakas (2006, p151), DBMS adalah *software* utama dalam pendekatan pengaturan *database*, hal ini disebabkan DBMS mengontrol pembuatan, pemeliharaan, dan penggunaan *database* dalam organisasi dan *end user*.

Selain itu terdapat pengertian juga yaitu mengenai DBMS adalah *software* komputer khusus yang disediakan oleh “computer vendors”, yang digunakan untuk membuat, mengakses, mengontrol, dan mengatur *database* , Whitten et al (2004, p554).

DBMS merupakan perangkat lunak yang berinteraksi dengan program aplikasi user dan database. Umumnya, sebuah Pengelolaan sistem basisdata menyediakan fasilitas sebagai berikut :

1. *Data Definition Language* (DDL) memberikan fasilitas kepada pengguna untuk menspesifikasikan tipe data,strukturnya, dan batasan aturan mengenai data yang bisa disimpan dalam basis data tersebut.
2. *Data Manipulation Language* (DML) memberikan fasilitas yang memperbolehkan pengguna untuk menambah, memperbaiki, menghapus data, dan mendapatkan kembali data.
3. Terdapat fasilitas untuk mengontrol akses ke basis data (Connolly, 2005, p16), seperti :
 - a. Suatu sistem keamanan yang mencegah pengguna yang tidak mempunyai hak untuk mengakses data.
 - b. Suatu sistem terintegrasi yang memelihara konsistensi penyimpanan data.
 - c. Suatu sistem kontrol konkurensi yang memperbolehkan berbagi hak akses ke basis data.
 - d. Suatu sistem kontrol pengembalian data yang dapat mengembalikan data ke keadaan

sebelumnya apabila terjadi kegagalan perangkat keras atau perangkat lunak.

- e. Terdapat suatu katalog yang dapat diakses oleh pengguna yang berisikan deskripsi dari data di dalam basis data tersebut.

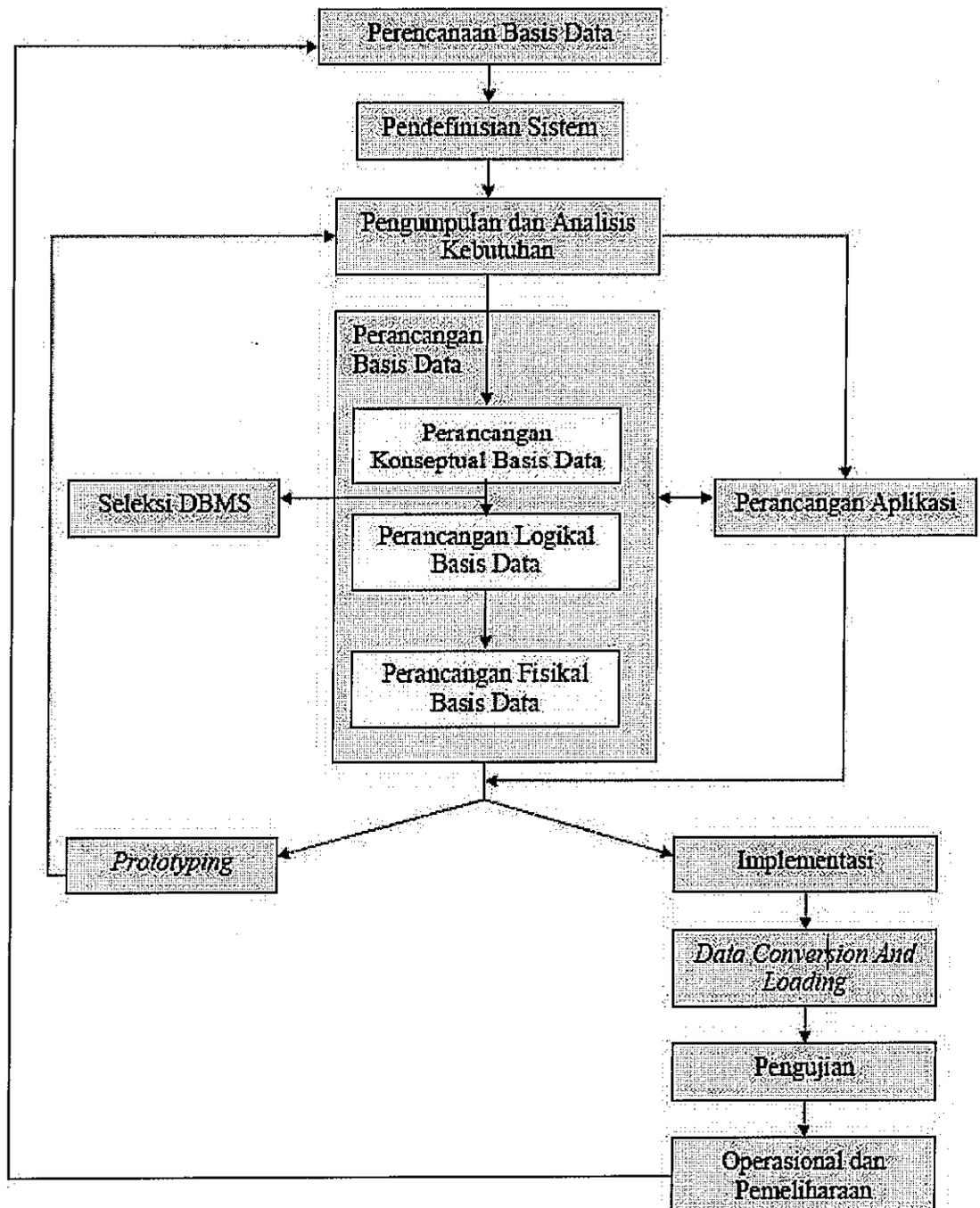
2.1.4.1 Komponen DBMS

1. *Hardware*, yaitu berupa PC hingga jaringan – jaringan komputer.
2. *Software*, yaitu DBMS, sistem operasi, *software* jaringan (bila diperlukan) dan juga aplikasi program.
3. *Data*, digunakan digunakan organisasi dan deskripsi dari data tersebut dinamakan *schema*.
4. *Prosedure* , yaitu instruksi dan aturan yang harus ada pada desain dan kegunaan dari database dan DBMS.
5. *People*, antara lain :
 - a. Data Administration
DA lebih memperhatikan tahap awal dari *lifecycle*.
DA mengatur sumberdaya data, meliputi :
perencanaan database, pengembangan dan pemeliharaan standar, kebijakan, prosedur dan desain database konseptual serta logikal.
 - b. Database Administration

DBA mengatur realisasi fisik dari aplikasi database yang meliputi desain fisik database dan implementasi, pengaturan keamanan dan kontrol integritas, pengawasan performa sistem dan pengaturan ulang database.

- c: Desainer database (logikal dan fisikal)
- d. Programmer aplikasi
- e. *End users*
 - i. *Naive* : user tidak perlu mengetahui database dan DBMS, hanya menggunakan aplikasi.
 - ii. *Sophisticated* : user familiar dengan struktur database dan DBMS.

2.1.5 Siklus hidup aplikasi basis data



Gambar 2.1 Siklus Hidup Basisdata

Menurut Connolly dan Begg basis data merupakan komponen terpenting dari suatu sistem informasi, dimana pengembangan dan penggunaannya harus dilihat dari sudut pandang yang lebih luas sesuai dengan kebutuhan organisasi, sehingga siklus hidup aplikasi basisdata berhubungan dengan siklus hidup sistem informasi. Berikut langkah – langkah siklus hidup aplikasi basis data :

A. Perencanaan basis data

Langkah awal yang merupakan aktivitas manajemen yang memungkinkan tahapan dari siklus hidup basis data direalisasikan se-efisien mungkin. Perencanaan basis data harus tergabung dan terintegrasi dengan keseluruhan strategi sistem informasi dari organisasi. Terdapat 3 hal pokok yang berkaitan dengan strategi sistem informasi, yaitu :

- a. Identifikasi rencana dan sasaran (*goals*) dari perusahaan termasuk mengenai sistem informasi yang dibutuhkan.
- b. Evaluasi sistem informasi yang ada untuk menetapkan kelebihan dan kekurangan yang dimiliki.
- c. Perkiraan kesempatan IT yang mungkin memberikan keuntungan kompetitif.

B. Pendefinisian sistem

Menjelaskan batasan – batasan dan lingkup dari aplikasi basis data dan sudut pandang user (*user view*) yang utama. User

view mendefinisikan apa yang diwajibkan dari suatu aplikasi basisdata dari sudut pandang aturan kerja khusus (seperti *Manager* atau *Supervisor*) atau area aplikasi perusahaan (seperti *marketing*, *personnel*, atau *stock control*). Aplikasi basis data dapat memiliki satu atau lebih *user view*. Identifikasi *user view*, membantu memastikan bahwa tidak ada pengguna utama dari suatu basis data yang terlupakan ketika pembuatan aplikasi baru yang dibutuhkan. *User view* juga membantu dalam pengembangan aplikasi basis data yang kompleks memungkinkan permintaan – permintaan dipecah kedalam bagian – bagian yang lebih mudah.

C. Pengumpulan dan analisis kebutuhan

Merupakan proses dan pengumpulan dan analisa informasi mengenai bagian organisasi yang didukung oleh aplikasi basis data, dan menggunakan informasi tersebut untuk identifikasi kebutuhan pengguna akan sistem yang baru. Informasi yang dikumpulkan untuk setiap *user view* utama meliputi :

- a. Deskripsi data yang digunakan atau yang dihasilkan
- b. Perincian mengenai bagaimana data digunakan dan dihasilkan.
- c. Beberapa kebutuhan tambahan untuk aplikasi basis data yang baru.

D. Perancangan basis data

Suatu proses pembuatan sebuah perancangan basis data yang akan mendukung tujuan dan operasi suatu perusahaan. Tujuan utamanya adalah:

- a. Merepresentasikan data dan hubungan antara data yang dibutuhkan oleh seluruh area aplikasi utama dan *user group*.
- b. Menyediakan model data yang mendukung segala transaksi yang diperlukan pada data.
- c. Menspesifikasikan perancangan minimal secara tepat dan disusun untuk memenuhi kebutuhan performa yang ditetapkan pada sistem.

Data Modeling adalah cara untuk membantu perancang untuk memahami data dan memfasilitasi komunikasi mengenai informasi yang dibutuhkan, Ada dua kegunaan utama dari data modeling, yaitu :

- a. Untuk membantu dalam memahami arti (semantik) dari data.
- b. Untuk memfasilitasi komunikasi mengenai informasi yang dibutuhkan.

Kriteria untuk menghasilkan model data yang optimal :

- a. Validitas Struktural, harus konosisten dengan definisi perusahaan dan informasi organisasi.
- b. Kesederhanaan (*simplicity*), mudah dimengerti baik oleh profesional sistem informasi maupun pengguna non-teknik.
- c. Ketepatan, kemampuan untuk membedakan antara data yang berlainan, hubungan antara data dan batasan – batasannya.
- d. Tidak rangka (*Nonredundancy*), pengeluaran informasi yang tidak berhubungan, dengan kata lain, representasi bagian informasi hanya satu kali.
- e. Digunakan bersama (*Shareability*), tidak ditentukan untuk aplikasi atau teknologi tertentu dan dapat digunakan oleh banyak pengguna.
- f. Perluasan penggunaan (*Extensibility*), kemampuan untuk menyusun dan mendukung kebutuhan baru dengan akibat sampingan yang minimal terhadap user yang sudah ada.

Ada beberapa pendekatan yang dapat digunakan dalam merancang basis data, yaitu sebagai berikut :

- i. Pendekatan *bottom-up*

Pendekatan *bottom-up* dimulai dari analisis atribut-atribut (property dari entity dan relationship), relasi

antara atribut, kemudian dikelompokkan ke dalam suatu relasi yang merepresentasikan tipe dari entiti-entiti, lalu menganalisa relasi antara entiti. Pendekatan ini lebih cocok untuk perancangan basis data yang sederhana dengan jumlah atribut yang relatif kecil.

ii. Pendekatan *top-down*

Pada pendekatan ini terlebih dahulu membangun model data tingkat tinggi, kemudian membangun model data yang lebih sederhana. Pendekatan ini diilustrasikan melalui konsep model Entity Relationship (ER).

iii. Pendekatan *inside-out*

iv. Pendekatan *mixed strategy*

Tiga fase database design :

i. **Conceptual database design**

Suatu proses pembentukan model dari informasi yang digunakan dalam perusahaan, independen dari keseluruhan aspek fisik. Model data dibangun dengan menggunakan informasi dalam spesifikasi kebutuhan user. Model data konseptual merupakan sumber informasi untuk fase desain logikal.

- a. Langkah pertama, buatlah data model lokal yang konseptual untuk setiap *user view*.
 1. Identifikasikan tipe – tipe *entity*.
 2. Identifikasikan tipe – tipe *relationship*.
 3. Identifikasikan dan hubungkan atribut – atribut dengan tipe *entity* atau *relationship*.
 4. Tentukan domain atribut.
 5. Tentukan atribut kandidat dan primary key.
 6. Pertimbangkan konsep permodelan yang tinggi / *enhanced modelling* (step optional).
 7. Periksa model untuk redundansi.
 8. Validasikan model lokal konseptual terhadap transaksi user.
 9. Tinjau kembali data model lokal yang konseptual dengan user.

ii. Logical database design

Suatu proses pembentukan model dari informasi yang digunakan dalam perusahaan berdasarkan model data tertentu (misal : relasional), tetapi independen terhadap DBMS tertentu dan aspek fisik lainnya. Model data konseptual yang telah dibentuk sebelumnya, diperbaiki dan dipetakan ke dalam model data logikal.

b. Langkah kedua buat dan validasikan data model lokal yang logikal untuk setiap view.

1. Hapus (*remove*) fitur – fitur yang tidak kompatibel dengan model relasional.
2. *Derrive relation* untuk model data logikal.
3. Validasikan hubungan menggunakan normalisasi.
4. Validasikan hubungan terhadap transaksi user.
5. Tentukan batasan *integrity*.
6. Tinjau kembali model data logikal lokal dengan user.

c. Langkah ketiga, buat dan validasikan model data logikal global.

1. Gabungkan model data logikal lokal menjadi data logikal global.
2. Validasikan model data logikal global.
3. Periksa untuk pengembangan mendatang.
4. Tinjau kembali model data logikal global dengan user.

iii. **Physical database design**

Proses yang menghasilkan deskripsi implementasi database pada penyimpanan sekunder. Menggambarkan struktur penyimpanan dan metode akses yang digunakan

untuk mencapai akses yang efisien terhadap data. Dapat dikatakan juga desain fisik merupakan cara pembuatan menuju sistem DBMS tertentu.

- d. Langkah keempat, terjemahkan model data logikal global target DBMS.
 1. Desain *base relation*.
 2. Desain representasi dan *derived data*.
 3. Desain batasan – batasan perusahaan
- e. Langkah kelima, desain representasi fisik
 1. Analisa transaksi – transaksi.
 2. Pilih organisasi file.
 3. Pilih indeks – indeks.
 4. Perkirakan kebutuhan tempat penyimpanan (*disc space*).
- f. Langkah keenam, desain sudut pandang pengguna.
- g. Langkah ketujuh, desain mekanisme keamanan.
- h. Langkah kedelapan, pertimbangkan pengenalan dari redundansi terkontrol.
- i. Langkah kesembilan, awasi dan atur sistem operasional.

E. Pemilihan DBMS (*optional*)

Pemilihan DBMS yang tepat dapat mendukung aplikasi database.

Pemilihan DBMS dapat dilakukan kapanpun sebelum menuju desain logikal asalkan terdapat informasi yang memadai mengenai kebutuhan sistem. Tahap – tahap utama dalam memilih DBMS :

1. Mendefinisikan terminologi studi referensi.
2. Mendaftar dua atau tiga produk.
3. Evaluasi produk.
4. Rekomendasi pilihan dan laporan produk.

F. Perancangan Aplikasi

Perancangan antarmuka untuk user dan program aplikasi yang menggunakan dan memproses *database*. Dalam perancangan aplikasi terdapat perancangan *database transaction* yang merupakan perancangan urutan aksi yang dilakukan oleh user untuk dapat mengakses, mengubah atau menghapus isi *database*.

G. Prototyping

Merupakan metode pembuatan prototipe dari aplikasi yang belum sempurna, namun dapat membantu user dan perancang dalam memvisualisasikan dan mengevaluasi aplikasi yang diinginkan. Terdapat dua strategi prototyping yang biasa digunakan, yaitu :

1) Requirements Prototyping

Strategi ini menggunakan hasil prototipe untuk menentukan kebutuhan dari sistem database yang diajukan, setelah

kebutuhan – kebutuhan tersebut terpenuhi maka prototipe tersebut di buang dan tidak digunakan lagi.

2) *Evolutionary prototyping*

Strategi ini juga digunakan untuk tujuan yang sama dengan strategi *Requirements prototyping*, hanya saja setelah kebutuhan – kebutuhan dari sistem database sudah terpenuhi prototipe ini tetepa digunakan untuk dikembangkan menjadi sistem database yang berfungsi untuk *user*.

H. Implementasi

Merupakan realisasi fisik dari perancangan database dan aplikasinya. Pada tahap ini aplikasi menggunakan DDL (Data Definition Language) dari DBMS yang dipilih atau Graphical User Interface (GUI). DDL digunakan untuk membuat struktur database dan file database kosong. Dalam tahap ini kontrol kemanan dan integritas sistem juga diimplementasikan.

I. Konversi dan Proses Memuat Data

Pada tahap ini dilakukan pemindahan semua data yang ada ke dalam database baru dan mengubah aplikasi yang ada untuk menjalankan dan menggunakan database yang baru. Namun pada masa sekarang, umumnya DBMS memiliki kegunaan untuk memasukan *file* ke dalam basis data baru, tujuannya adalah untuk memungkinkan pengembang mengkonversi dan menggunakan

aplikasi program yang lama untuk digunakan oleh sistem yang baru.

J. Pengujian (*Testing*)

Testing merupakan proses percobaan sistem database untuk tujuan menemukan kekurangan dan *error* yang terdapat pada aplikasi dan sistem database. Pengguna – pengguna sistem yang baru seharusnya diikutsertakan dalam proses pengujian. Situasi yang ideal untuk suatu proses pengujian sistem adalah dengan menguji basis data pada sistem *hardware* yang berbeda, dan juga sangat penting sekali untuk menyiapkan back up untuk menangkap *error* yang terjadi.

K. Pemeliharaan Operasional

Pemeliharaan operasional (*Operational Maintenance*) merupakan proses mengawasi dan memelihara sistem setelah instalasi. Tahap ini meliputi fakta- fakta berikut :

- a. Mengawasi performa dari sistem.
- b. Memelihara dan mengembangkan basis data (bila diperlukan).

2.1.6 *Relational Database*

Menurut Connolly dan Begg (2005, p74), relational database merupakan kumpulan relasi ternormalisasi dengan perbedaan nama relasi. Relational database terdiri dari relasi yang tepat terstruktur, yang lebih tertuju pada normalisasi.

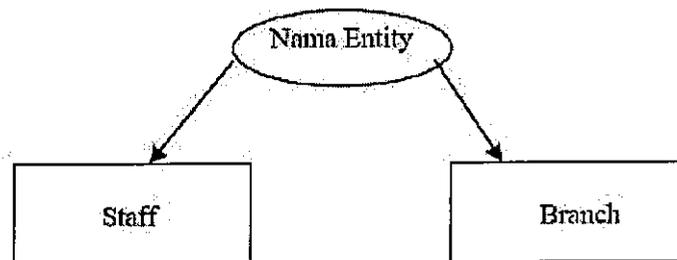
2.1.7 *Entity Relational Model*

Menurut Connolly (2005, p342), *ER Modeling* adalah pendekatan *top-down* pada perancangan basis data, yang dimulai dengan identifikasi data yang penting, disebut juga entitas, dan hubungan antar entitas yang harus direpresentasikan model.

2.1.7.1 *Entity Type*

Tipe entity adalah kumpulan objek – objek dengan *property* yang sama, yang didefinisikan oleh perusahaan yang keberadaannya tidak tergantung. Konsep dasar dari bentuk *entity relationship* adalah tipe *entity*. Sebuah tipe *entity* memiliki keberadaan yang bebas dan bisa menjadi objek dengan keberadaan fisik atau menjadi objek dengan keberadaan konseptual.

Entity Occurrence yaitu pengidentifikasian objek yang unik dari sebuah *entity type*. Setiap entitas diidentifikasi dan disertakan *property*-nya.

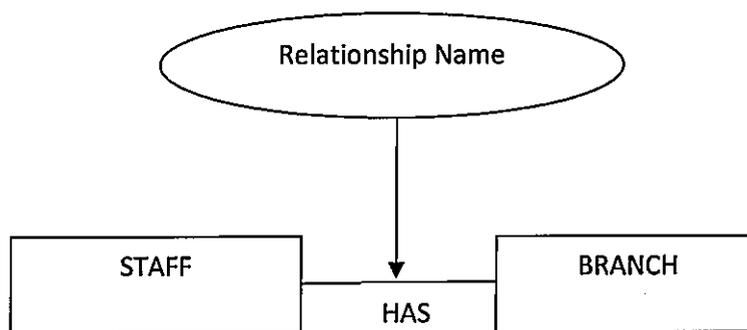


Gambar 2.2 *Entity Occurance*

Menurut Connolly (2005, p354-355), tipe entitas dibedakan menjadi 2 yaitu tipe entitas kuat dan tipe entitas lemah. Tipe entitas kuat yaitu yang keberadaannya tidak bergantung pada entitas lain. Sedangkan tipe entitas lemah yaitu entitas yang keberadaannya bergantung pada entitas lain.

2.1.7.2 Relationship Type

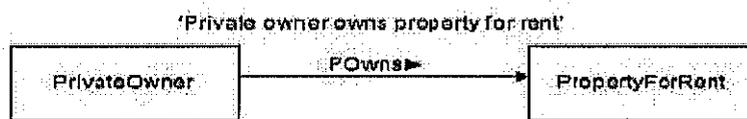
Tipe relationship adalah sebuah gabungan yang mempunyai arti diantara tipe – tipe entity. Relationship Occurrence adalah keterhubungan yang diidentifikasi secara unik yang meliputi keberadaan tiap entity type yang berpartisipasi.



Gambar 2.3 Relationship Type

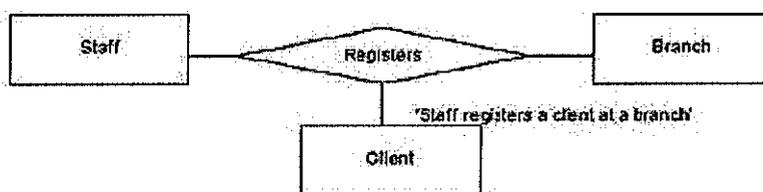
Derajat tipe hubungan (*degree of relationship type*) yaitu jumlah entitas yang berpartisipasi dalam suatu hubungan. Derajat tipe hubungan terdiri dari (Connolly, 2005, p347-349):

1. Binary Relationship



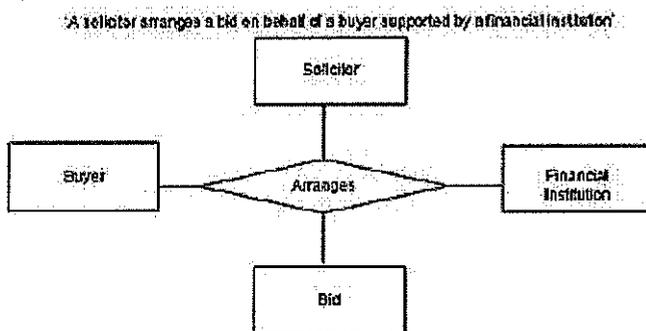
Gambar 2.4 *Binary Relationship*

2. Ternary Relationship



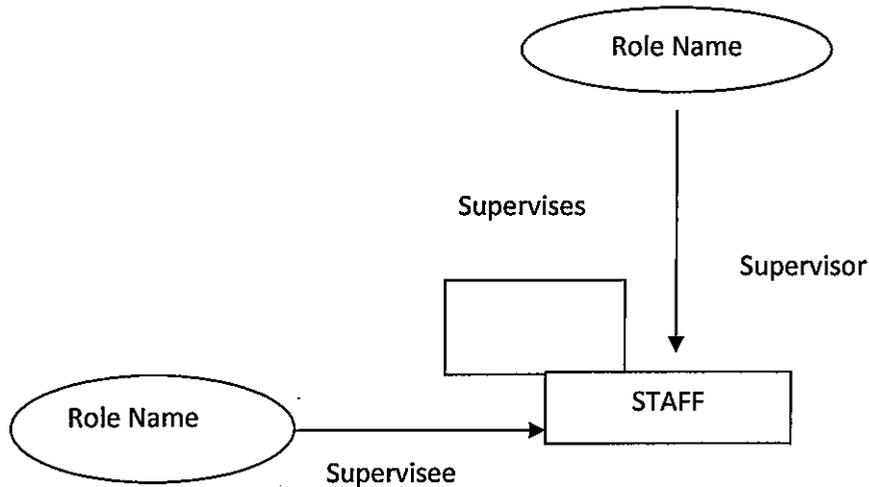
Gambar 2.5 *Ternary Relationship*

3. Quaternary Relationship



Gambar 2.6 *Quaternary Relationship*

4. Unary Relationship



Gambar 2.7 *Unary Relationship*

2.1.7.3 *Attributes*

Atribut adalah sifat dari sebuah entity atau sebuah tipe relationship (Connolly, 2005, p350).

Simple attribute yaitu atribut yang terdiri dari satu komponen tunggal dengan keberadaan yang independen dan tidak dapat dibagi menjadi bagian yang lebih kecil lagi.

Composite attribute yaitu atribut yang terdiri dari beberapa komponen, dimana masing-masing komponen memiliki keberadaan yang independen.

Single-valued attribute yaitu atribut yang mempunyai nilai tunggal untuk setiap kejadian. Sedangkan *multi-valued* attribute yaitu yang mempunyai beberapa nilai untuk setiap kejadian.

Derived attribute yaitu atribut yang memiliki nilai yang dihasilkan dari satu atau beberapa atribut lainnya, dan tidak harus berasal dari satu entitas.

2.1.7.4 *Key*

Menurut Connolly (2005, p352-353), ada tiga jenis kunci yaitu:

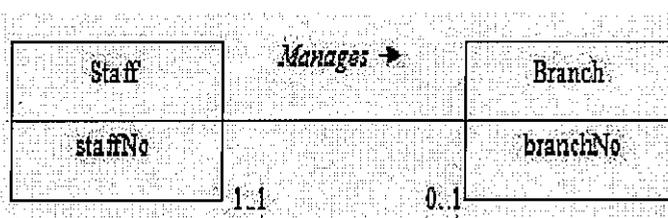
1. *Candidate key* yaitu jumlah minimal atribut-atribut yang secara unik mengidentifikasi setiap kejadian dari tipe entitas.
2. *Primary key* adalah *candidate key* yang dipilih sebagai kunci primer untuk mengenali secara unik setiap occurrence dari tipe entitas. Pemilihan *Primary key* untuk sebuah entitas adalah berdasarkan pada pertimbangan panjang atribut, jumlah minimal dari kebutuhan atribut dan memenuhi syarat unik.
3. *Composite key* yaitu kunci kandidat yang terdiri dari dua atau lebih atribut.

2.1.8 Struktural Constraint

Menurut Connolly (2005, p356), batasan utama pada *relationship* disebut *multiplicity*, yaitu jumlah (*range*) dari kejadian yang mungkin terjadi pada suatu entitas yang terhubung ke satu kejadian dari entitas lain yang berhubungan melalui suatu *relationship*. Relasi yang paling umum adalah *binary relationship*.

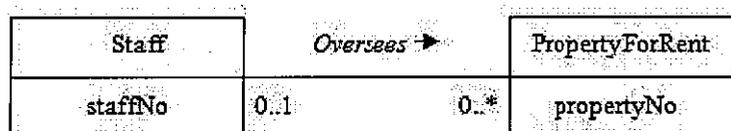
Macam-macam *binary relationship*:

1. One to One (1 : 1)



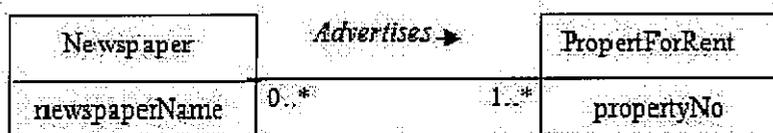
Gambar 2.8 One To One Relationship

2. One to Many (1...*)



Gambar 2.9 One to Many Relationship

3. Many to Many (* : *)



Gambar 2.10 Many to Many Relationship

Multiplicity dibentuk dari dua macam batasan pada relationship yaitu (Connolly, 2005, p363):

- a. *Cardinality*, menjelaskan jumlah maksimum dari kejadian relasi yang mungkin untuk entitas yang berpartisipasi di dalam relasi tersebut.
- b. *Participation*, merupakan apakah seluruh atau hanya sebagian entitas yang berpartisipasi dalam suatu relasi.

2.1.9 Normalisasi

Menurut Conolly (2005, p387-388), normalisasi adalah suatu teknik untuk menghasilkan suatu relasi dengan properti-properti yang sesuai dengan persyaratan data yang diberikan oleh sebuah perusahaan.

Tujuan dari normalisasi adalah untuk menghilangkan kerangkapan data, untuk mengurangi kompleksitas dan untuk mempermudah memodifikasi data.

Adapun Manfaat dari normalisasi adalah sebagai berikut:

1. Meminimalkan jumlah *storage space* yang diperlukan untuk menyimpan data.
2. Meminimalkan resiko data yang tidak konsisten dalam suatu basisdata.
3. Meminimalkan kemungkinan *update* dan *delete anomaly*.
4. Memaksimalkan stabilitas dari struktur data.

Proses-proses normalisasi :

1. Bentuk Tidak Normal / Unnormalized Form (UNF)

Unnormalized Form adalah sebuah tabel yang mengandung satu atau lebih *repeating groups*.

2. Bentuk Normal Pertama / First Normal Form (1NF)

First Normal Form adalah sebuah relasi di dalam mana titik potong setiap baris dan kolom mengandung satu dan hanya satu nilai. Untuk mengubah *unnormalized table* menjadi *first normal form* (1NF), kita harus mengidentifikasi dan menghilangkan *repeating groups* dari tabel serta kolom yang dapat dihitung.

3. Bentuk Normal Kedua / Second Normal Form (2NF)

Second Normal Form (2NF) dibuat berdasarkan konsep *full functional dependency*. *Second Normal Form* (2NF) adalah sebuah relasi yang berada dalam first normal form dan setiap atribut *non-primary-key* secara penuh bergantung secara fungsional pada *primary key*.

4. Bentuk Normal Ketiga / Third Normal Form (3NF)

Third Normal Form (3NF) adalah sebuah relasi dalam bentuk 1NF dan 2NF dimana tidak ada atribut *non-primary-key*

yang bergantung secara transitif terhadap *primary key*. Proses normalisasi dari relasi 2NF ke 3NF melibatkan penghapusan ketergantungan transitif. Jika terdapat ketergantungan transitif, kita menghilangkan atribut yang bergantung secara transitif dari relasi dengan menggantikan atribut dalam relasi yang baru bersamaan dengan duplikasi dari determinannya.

5. Bentuk Normal Keempat / Fourth Normal Form (4NF)

Fourth Normal Form (4NF) adalah sebuah relasi di mana relasi tersebut berada pada bentuk *Boyce-Codd Normal Form (BCNF)* dan tidak terdapat dua atau lebih atribut yang memiliki ketergantungan nilai banyak (*multi valued dependency*). Meskipun *Boyce-Codd Normal Form* telah membuang beberapa *anomaly* tetapi tidak menutup kemungkinan bahwa hasil dari *Boyce-Codd Normal Form* tersebut akan memiliki dua atau lebih atribut yang bernilai banyak (*multi valued*).

2.2 Teori-teori Khusus

2.2.1 Pengertian Sumber Daya Manusia

Menurut Kamus Besar Bahasa Indonesia Daring, sumber daya manusia adalah potensi manusia yang dapat dikembangkan untuk proses produk.

Menurut Gomes (1995, p1), sumber daya manusia adalah salah satu sumber daya yang ada dalam organisasi, meliputi semua orang yang melakukan aktivitas.

Berdasarkan beberapa pengertian sumber daya manusia di atas, dapat disimpulkan bahwa sumber daya manusia adalah potensi manusia dalam sebuah organisasi yang melakukan aktivitas, yang dapat dikembangkan untuk proses produk.

2.2.2 Pengertian Departemen Sumber Daya Manusia

Menurut Kamus Besar Bahasa Indonesia Daring, departemen adalah cabang pekerjaan yang dikepalai oleh manajer tunggal.

2.2.3 Pengertian Manajemen Sumber Daya Manusia

Pengertian Manajemen Sumber daya Manusia menurut Mathis dan Jackson (2003, p4), human resource management (management sumber daya manusia) adalah rancangan dari sistem formal di dalam organisasi untuk memastikan efektifitas dan efisiensi penggunaan potensi manusia untuk mencapai tujuan organisasi.

Ada juga pengertian menurut Noe et al (2007, p2), human resource management (management sumber daya manusia) adalah peraturan-peraturan, latihan-latihan, dan sistem yang dapat mempengaruhi tingkah laku dan sikap dan kinerja kerja karyawan.

Sedangkan menurut Dessler (2005, p4), human resource management (management sumber daya manusia) adalah peraturan dan pelatihan yang diikutsertakan dalam memberikan solusi kepada aspek “orang” ataupun sumber daya dari posisi manajemen, termasuk perekrutan, screening, pelatihan, penghargaan, dan penilaian.

2.2.4 Pengertian Manajemen Personil

Menurut SAP01 Fundamentals (2006, 6-6), manajemen personil merupakan integrasi antara perekrutan dan administrasi personil yang memungkinkan data pelamar dikirim sebagai data karyawan yang harus dilampirkan.

2.2.5 Pengertian PHP

Berdasarkan Artikel Nimhost.com PHP adalah kependekan dari *HypertextPreprocessor*, PHP dapat digunakan bersama dengan HTML sehingga memudahkan dalam membuat aplikasi web dengan cepat. PHP dapat digunakan untuk menciptakan *dynamic website* baik itu yang memerlukan penggunaan database ataupun tidak.

2.2.5.1 Sejarah PHP

Menurut artikel Wikipedia.com, pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form*

Interpreted), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi *open source*, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang *interpreter* PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP dirubah menjadi akronim berulang *PHP: Hypertext Preprocessing*.

Pada pertengahan tahun 1999, Zend merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

2.2.5.2 Kelebihan PHP

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

2.2.6 MySQL

Menurut artikel Wikipedia.com, MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

2.2.6.1 Kelebihan MySQL

Berikut adalah kelebihan – kelebihan MySQL :

1. **Portabilitas.** MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. **Open Source.** MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
3. **Multiuser.** MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. **Performance tuning.** MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. **Jenis Kolom.** MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. **Perintah dan Fungsi.** MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. **Keamanan.** MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. **Skalabilitas dan Pembatasan.** MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta

5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. **Konektivitas.** MySQL dapat melakukan koneksi dengan 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 interface (antar muka) 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.
13. **Struktur tabel.** MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.2.7 XAMPP

Berdasarkan artikel Wikipedia.com, XAMPP adalah kepanjangan dari Apache, PHP, MySQL dan phpMyAdmin. XAMPP

adalah *PHP bundle* yang didalamnya sudah ada Apache, MySQL dan beberapa aplikasi yang mungkin dibutuhkan dalam mengembangkan aplikasi berbasis web antara lain filezilla, Mysql admin dan lain-lainnya. XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket.

2.2.8 LAN

Local Area Network biasa disingkat **LAN** menurut penjelasan artikel www.wikipedia.com adalah jaringan komputer yang jaringannya hanya mencakup wilayah kecil; seperti jaringan komputer kampus, gedung, kantor, dalam rumah, sekolah atau yang lebih kecil. Saat ini, kebanyakan LAN berbasis pada teknologi IEEE 802.3 Ethernet menggunakan perangkat switch, yang mempunyai kecepatan transfer data 10, 100, atau 1000 Mbit/s. Selain teknologi Ethernet, saat ini teknologi 802.11b (atau biasa disebut *Wi-fi*) juga sering digunakan untuk membentuk LAN. Tempat-tempat yang menyediakan koneksi LAN dengan teknologi *Wi-fi* biasa disebut *hotspot*.

Pada sebuah LAN, setiap node atau komputer mempunyai daya komputasi sendiri, berbeda dengan konsep *dump terminal*. Setiap komputer juga dapat mengakses sumber daya yang ada di LAN sesuai dengan hak akses yang telah diatur. Sumber daya tersebut dapat berupa data atau perangkat seperti printer. Pada LAN, seorang pengguna juga

dapat berkomunikasi dengan pengguna yang lain dengan menggunakan aplikasi yang sesuai.

Berbeda dengan Jaringan Area Luas atau Wide Area Network (WAN), maka LAN mempunyai karakteristik sebagai berikut :

1. Mempunyai pesat data yang lebih tinggi
2. Meliputi wilayah geografi yang lebih sempit
3. Tidak membutuhkan jalur telekomunikasi yang disewa dari operator telekomunikasi

Biasanya salah satu komputer di antara jaringan komputer itu akan digunakan menjadi server yang mengatur semua sistem di dalam jaringan tersebut.

2.2.9 Intranet

Berdasarkan www.wikipedia.com sebuah **intranet** adalah sebuah jaringan privat (*private network*) yang menggunakan protokol-protokol Internet (TCP/IP), untuk membagi informasi rahasia perusahaan atau operasi dalam perusahaan tersebut kepada karyawannya. Kadang-kadang, istilah intranet hanya merujuk kepada layanan yang terlihat, yakni situs web internal perusahaan. Untuk membangun sebuah intranet, maka sebuah jaringan haruslah memiliki beberapa komponen yang membangun Internet, yakni protokol Internet (Protokol TCP/IP, alamat IP, dan protokol lainnya), klien dan juga server. Protokol HTTP dan beberapa

protokol Internet lainnya (FTP, POP3, atau SMTP) umumnya merupakan komponen protokol yang sering digunakan.

Umumnya, sebuah intranet dapat dipahami sebagai sebuah "versi pribadi dari jaringan Internet", atau sebagai sebuah versi dari Internet yang dimiliki oleh sebuah organisasi.