

PERTEMUAN 2

STRUKTUR BASIS DATA

A. TUJUAN PEMBELAJARAN

Setelah mempelajari dengan baik dari materi ini, diharapkan Mahasiswa dapat mendefinisikan tipe-tipe data, struktur dan batasan data yang disimpan dalam basis data.

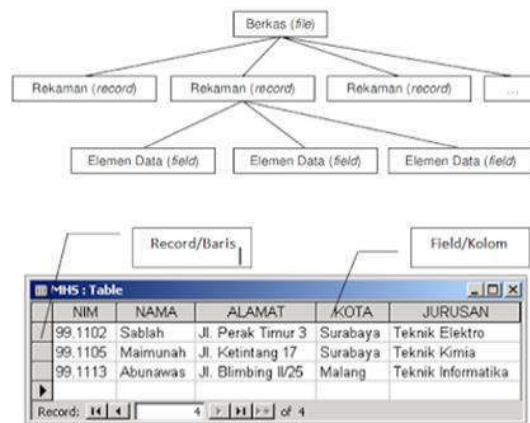
B. URAIAN MATERI

1. Struktur Basis Data

a. Struktur basis data

Konstruksi kumpulan data adalah bermacam-macam informasi tentang informasi yang didemonstrasikan. Informasi tentang record, tabel, *field*, list set, musyawarah informasi, dan berbagai ide yang digunakan untuk membuat gambaran desain kumpulan data. Menggambarkan desain basis informasi dapat memutuskan ide informasi, koneksi, dan persyaratan pada informasi yang ditangani. Dalam basis informasi, informasi dikoordinasikan menjadi komponen informasi (bidang), (catatan) dan dokumen (catatan). Pengertian dari ketiganya adalah sebagai berikut:

- 1) Elemen data (kolom atau bidang) adalah unit data terkecil yang tidak dapat dipecah menjadi unit lain yang bermakna. Data siswa terdiri dari misalnya NIS, nama, alamat, telepon atau jenis kelamin.
- 2) Sebuah record data adalah kombinasi dari beberapa elemen data yang terkait. Istilah lain dari record adalah line atau tuple.
- 3) Berkas (file) adalah himpunan seluruh record yang bertipe sama.

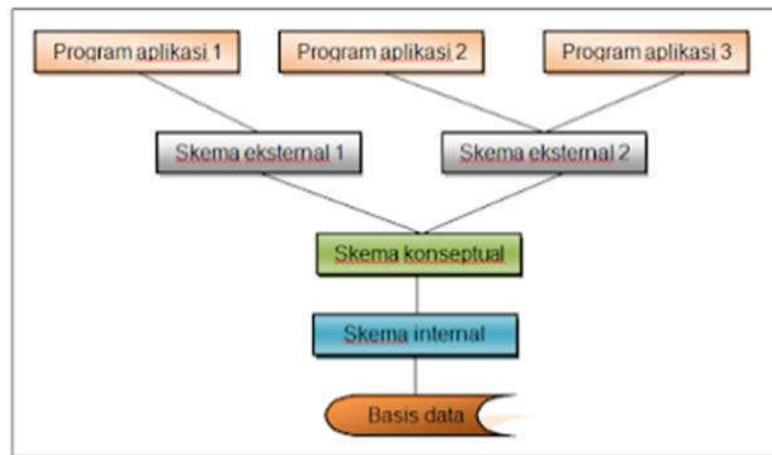


Gambar 0.1 Gambar Struktur Basis Data

b. Skema atau abstraksi data

Abstraksi data adalah tingkat atau cara informasi ditampilkan dalam kerangka kumpulan data. Abstraksi data diimplementasikan dalam penyajian informasi, ide-ide yang digunakan untuk membuat penggambaran struktur kumpulan data. Dengan menggambarkan struktur *database*, sifat data dan hubungannya dengan data lain dapat ditentukan. Penggambaran pemetaan basis informasi dari kumpulan data, penentuan yang ditunjukkan dalam tahap rencana. Komposisi ini digunakan untuk mengisolasi basis informasi aktual dan program aplikasi klien. Penggambaran skematis dari basis informasi biasanya diperkenalkan dalam bagan yang berisi beberapa seluk-beluk informasi dari penggambaran kumpulan data. Secara umum arsitektur *database* menggunakan arsitektur tiga skema yang mencakup tiga tingkatan, yaitu:

- 1) Tingkat *internal* atau skema *internal*. Tingkat ini merincikan penyimpanan kumpulan informasi dan akses informasi secara medalam. *Level* ini berisi penggambaran struktur penyimpanan basis informasi yang memanfaatkan model informasi aktual,
- 2) Tingkat konseptual (*conceptual schema*), berisi penggambaran seluruh desain kumpulan data untuk semua klien. *Level* ini berisi penggambaran elemen, koneksi, dan persyaratan tanpa menyertakan informasi poin demi poin.
- 3) Tingkat eksternal (skema atau tampilan eksternal mencirikan tampilan informasi kelompok klien (lingkungan) dengan menyembunyikan informasi lain yang tidak diperlukan oleh kelompok klien ini.



Gambar 0.2 Arsitektur Tiga Skema Manajemen Basis Data

c. Pemodelan data

Pemodelan data adalah metode untuk mengabstraksi informasi, sebuah ide yang digunakan untuk membuat penggambaran desain kumpulan data. Beberapa model informasi berisi rincian untuk aktivitas fundamental yang berhubungan dengan informasi yang sangat besar. Dalam perkembangan yang lebih baru, sifat data (perilaku data) dikenal dalam pemrograman berorientasi objek. Ada beberapa cara untuk merepresentasikan model untuk desain *database*. Secara umum pemodelan *database* dapat dibagi menjadi dua kelompok, yaitu:

- 1) Model logika berbasis objek. Dalam pemodelan ini, struktur atau hierarki *database* direpresentasikan menggunakan objek, model ini meliputi:
- 2) *Entity Relationship Model* (ERD).
- 3) Model data semantik (semantic data model). 2) Model data fungsional (functional data model). Record-based *logical* model. Dalam model ini struktur basis data diilustrasikan berdasarkan record. Model ini meliputi:
 - a) Model Racional
 - b) Model Herarkis
 - c) Model Jaringan

d. Model keterhubungan entitas (*Entity Relationship Model*/ERD)

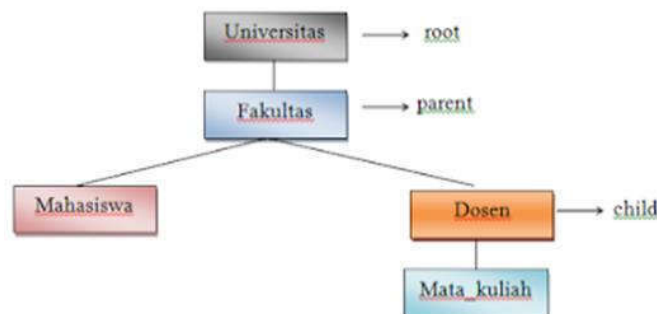
Diagram hubungan entitas atau *entity relationship* diagram (ERD) adalah grafik sebagai gambaran gambar yang membedakan jenis entitas

yang dijelaskan dalam data dalam suatu sistem dengan atribut-atributnya dan menggambarkan hubungan atau hubungan antar entitas tersebut.

ERD adalah model jaringan yang tindakan teoritis dari informasi yang disimpan dalam kerangka kerja. ERD adalah model data konseptual yang menangani informasi dalam sebuah asosiasi. ERD lebih menekankan pada tentang desain informasi dan koneksi. ERD digunakan oleh profesional sistem untuk berbicara dengan klien tingkat tinggi dalam organisasi atau asosiasi yang tidak tertarik untuk menjalankan kerangka kerja sehari-hari.

e. Model Hirarki Basis Data

Dalam model ini, informasi dikoordinasikan oleh struktur pohon. Titik tertinggi dari rantai kepentingan dikenal sebagai root, elemen atau antarmuka di bawahnya dikenal sebagai induk. Perusahaan induk memiliki beberapa pembantu yang dikenal sebagai pembantu. Entitas dalam model hierarki diwakili oleh persegi panjang. Sedangkan hubungan atau hubungan dengan entitas lain yang berbeda ditunjukkan oleh garis. Gambar berikut menjelaskan contoh model hierarki basis data tingkat konseptual untuk sistem perkuliahan



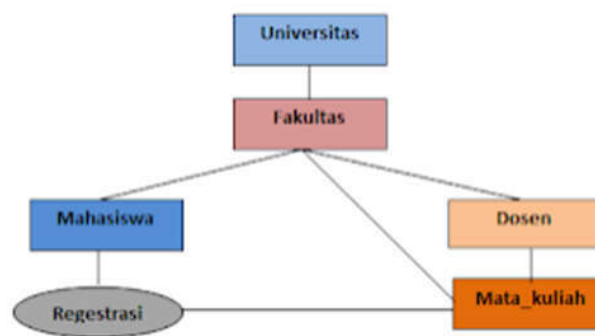
Gambar 0.3 Model Hirarki Basis Data Universitas

Dari gambar struktur hierarki *database* di atas, dapat dibuat struktur pengkodean record (*level* fisik) untuk setiap entitas dan hubungan antar entitas. Susunan hierarkis ditunjukkan oleh panah yang menunjuk ke data (bidang) yang digunakan sebagai kunci data (kunci utama, area penetasan). Hubungan dalam model hierarki hubungan antar elemen dikomunikasikan sebagai satu-ke-banyak (*one-to-many*) atau satu-ke-satu (*one-to-one*). Kelemahan model hierarkis adalah tidak dapat menemukan informasi di lapangan. Sifat. Misalnya, itu tidak dapat menampilkan informasi di tabel tarif

tergantung pada jumlah kredit karena ukuran kredit bukanlah kunci informasi. Masalah ini dapat diatasi dengan mengubah struktur informasi dengan menunjukkan koneksi yang tidak biasa (misalnya dengan variabel pointer).

f. Model Jaringan Basis Data

Dalam model jaringan, perusahaan induk dan anak perusahaan dapat terdiri dari beberapa unit.. Model ini merupakan pengembangan model hirarkis. Keterkaitan antar elemen dalam model jaringan adalah satu ke satu (*one to one*) atau satu ke banyak (*one to many*).



Gambar 0.4 Model Struktur Jaringan Basis Data

Hubungan banyak ke banyak tidak diperbolehkan dalam model data jaringan. Untuk membuat hubungan banyak ke banyak dalam model jaringan, diperlukan entitas perantara yang disebut catatan persimpangan. Dari gambar di atas terlihat bahwa entitas registrasi merupakan entitas perantara antara entitas mahasiswa dan entitas Mata kuliah.

g. Model Data Relasional

Model data relasional adalah model kumpulan informasi yang menggunakan tabel dua dimensi yang terdiri dari garis serta segmen untuk menggambarkan informasi dokumen. Model ini menunjukkan Administrasi aktual atau informasi otoritatif dalam kapasitas opsional. Ini juga mempengaruhi bagaimana klien mengumpulkan informasi dan membuat semua informasi yang terhubung dalam kerangka yang menjadi dasarnya.

2. Tipe-Tipe Data dalam Basis Data

memori yang dapat kita gunakan untuk menyimpan nilai suatu variabel.

Ada dua tipe data dalam basis data: tipe data *primitive* (sederhana) dan tipe data *composite*. Berikut ini pembahasan mengenai tipe data :

a. *Primitive* (sederhana)

Tipe data ini merupakan tipe data yang hanya dapat menyimpan satu nilai saja pada setiap variabelnya. Tipe data ini merupakan tipe data yang paling dasar dan sering digunakan untuk membuat sebuah program, berikut ini beberapa contoh tipe data primitive dan penjelasannya :

- 1) Tipe numerik, jenis ini digunakan untuk variabel sehingga mereka dapat menyimpan nilai dalam jenis angka. Tipe data ini dibagi menjadi “Real” dan “Integer”. Integer (int), merupakan tipe data bilangan bulat.

Tipe data	Ukuran Tempat	Range Nilai
Byte	1 byte	0 s/d +255
Shortint	1 byte	-28 s/d +127
Integer	2 byte	-32768 s/d 32767
Word	2 byte	0 s/d 65535
Longint	4 byte	2147483648 s/d 2147483647

Gambar 0.5 Tipe Data Numerik

- 2) Karakter (char) adalah tipe informasi yang dengannya kita hanya dapat menyimpan satu digit karakter, dengan alasan bahwa satu digit adalah 1 byte (dan 1 byte = 8 bit). Untuk menulis kita dapat menggunakan pernyataan tunggal (') sebelumnya dan sebelum karakter yang dikomposisikan. silahkan perhatikan gambar dibawah ini :

Jenis Karakter	Contoh penggunaan
Huruf	'a', 'b', 'D', 'Z'
Angka	'1', '2', '3' dst
Tanda baca	titik (.), koma (,), titik koma (;), dst)
Khusus	\$, %, #, @ dst

Gambar 0.6 Tipe Data Char

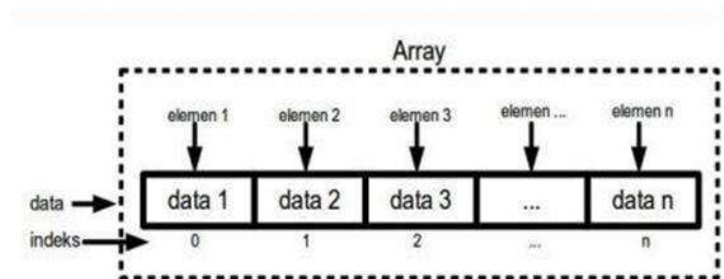
- 3) Berikutnya adalah tipe informasi Boolean, yang harus diatur sebagai tipe informasi yang sah karena satu nilai informasi valid atau palsu dan tipe informasi ini hampir tidak memakan memori. Contoh: $(7 > 3) = \text{true}$.

b. *Composite*

Tipe data ini sesuatu yang Berlawanan dengan tipe data primitif karena tipe informasi ini dapat berisi banyak kualitas, berikut adalah beberapa tipe informasi majemuk dan penjelasannya:

a) *Array*

array merupakan tipe informasi yang terorganisir, meskipun tipe informasi ini bersifat dasar namun dapat menyimpan informasi sejenis dalam suatu variabel.



Gambar 0.7 Tipe Data Char

b) *Record*

Kumpulan informasi ini dapat diakses dalam bahasa pemrograman Pascal atau C++. berbeda dari array. record dapat berisi tipe data yang berbeda.

Contoh

```

type
  data_pegawai = record
    kd_peg : string[5];
    nama   : string[15];
    alamat : string[20];
    kota   : string[20];
    gaji   : longint;
  end;
var
  pegawai : data_pegawai;
```

Gambar 0.8 Tipe Data Record

c) Image

Image merupakan tipe data grafis, contohnya grafik pertumbuhan atau penurunan penduduk, dalam bahasa berbasis visual tipe data ini sangat mendukung.

d) Date Time

Tanggal dan waktu ini disimpan dalam organisasi tertentu karena variabel yang diumumkan atau tetap dapat berupa jenis informasi tanggal atau waktu. Jenis informasi ini terdiri dari beberapa jenis informasi, dengan alasan bahwa ini dikenal sebagai jenis informasi gabungan.

No	Nama	Fungsi	Jangkauan	Ukuran
1	DATE	menyimpan data tanggal	1000-01-01 s/d 9999-12-31 (YYYY-MM-DD)	3 byte
2	TIME	menyimpan data waktu	-838:59:59 s/d +838:59:59 (HH:MM:SS)	3 byte
3	DATETIME	menyimpan data tanggal dan waktu	1000-01-01 00:00:00 s/d 9999-12-31 23:59:59	8 byte
4	YEAR	menyimpan data tahun dari tanggal	1900 s/d 2155	1 byte

Gambar 0.9 Tipe Data Date Time

3. Komponen-Komponen dalam DBMS

Dalam kumpulan data, akan ada segmen utama yang menyertainya diantaranya:

a. Perangkat keras (Hardware)

Perangkat keras terdiri dari perangkat penyimpanan sekunder seperti disk, drum, dan sebagainya, di mana basis data berada bersama dengan perangkat lain. Ada dua jenis perangkat keras. Yang pertama yaitu prosesor dan memori utama yang mendukung dalam menjalankan DBMS. Yang kedua adalah perangkat penyimpanan sekunder, penyimpanan sekeunder terbagi dua yaitu, Penyimpanan sekunder online (hard disk) dan penyimpanan sekunder offline (media yang dapat dilepas, floppy disk, flash disk, hard disk eksternal, CD) untuk backup data. Kemudian media/perangkat komunikasi (untuk sistem jaringan).

b. Sistem Operasi (*operating System*)

Dalam istilah dasar, *operating system* adalah program yang memulai/elementer kerangka kerja komputer, mengontrol setiap aset pada komputer, dan melakukan prosedur dasar pada komputer. (Operasi I/O, manajemen file, dll.). Sistem operasi MS-Dos, MS-Windows (untuk komputer yang berdiri sendiri atau untuk komputer klien dalam sistem jaringan), Unix, Xenix, Sun-Solaris, Linux, MS-Windows NT (untuk komputer dalam sistem) tersebar luas. Jaringan), Apple Mac OS, Macintosh, OS / 2. Program pengelola basis data hanya dapat aktif ketika sistem operasi yang diinginkan (cocok) aktif.

c. Basis data

Sebuah sistem *database* dapat memiliki beberapa *database*. Setiap *database* dapat berisi / memiliki sejumlah objek *database* (seperti tabel, indeks, dll). Setiap *database* berisi/menyimpan tidak hanya data, tetapi juga definisi struktur (baik untuk *database* maupun untuk objeknya secara detail).

d. Sistem (Aplikasi/Software/Perangkat lunak) pengelola basis data (Basis data manajemen sistem-DBMS)

Lapisan atau antarmuka perangkat lunak ada di antara *database* fisik dan pengguna. Lapisan ini disebut DBMS. Semua permintaan dari pengguna untuk mengakses *database* ditangani oleh DBMS. Jadi, DBMS melindungi pengguna *database* dari detail perangkat keras. Selain itu, DBMS juga menyediakan fasilitas lain seperti mengakses dan memperbarui data di dalam file serta menambah dan menghapus file itu sendiri. Perangkat lunak yang merupakan DBMS, seperti dBase, Borland-paradox, MS-Acces, Borland-Interbase, MS-SQL Server, MySQL, Sybase, CA-Open ingres, Informix, Oracle, dan firebird.

e. Pemakai (User)

Pengguna adalah orang-orang yang berinteraksi dengan sistem *database* dengan cara apa pun. Ada empat jenis pengguna yang berinteraksi dengan sistem *database*. Ini adalah Pemrogram Aplikasi, pengguna online, pengguna akhir atau pengguna naif dan terakhir Administrator Basis Data (DBA).

DBMS ini memiliki beberapa komponen atau modul fungsional. Dan berikut adalah 6 komponen yang ada pada DBMS :

a. *Database manager*

Komponen ini dapat memberikan antarmuka antara informasi tingkat rendah, yang ada di setiap kumpulan data, dan program aplikasi seperti halnya pertanyaan yang diberikan kepada kerangka kerja. Dalam istilah dasar, administrator kumpulan data ini memiliki tugas untuk memiliki opsi untuk menguji dan mendapatkan permintaan dari jarak jauh dan konstruksi teoretis untuk memutuskan apakah catatan informasi diharapkan memenuhi ajakan tersebut.

b. *File manager*

File manager adalah komponen yang dapat mengatur ruang yang tersedia pada hard drive dan struktur data yang dapat digunakan untuk mewakili informasi yang disimpan pada hard drive. Atau, bisa dibilang file manager dapat digunakan untuk memanipulasi penyimpanan dalam file dan mengatur alokasi ruang pada hard drive. Manajer file juga memainkan peran utama dan bertanggung jawab untuk penyimpanan data.

c. *Query Processor*

Pada komponen DBMS ini dapat berfungsi sebagai penerjemahan suatu perintah dalam bahasa query menjadi perintah tingkat rendah dan dapat dipahami oleh pengelola *database*.

d. *DDL Compiler*

Compiler DDL ini merupakan komponen yang dapat mengubah berbagai perintah DDL menjadi kumpulan tabel yang sudah berisi data. Sederhananya, tugas dari compiler DDL ini adalah untuk mengubah instruksi tingkat tinggi, atau instruksi tingkat tinggi, yang kemudian menjadi bahasa tingkat rendah yang akhirnya menerima sekumpulan label yang sudah berisi metadata.

e. *DML Compiler*

Kompiler DML adalah komponen yang dapat mengubah perintah dari DML yang kemudian ditambahkan ke program aplikasi untuk menjalankan prosedur normal dalam bahasa induk. Dan bisa juga disebut DML compiler,

ini merupakan komponen dari DBMS yang dapat mengubah perintah dari embedded DML menjadi program aplikasi berupa fungsi-fungsi yang dipanggil dalam bahasa host.

f. Dictionary Manager

Dictionary manager adalah komponen DBMS yang dapat mengatur akses ke dan mengelola data kamus.

4. Karakteristik Basis Data

Sejumlah karakteristik membedakan pendekatan *database* dari pendekatan pemrograman yang jauh lebih lama dengan file. Dalam pemrosesan file tradisional, setiap pengguna mendefinisikan dan mengimplementasikan file yang diperlukan untuk aplikasi perangkat lunak tertentu sebagai bagian dari pemrograman aplikasi. Misalnya, satu pengguna, kantor pelaporan nilai, dapat menyimpan file tentang siswa dan nilai mereka. Program untuk mencetak transkrip nilai siswa dan untuk memasukkan nilai baru diimplementasikan sebagai bagian dari aplikasi. Pengguna kedua, kantor akuntansi, dapat melacak biaya siswa dan pembayaran mereka. Meskipun kedua pengguna tertarik dengan data tentang siswa, setiap pengguna menyimpan file terpisah dan program untuk memanipulasi file ini — karena masing-masing memerlukan beberapa data yang tidak tersedia dari file pengguna lain. Redundansi dalam menentukan dan menyimpan hasil data di ruang penyimpanan yang terbuang dan dalam upaya yang berlebihan untuk mempertahankan data terkini yang umum. Dengan metode kumpulan data, repositori tunggal mengawasi informasi yang ditandai sekali dan kemudian diakses oleh berbagai klien. Dalam kerangka catatan, setiap aplikasi diizinkan untuk memberi nama komponen informasi secara mandiri satu sama lain. Menariknya, dalam kumpulan data, nama atau tanda informasi dicirikan sekali dan digunakan berulang kali oleh pertanyaan, pertukaran, dan aplikasi. Karakteristik utama dari pendekatan *database* versus pendekatan pemrosesan file adalah sebagai berikut:

a. Sifat yang mendeskripsikan diri sendiri dari sistem *database*;

Sifat penting dari metodologi kumpulan data adalah bahwa kerangka basis informasi berisi kumpulan data itu sendiri, namun juga definisi atau gambaran total dari desain kumpulan data dan batasannya. Definisi ini

disimpan dalam daftar DBMS, yang berisi data seperti desain setiap record, jenis dan pengaturan kapasitas setiap hal informasi, dan batasan yang berbeda pada informasi. Data yang disimpan dalam inventaris dikenal sebagai *database* dan menggambarkan desain kumpulan data dasar. Katalog Digunakan oleh perangkat lunak DBMS dan oleh pengguna *database* yang membutuhkan informasi tentang struktur *database*. Software DBMS tidak ditulis untuk aplikasi *database* tertentu. Oleh karena itu, harus mengacu pada katalog untuk mengetahui struktur file dalam *database* tertentu, seperti jenis dan format data yang akan diaksesnya. Software DBMS harus bekerja sama dengan sejumlah aplikasi basis data. Misalnya, basis data universitas, basis data perbankan, atau basis data perusahaan. Selama definisi basis data disimpan dalam katalog.

Dalam pemrosesan file tradisional, definisi data biasanya merupakan bagian dari program aplikasi itu sendiri. Oleh karena itu, program ini dibatasi untuk bekerja hanya dengan satu *database* tertentu, yang strukturnya dideklarasikan dalam program aplikasi. Misalnya, program aplikasi yang ditulis dalam C ++ mungkin memiliki deklarasi struct atau kelas, dan program COBOL memiliki pernyataan pembagian data untuk menentukan file-nya. Sedangkan Software pengolah file hanya dapat mengakses *database* tertentu, Software DBMS dapat mengakses *database* yang beragam dengan mengekstrak definisi *database* dari katalog dan menggunakan definisi ini.

b. Isolasi antara program dan data, dan abstraksi data;

Dalam pemrosesan file tradisional, struktur file data dipasang di dalam program aplikasi, sehingga setiap perubahan dalam struktur file mungkin memerlukan perubahan untuk dilakukan pada semua program yang mengakses file tersebut. Sebaliknya, program akses DBMS tidak memerlukan perubahan seperti itu dalam banyak kasus. Struktur file disimpan dalam katalog DBMS secara terpisah dari program akses. Kami menyebut properti ini sebagai independensi data program.

Dalam beberapa tipe sistem basis data, seperti sistem berorientasi objek dan sistem relasional objek, pengguna dapat mendefinisikan operasi pada data sebagai bagian dari definisi basis data. Operasi (juga disebut fungsi atau metode) didefinisikan dalam dua bagian. Antarmuka (atau tanda tangan) operasi berisi nama operasi dan tipe data argumen (atau

parameter). Implementasi (atau metode) didefinisikan secara terpisah dan dapat diubah tanpa mempengaruhi antarmuka. Program aplikasi pengguna dapat bekerja dengan data dengan melakukan operasi ini dengan nama dan argumennya, terlepas dari bagaimana operasi tersebut diimplementasikan. Hal ini dapat disebut sebagai kemandirian operasi program.

Fitur yang memungkinkan kemandirian operasi program dan kemandirian data program disebut abstraksi data. DBMS menyediakan pengguna dengan representasi konseptual data yang tidak mengandung banyak detail tentang bagaimana data disimpan atau bagaimana operasi diimplementasikan. Secara informal, model data adalah jenis abstraksi data yang digunakan untuk menyediakan representasi konseptual ini. Model data menggunakan konsep logistik seperti objek, propertinya, dan hubungannya, yang mungkin lebih mudah dijangkau oleh sebagian besar pengguna daripada konsep penyimpanan komputer. Karenanya, model data menyembunyikan detail penyimpanan dan implementasi yang tidak menarik bagi sebagian besar pengguna *database*.

c. Mendukung berbagai tampilan data;

Kumpulan data secara teratur memiliki berbagai klien, yang masing-masing mungkin memerlukan sudut pandang atau perspektif alternatif pada basis informasi. Perspektif mungkin penting untuk basis informasi atau mungkin berisi informasi virtual yang berasal dari catatan kumpulan data namun tidak secara tegas disingkirkan. Beberapa klien tidak perlu tahu apakah informasi yang mereka singgung disimpan atau diminimalkan. Sebuah DBMS multi-klien yang kliennya memiliki berbagai aplikasi harus memberikan kantor untuk mengkarakterisasi perspektif yang berbeda. Misalnya, salah satu pengguna *database* mungkin hanya tertarik untuk mengakses dan mencetak transkrip setiap siswa; Pengguna kedua, yang hanya tertarik untuk memeriksa bahwa siswa telah mengambil semua prasyarat dari setiap mata kuliah yang mereka daftarkan, mungkin memerlukan tampilan Berbagi data dan pemrosesan transaksi multipengguna.

d. Berbagi Data dan Pemrosesan Transaksi Multiuser

Seperti namanya, DBMS multi-pengguna harus mengizinkan klien yang berbeda untuk sampai ke basis informasi secara bersamaan. Ini

penting ketika informasi untuk berbagai aplikasi akan digabungkan dan diawasi dalam satu set data tunggal. DBMS harus menggabungkan pemrograman kontrol simultan untuk menjamin bahwa berbagai klien mencoba informasi itu sendiri dengan cara yang menjamin hasil yang benar. Misalnya, jika spesialis yang berbeda berusaha memesan untuk menunjuk kursi ke maskapai penerbangan hanya pada penerbangan, DBMS harus menjamin bahwa setiap kursi dapat digunakan oleh satu kesempatan untuk dibagikan kepada wisatawan. Penggunaan semacam ini umumnya disepakati sebagai aplikasi pertukaran online (OLTP). Tugas utama pemrograman DBMS multi-klien adalah untuk menjamin bahwa pertukaran simultan bekerja secara efektif dan mahir.

C. SOAL LATIHAN/TUGAS

1. Sebutkan dan jelaskan Struktur basis data ?
2. Sebutkan komponen-komponen dalam DBMS ?
3. Berikan contoh dari Data Description Language, dan Data Query Language ?
4. Jelaskan secara singkat perbedaan DDBMS dan CDBMS ?
5. Jelaskan mengenai karakteristik basis data ?
6. Sebutkan dan jelaskan tipe data dalam basis data ?
7. Jelaskan perbedaan DDL Compiler dengan DML Compiler dan berikan contohnya ?

D. REFERENSI

- Elmasri, R. & Navathe, S.B., 2011. *Fundamentals of Database System Sixth Edition*. Boston: Addison-Wesley.
- Dantes, Gede Rasben, dkk., 2019. *Pengantar Basis data*. Depok : Rajawali Pers.

GLOSARIUM

Object based *logical* model Adalah sekumpulan information dan prosedur atau hubungan yang menjelaskan hubungan logis antar information dalam data set berdasarkan objek information.

DDL Compiler adalah komponen yang dapat mengubah berbagai perintah DDL menjadi kumpulan tabel yang sudah berisi metadata.

DML Compiler adalah komponen yang dapat mengubah perintah dari DML.

Pengelola kamus adalah komponen DBMS yang dapat mengatur akses ke dan mengelola data kamus