



Institut Informatika & Bisnis
DARMAJAYA
Yayasan Alfian Husin



KEMENTERIAN
PENDIDIKAN DAN KEBUDAYAAN

**Kampus
Merdeka**
INDONESIA JAYA

BUKU AJAR

KODE MATAKULIAH : TIF20406

DATABASE TECHNOLOGY

Oleh:

**Yuni Arkhiansyah
Muhammad Fauzan Azima**



**PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT INFORMATIKA DAN BISNIS (IIB) DARMAJAYA**

KATA PENGANTAR

Alhamdulillah, segala puji syukur penulis haturkan kehadirat ALLAH SWT yang mana atas rahmad dan karunianyalah buku ini dapat selesai disusun. Buku ini ditujukan agar dapat membantu mahasiswa/i dalam mempelajari Database Technology dan mempermudah mempelajari materi perkuliahan terutama bagi mahasiswa yang ingin mengenal lebih jauh dari pemahaman tentang basis data.

Penulis pun menyadari sebagai manusia biasa bahwa dalam penyusunan buku ini tidak terlepas dari banyak kekurangan ataupun kekhilafan, namun penulis sepenuhnya akan sekecil apapun buku ini akan dapat memberikan manfaat bagi pembaca.

Maka dengan kerendahan hati demi penyempurnaan buku ini kritik dan saran dari pembaca sangat berguna untuk penulis kedepannya.

Sekian semoga buku ajar ini dapat bermanfaat bagi penulis khususnya dan para pembaca pada umumnya.

Bandar Lampung, 03 Desember 2020

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	v
DAFTAR TABEL.....	ix
BAB I PENGANTAR BASIS DATA.....	1
1.1 Pendahuluan.....	1
1.2 Pengenalan Basis Data.....	2
1.3 Konsep Dasar Basis Data.....	4
1.4 Operasi Dasar Basis Data.....	6
1.5 Sistem basis Data.....	7
1.6 Data Base Management System (DBMS).....	9
1.7 Komponen Sistem Basis Data.....	10
1.8 Komponen Sebuah DBMS.....	12
1.9 Keuntungan dan Kekurangan DBMS.....	18
BAB II KONSEP DAN ARSITEKTUR BASIS DATA.....	20
2.1 Pendahuluan.....	20
2.2 Basis Data Relational.....	20
2.3 Perancangan Basis Data.....	30
2.4 Studi Kasus.....	32
BAB III BASIS DATA RELATIONAL.....	34
3.1 Pendahuluan.....	34
3.2 Pengertian Model Data.....	34
3.3 Jenis-jenis Model Data.....	35
3.4 Model Data Semantik (Semantik Data Model).....	41
BAB IV KONTROL USER INTERFACE.....	46
4.1 Pendahuluan.....	46
4.2 Pengertian Entity Relationship Diagram (ERD).....	47
4.3 Komponen ERD.....	47

4.4 Kardinalitas / Derajat Relasi	47
4.5 Tahapan Pembuatan ERD	47
4.6 Logical Record Structured (LRS)	47
4.7 Membuat ERD	47
BAB V ACTIVITY DAN INTENT.....	49
5.1 Pendahuluan.....	49
5.2 Pengertian Normalisasi	49
5.3 Anormly	49
5.4 Atribut dan Ketergantungan Fungsi	49
5.5 Bentuk Normalisasi.....	49
a. Sadfasfas	50
b. Asdfasdfsaf.....	50
c. Sadfsadf.....	50
BAB VI Shared Preference, file dan Penyimpanan Data	51
6.1 Pendahuluan.....	51
6.2 Langkah-langkah pembuatan Normalisasi.....	51
6.3 Studi Kasus	51
BAB VII APLIKASI CRUD DATABASE.....	53
7.1 Pendahuluan.....	53
7.2 Pengertian Bahasa Query Formal	53
7.3 Operator Aljabar Relational.....	53
BAB VIII PENGEMBANGAN PERANGKAT LUNAK APLIKASI ANDROID.....	55
8.1 Pendahuluan.....	55
8.2 Structured Query Language (SQL).....	55
BAB IX PENGEMBANGAN PERANGKAT LUNAK APLIKASI ANDROID	57
9.1 Pendahuluan.....	57
9.2 Join.....	57
9.3 Fungsi Agregat.....	57
9.4 SubQuery	57

BAB X.....	59
10.1Pendahuluan.....	59
10.2Pengertian Basis Data Terdistribusi.....	59
10.3Topologi Distribusi Data.....	59
10.4Fragmentasi Data	59
BAB XI.....	60
11.1Pendahuluan.....	60
11.2K.....	60
BAB XII.....	61
12.1Pendahuluan.....	61
12.2Konkurensi	62
12.3Locking	62
12.4Timestamping	62
12.5Crass dan Recovery.....	62
12.6Security	62
12.7Pemberian Wewenang dan View	62
12.8Integrity.....	62
DAFTAR PUSTAKA	63

DAFTAR GAMBAR

- Gambar 1.1 Arsitektur Sistem Operasi Android....**Error! Bookmark not defined.**
- Gambar 1.2 Cocoa Touch Layer**Error! Bookmark not defined.**
- Gambar 2.1 HTC Dream berbasis Android.....**Error! Bookmark not defined.**
- Gambar 2.2 Halaman Download Java.....**Error! Bookmark not defined.**
- Gambar 2.3 Halaman persetujuan syarat dan ketentuan java**Error! Bookmark not defined.**
- Gambar 2.4 Memulai proses instalasi java**Error! Bookmark not defined.**
- Gambar 2.5 konfirmasi penginstalan java.....**Error! Bookmark not defined.**
- Gambar 2.6 tampilan pertama instalasi java**Error! Bookmark not defined.**
- Gambar 2.7 Menunggu proses instalasi java.....**Error! Bookmark not defined.**
- Gambar 2.8 Selesai proses instalasi java.....**Error! Bookmark not defined.**
- Gambar 2.9 Cek java version pada komputer**Error! Bookmark not defined.**
- Gambar 2.10 Memilih Java Development Kit Installer**Error! Bookmark not defined.**
- Gambar 2.11 Agreement Download JDK**Error! Bookmark not defined.**
- Gambar 2.12 Wizard Instalasi JDK**Error! Bookmark not defined.**
- Gambar 2.13 Setting Folder Instalasi JDK**Error! Bookmark not defined.**
- Gambar 2.14 Menunggu Proses Instalasi JDK**Error! Bookmark not defined.**
- Gambar 2.15 Instalasi JDK Complete.....**Error! Bookmark not defined.**
- Gambar 2.16 Halaman Unduh Android Studio.....**Error! Bookmark not defined.**
- Gambar 2.17 Halaman instalasi android studio**Error! Bookmark not defined.**
- Gambar 2.18 Memilih komponen penginstalan**Error! Bookmark not defined.**
- Gambar 2.19 Konfigurasi lokasi penginstalan Android Studio**Error! Bookmark not defined.**
- Gambar 2.20 Setting Start Menu Folder**Error! Bookmark not defined.**
- Gambar 2.21 Proses Instalasi Android Studio**Error! Bookmark not defined.**
- Gambar 2.22 Halaman Welcome Android Studio .**Error! Bookmark not defined.**
- Gambar 2.23 Halaman Instalasi Tipe Standard / Custom**Error! Bookmark not defined.**

Gambar 2.24 Memilih UI Theme Dracula / Light .**Error! Bookmark not defined.**

Gambar 2.25 Menentukan lokasi Penginstalan SDK**Error! Bookmark not defined.**

Gambar 2.26 Menentukan jumlah Memory Emulator Settings**Error! Bookmark not defined.**

Gambar 2.27 Verify Setting Android Studio**Error! Bookmark not defined.**

Gambar 2.28 Proses Download dan Instalasi Komponen Pendukung **Error! Bookmark not defined.**

Gambar 2.29 Halaman pertama ketika Android Studio Selesai Di Instalasi . **Error! Bookmark not defined.**

Gambar 2.30 Aplikasi Android Studio pada menu Start**Error! Bookmark not defined.**

Gambar 2.31 Halaman awal Android Studio**Error! Bookmark not defined.**

Gambar 2.32 Memilih Project Template.....**Error! Bookmark not defined.**

Gambar 2.33 Mengkonfigurasi Project Baru**Error! Bookmark not defined.**

Gambar 2.34 Halaman Pertama Project Baru**Error! Bookmark not defined.**

Gambar 2.35 Struktur File Project**Error! Bookmark not defined.**

Gambar 2.36 MainActivity.java.....**Error! Bookmark not defined.**

Gambar 2.37 Test Aplikasi Pertama dengan Toast**Error! Bookmark not defined.**

Gambar 3.1 User Interface Sederhana**Error! Bookmark not defined.**

Gambar 3.2 Contoh Layout User Interface**Error! Bookmark not defined.**

Gambar 3.3 Vertical Linear Layout**Error! Bookmark not defined.**

Gambar 3.4 Horizontal Linear Layout**Error! Bookmark not defined.**

Gambar 3.5 Relative Layout**Error! Bookmark not defined.**

Gambar 3.6 Frame Layout**Error! Bookmark not defined.**

Gambar 3.7 Constraint Layout.....**Error! Bookmark not defined.**

Gambar 3.8 ListView Nama-nama Negara**Error! Bookmark not defined.**

Gambar 4.1 TextView.....**Error! Bookmark not defined.**

Gambar 4.2 Tampilan Textview**Error! Bookmark not defined.**

Gambar 4.3 EditText.....**Error! Bookmark not defined.**

Gambar 4.4 Tampilan EditText.....**Error! Bookmark not defined.**

Gambar 4.5 Button	Error! Bookmark not defined.
Gambar 4.6 Tampilan Button.....	Error! Bookmark not defined.
Gambar 4.7 Tampilan CheckBox.....	Error! Bookmark not defined.
Gambar 4.8 Tampilan Radio Button Jenis Kelamin	Error! Bookmark not defined.
	defined.
Gambar 4.9 Tampilan ToggleButton	Error! Bookmark not defined.
Gambar 4.10 Tampilan Switch	Error! Bookmark not defined.
Gambar 4.11 Aritmatika Penjumlahan dua Buah Bilangan	Error! Bookmark not defined.
	defined.
Gambar 4.12 Custom ListView.....	Error! Bookmark not defined.
Gambar 4.13 Membuat Activity Baru.....	Error! Bookmark not defined.
Gambar 4.14 Activity CustomListView	Error! Bookmark not defined.
Gambar 4.15 Layout Resource Baru	Error! Bookmark not defined.
Gambar 4.16 Membuat item_list.xml	Error! Bookmark not defined.
Gambar 4.17 item_list yang baru terbentuk	Error! Bookmark not defined.
Gambar 4.18 Tampilan Layout item_list.xml	Error! Bookmark not defined.
Gambar 4.19 Custom ListView.....	Error! Bookmark not defined.
Gambar 5.1 Perumpamaan Activity pada Web dan Desktop	Error! Bookmark not defined.
	defined.
Gambar 5.2 Activity LifeCycle.....	Error! Bookmark not defined.
Gambar 5.3 Activity Implicit dan Explicit Activity	Error! Bookmark not defined.
Gambar 5. 4 Aksi Implicit Intent	Error! Bookmark not defined.
Gambar 5. 5 Explicit Intent ke halaman Activity Dua	Error! Bookmark not defined.
	defined.
Gambar 5.6 Menambahkan Tombol Intent dengan Extra	Error! Bookmark not defined.
	defined.
Gambar 5.7 Penambahan Intent dengan Extra	Error! Bookmark not defined.
Gambar 5.8 Menampilkan Intent PutExtra denan Toast	Error! Bookmark not defined.
	defined.
Gambar 5.9 Media Penyimpanan Data	Error! Bookmark not defined.
Gambar 5.10 Cara Kerja Shared Preferences.....	Error! Bookmark not defined.

Gambar 5. 11 Tampilan halaman aplikasi DB Browser for SQLite **Error! Bookmark not defined.**

Gambar 5.12 Menambahkan Permission Storage ..**Error! Bookmark not defined.**

Gambar 5.13 Cloud Storage Firebase**Error! Bookmark not defined.**

Gambar 6.1 Content Profiders.....**Error! Bookmark not defined.**

Gambar 6.2 Membuat Database Baru**Error! Bookmark not defined.**

Gambar 6.3 Menentukan tempat penyimpanan database SQLite**Error! Bookmark not defined.**

Gambar 6.4 Membuat tabel daftar doa harian.....**Error! Bookmark not defined.**

Gambar 6.5 Membuat dan menyimpan tabel daftar_doa_harian**Error! Bookmark not defined.**

Gambar 6.6 Menjalankan perintah SQL pada SQLite**Error! Bookmark not defined.**

Gambar 7.1 activity_aplikasi_crud_activity.xml ...**Error! Bookmark not defined.**

Gambar 7.2 Menambahkan dependencies.....**Error! Bookmark not defined.**

Gambar 7.3 Membuat package dbhelper**Error! Bookmark not defined.**

Gambar 7.4 Membuat Java Class DatabaseAccess**Error! Bookmark not defined.**

Gambar 7.5 Membuat package model**Error! Bookmark not defined.**

Gambar 7.6 Membuat Java Class ModelCrud.java**Error! Bookmark not defined.**

Gambar 7.7 Membuat package recylerview.....**Error! Bookmark not defined.**

Gambar 7.8 Membuat Java Class DatabaseRecyclerCRUD**Error! Bookmark not defined.**

Gambar 7.9 Build dan Run Aplikasi**Error! Bookmark not defined.**

Gambar 7.10 Aplikasi CRUD Doa Harian.....**Error! Bookmark not defined.**

Gambar 7.11 Tampilan Tambah Data Doa**Error! Bookmark not defined.**

Gambar 7.12 Menyimpan data doa**Error! Bookmark not defined.**

Gambar 7.13 Tampilan View Doa**Error! Bookmark not defined.**

Gambar 7.14 Tampilan halaman update data doa..**Error! Bookmark not defined.**

Gambar 8.1 Metode Waterfall.....**Error! Bookmark not defined.**

Gambar 8.2 Metode Pengembangan Agile**Error! Bookmark not defined.**

Gambar 8.3 Scrum Process**Error! Bookmark not defined.**

Gambar 8.4 Rapid Application Development (RAD)**Error! Bookmark not defined.**

Gambar 8. 5 Diagram Model Prototype**Error! Bookmark not defined.**

Gambar 8.6 Metode DevOps.....**Error! Bookmark not defined.**

Gambar 8.7 Use Case Diagram.....**Error! Bookmark not defined.**

Gambar 8.8 Contoh Activity Diagram**Error! Bookmark not defined.**

Gambar 8.9 Contoh Class Diagram**Error! Bookmark not defined.**

Gambar 8.10 Build Keystore.....**Error! Bookmark not defined.**

Gambar 8.11 Memilih tipe signed apk.....**Error! Bookmark not defined.**

Gambar 8.12 Membuat Keystore Baru**Error! Bookmark not defined.**

Gambar 8.13 Memberi nama dan menentukan lokasi penyimpanan keystore
.....**Error! Bookmark not defined.**

Gambar 8.14 Mengisi form isian pembuatan keystore**Error! Bookmark not defined.**

Gambar 8.15 Mengisi keystore password dan alias**Error! Bookmark not defined.**

Gambar 8.16 Menentukan tipe Generate Signed Apk**Error! Bookmark not defined.**

Gambar 8.17 Menunggu file (.apk) terbentuk.....**Error! Bookmark not defined.**

Gambar 8.18 Pemberitahuan aplikasi telah sukses dibentuk**Error! Bookmark not defined.**

Gambar 8.19 Aplikasi telah berhasil terbentuk.....**Error! Bookmark not defined.**

DAFTAR TABEL

Tabel 2.1 Versi Perkembangan Android.....**Error! Bookmark not defined.**

1.1 Pendahuluan

Mata kuliah Database Technology ini memberikan pemahaman dan penguasaan mengenai konsep - konsep database, model data relasional, teknik pembentukan database dan normalisasi, penggunaan bahasa query (sql) untuk pencarian, pengurutan, penyaringan, penghapusan dan update data. Database merupakan mata kuliah dasar yang sangat penting dalam Teknik Informatika, dengan harapan lulusan dari jurusan dapat mendesain dan membangun database untuk kepentingan pembangunan perangkat lunak. Proses pembelajaran mata kuliah Database Technology

Bahan Kajian

Pada bab ini berisi penjelasan secara komprehensif tentang pengantar database teknologi mulai dari konsep dasar basis data teknik pembentukan database hingga normalisasi data.

Sub Capaian

Sub capaian pembelajaran mata kuliah dalam bab ini adalah mahasiswa diharapkan dapat memahami konsep dasar basis data, serta teknik pembentukan basis data dan teknik normalisasi data.

Indikator

Mahasiswa dapat memahami perkembangan teknologi mobile, mahasiswa mampu memahami konsep dasar basis data Mampu mendesain database relasional dari hasil pemecahan masalah berdasarkan kasus yang diberikan baik secara mandiri maupun kerjasama tim.

1.2 Pengenalan Basis Data

Data merupakan sesuatu hal yang sangat penting dalam kehidupan manusia. Hal ini tidak dapat dipungkiri, karena setiap harinya kita selalu memerlukan dan menggunakan data dalam merencanakan segala sesuatu, mempertimbangkan hal apapun, dan mengambil keputusan dalam kehidupan kita. Sebagai contoh, Mahasiswa pastinya memerlukan data antara lain nilai tugas, nilai ujian tengah semester, nilai ujian akhir semester, nilai indeks prestasi sementara, nilai indeks prestasi kumulatif, biaya kuliah, jadwal kuliah serta data-data yang lainnya yang berkaitan dengan kegiatan perkuliahan.

Sekarang, dapatkah kita bayangkan kebutuhan suatu perusahaan atau organisasi terhadap data yang diperlukan bagi kepentingan bisnisnya? Saat ini, baik perusahaan kecil sampai perusahaan besar, pada umumnya telah memiliki informasi dan aplikasi, yang sangat memerlukan data. Artinya, antara sistem informasi dengan data memiliki hubungan yang sangat erat dan tidak dapat dipisahkan satu sama lainnya. Hal yang penting yang perlu diperhatikan adalah jika kita berbicara mengenai data, maka akan muncul istilah database atau basis data.

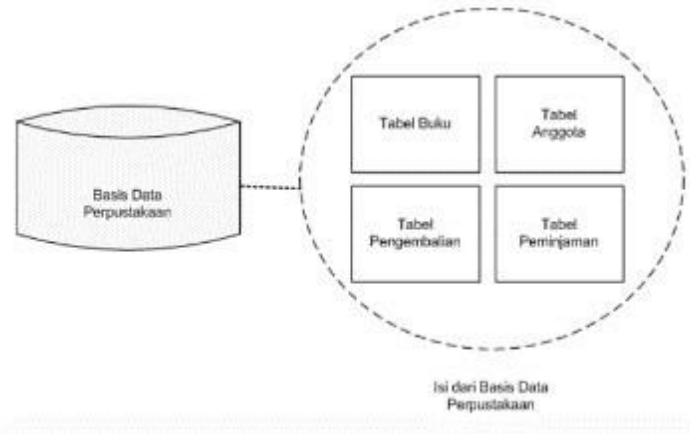
Basis data (database) dapat dibayangkan sebagai sebuah almari arsip. Jika kita memiliki sebuah lemari arsip dan bertugas untuk mengelolanya, maka kemungkinan besar kita akan melakukan hal-hal seperti : memberi map pada kumpulan arsip, memberi penomoran dengan pola tertentu yang nilainya unik pada setiap map, lalu menempatkan arsip-arsip tersebut dengan urutan tertentu didalam lemari. Kalaupun hal-hal tersebut tidak seluruhnya dilakukan, paling tidak, semua lemari arsip menerapkan suatu aturan tertentu bagaimana keseluruhan arsip-arsip

tadi disusun. Yang paling sederhana, tentu menyusun arsip-arsip tadi sesuai kronologisnya dan tanpa pengelompokkan. Hampir tidak akan pernah kita jumpai adanya lemari arsip yang tidak memiliki aturan dalam penyusunan arsip-arsip didalamnya.

Mengapa hal-hal itu kita lakukan? Jawabannya sederhana : kita berharap agar pada suatu saat nanti, sewaktu kita bermaksud untuk mencari dan mengambil arsip dari lemari maka kita akan dapat melakukannya dengan mudah dan cepat (Fathansyah, 2012). Basis Data (Database), pada saat ini sangat berdampak besar pada perkembangan ekonomi dan masyarakat. Sistem basis data berkaitan penting dalam pengembangan bidang rekayasa perangkat lunak, dan database menjadi kerangka kerja yang mendasari sistem informasi dan secara mendasar merubah cara banyak organisasi beroperasi. Contoh Penggunaan Basis Data pada aplikasi: aplikasi pengelolaan nomor telepon, aplikasi pembayaran gaji perusahaan, dll.

Berikut contoh penggunaan basis data : peminjaman pada perpustakaan : Ketika kita melakukan peminjaman di perpustakaan, kemungkinan besar basis data diakses. Petugas akan memasukkan kode buku atau menggunakan mesin pembaca, mesin ini dihubungkan dengan aplikasi database barang untuk mengetahui data buku tersebut. Aplikasi itu kemudian akan mengurangi jumlah stok buku tersebut dan menampilkan jumlah stok yang ada kepada petugas. Jika jumlah stok buku yang ada sudah di bawah ambang batas bawah stok, maka sistem database akan secara otomatis menginformasikan kepada petugas bahwa peminjaman sudah tidak bisa dilakukan. Atau, jika pembaca menanyakan ketersediaan sebuah buku , maka

petugas bisa melakukan pemeriksaan stok buku dan lokasi penyimpanan buku, dengan menjalankan aplikasi yang menentukan ketersediaan buku dari basis data.



Gambar I.1

Contoh Penggunaan Basis Data – Peminjaman Buku Pada Perpustakaan

Gambar diatas merupakan file-file yang disimpan dalam basis data pada harddisk.

Basis data Perpustakaan berisi beberapa file atau tabel seperti Tabel Buku,

Tabel anggota, Tabel Peminjaman dan Tabel Pengembalian.

1.3 Konsep Dasar Basis Data

1. Definisi Basis Data (Database)

Menurut (Fathansyah, 2012) menyampaikan bahwa :

Basis data terdiri dari dua kata , yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya yang diwujudkan dalam bentuk angka, huruf, symbol, teks, gambar, bunyi atau kombinasinya.

Sedangkan menurut (Indrajani, 2009) menyampaikan bahwa, ada beberapa

definisi tentang data, antara lain :

- a. Data adalah fakta atau observasi mentah yang biasanya mengenai fenomena fisik atau transaksi bisnis.
- b. Lebih khusus lagi, data adalah ukuran objektif dari atribut (karakteristik) dari entitas seperti orang, tempat, benda atau kejadian.
- c. Representasi fakta yang mewakili suatu objek seperti pelanggan, karyawan, mahasiswa, dan lain-lain, yang disimpan dalam bentuk angka, huruf, symbol, teks, gambar, bunyi dan kombinasinya.

Menurut (Fathansyah, 2012), sebagai satu kesatuan istilah, Basis Data (database) dapat didefinisikan dalam sejumlah sudut pandang seperti :

- a. Himpunan kelompok data(arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2. Prinsip dan Tujuan Basis Data

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip.

Perbedaannya hanya terletak pada media penyimpanan yang digunakan, jika lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan

media penyimpanan elektronik seperti cakram magnetis (magnetic disk) (Fathansyah, 2012).

Hal ini merupakan konsekuensi yang logis, karena lemari arsip langsung dikelola oleh manusia, sementara basis data dikelola melalui mesin pintar elektronik (komputer). Yang sangat ditonjolkan dalam basis data adalah pengaturan, pemilahan, pengelompokkan, pengorganisasian data yang akan disimpan sesuai dengan fungsi atau jenisnya. Pemilahan, pengelompokkan, pengorganisasian ini dapat berbentuk sejumlah tabel terpisah atau dalam bentuk pendefinisian kolom-kolom (field) dalam setiap tabel. Seperti contoh Gambar I.1, Basis data Perpustakaan berisi beberapa tabel didalamnya, yaitu tabel Anggota, tabel Buku, tabel Peminjaman dan tabel Pengembalian.

1.4 Operasi Dasar Basis Data

Di dalam sebuah disk, kita dapat menempatkan beberapa (lebih dari satu) basis data. Sementara dalam sebuah basis data, kita dapat menempatkan satu atau lebih tabel. Pada tabel ini sesungguhnya data disimpan dan ditempatkan. Setiap basis data umumnya dibuat untuk mewakili sebuah semesta data yang spesifik. Misalnya ada basis data kepegawaian, basis data akademik, basis data inventori, dan sebagainya.

Karena itu, operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data menurut (Fathansyah, 2012) dapat meliputi:

1. Pembuatan basis data baru (create database), yang identik dengan pembuatan lemari arsip yang baru

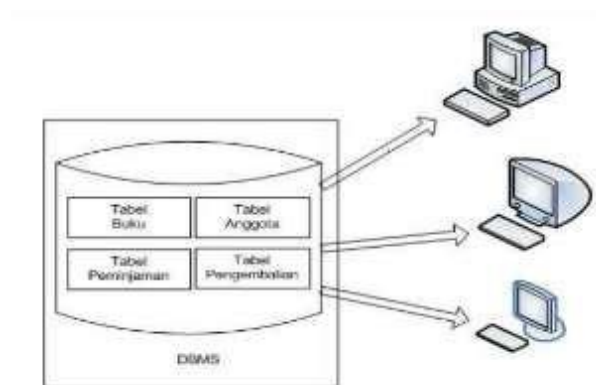
2. Penghapusan basis data (drop database), yang identik dengan perusakan lemari arsip (sekaligus berserta isinya, jika ada)
3. Pembuatan tabel baru ke suatu basis data (create table), yang identik dengan penambahan map arsip ke sebuah lemari arsip yang telah ada
4. Penghapusan tabel dari suatu basis data (drop table), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip
5. Penambahan atau pengisian data baru ke sebuah tabel di sebuah basis data (insert), yang identik dengan penambahan lembaran arsip ke sebuah map arsip
6. Pengambilan data dari sebuah tabel (query), yang identik dengan pencarian lembaran arsip dari sebuah map arsip
7. Pengubahan data dari sebuah tabel (update), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip
8. Penghapusan data dari sebuah tabel (delete), yang identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip

1.5 Sistem basis Data

Menurut (Fathansyah, 2012), sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu. Sebuah kendaraan dapat mewakili sebuah sistem yang terdiri atas komponen pemantik/starter (untuk memulai pengapian), komponen pengapian (untuk pembakaran BBM yang membuat torak bekerja), komponen penggerak/torak (untuk menggerakkan roda), komponen pengereman (untuk

memperlambat dan menghentikan gerakan torak dan roda), komponen pelistrikan (untuk mengaktifkan speedometer, lampu, dan lain-lain) yang secara bersama-sama melaksanakan fungsi kendaraan secara umum, yakni sebagai sarana transportasi.

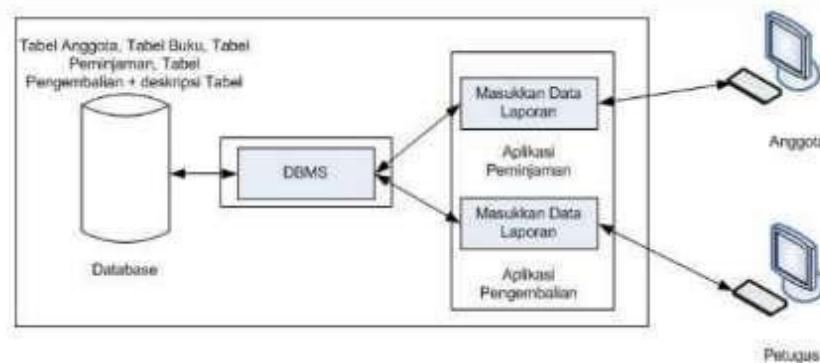
Basis data hanyalah sebuah objek yang pasif. Ia ada karena ada pembuatnya. Ia tidak akan pernah berguna jika tidak ada penegelolanya atau penggeraknya. Yang menjadi pengelola atau penggeraknya secara langsung adalah program/aplikasi (software). Gabungan keduanya (basis data dan pengelolaannya) menghasilkan sebuah sistem. Karena itu, secara umum menurut (Fathansyah, 2012), Sistem Basis Data merupakan sistem yang terdiri atas kumpulan tabel data yang saling berhubungan (dalam sebuah basis data di sebuah sistem computer) dan sekumpulan program (yang biasa disebut dengan DBMS (Database Management System) yang memungkinkan beberapa pemakai dan atau program lain untuk mengakses dan memanipulasi tabel-tabel tersebut.



Gambar I.2
1.6 Sistem Basis Data

1.7 Data Base Management System (DBMS)

DBMS adalah perangkat lunak yang memungkinkan pemakai untuk mendefinisikan, mengelola, dan mengontrol akses ke basis data. DBMS yang mengelola basis data relational disebut dengan Relational DBMS (RDBMS) Contoh perangkat lunak yang termasuk DBMS: dBase, FoxBase, Rbase, Microsoft-Access, Borland Paradox / Borland Interbase, MS-SQL Server, Oracle, Informix, Sybase, MySQL, dll.



Gambar I.3
Contoh Komputer Mengakses Database

Penjelasan Gambar diatas menunjukkan bagaimana sebuah komputer mengakses sebuah database : Database menampung semua data, dimulai dari anggota, data buku sampai dengan data transaksi peminjaman dan pengembalian, Sehingga anggota dapat melihat data buku yang tersedia dalam perpustakaan begitu pula anggota dapat melihat buku apa saja yang sudah dipinjam dan waktu pengembaliannya. Begitu pula dengan petugas dapat melihat fungsi yang sama pula.

1.8 Komponen Sistem Basis Data

Menurut (Fathansyah, 2012), dalam sebuah basis data, secara lengkap akan terdapat komponen-komponen utama sebagai berikut:

1. Perangkat Keras (Hardware)

Perangkat keras yang biasanya terdapat dalam sebuah sistem basis data adalah

- a. Komputer (satu untuk sistem stand alone atau lebih dari satu untuk sistem jaringan)
- b. Memori sekunder yang on line (Harddisk)
- c. Memori sekunder yang off line (Tape atau removable disk) untuk keperluan backup data
- d. Media/perangkat komunikasi (untuk sistem jaringan)

2. Sistem Operasi (Operating System)

Sistem operasi merupakan program yang mengaktifkan sistem komputer, mengendalikan seluruh sumber daya (resource) dalam komputer dan melakukan operasi-operasi dasar dalam komputer (operasi I/O, pengelolaan file, dan lain-lain). Sejumlah sistem operasi yang banyak digunakan seperti : MS-DOS, MS-Windows, Linux (untuk komputer stand alone atau komputer client dalam sistem jaringan) atau Novel-Netware, MS-Windows server, Unix,

Linux (untuk komputer server dalam sistem jaringan komputer)

3. Basis Data (Database)

Sebuah sistem basis data dapat memiliki beberapa basis data. Setiap basis data berisi sejumlah objek basis data (seperti tabel, indeks, dan lain-lain).

4. Sistem (Aplikasi/Perangkat Lunak) Pengelola Basis Data (DBMS)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus. Perangkat lunak inilah (disebut DBMS) yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. Ia juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pemaksaan keakuratan/konsistensi data, dan sebagainya.

5. Pemakai (User)

Ada beberapa jenis atau tipe pemakai terhadap suatu sistem basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem :

a. Programmer Aplikasi

Pemakai yang berinteraksi dengan basis data melalui Data Manipulation Language (DML), yang disertakan (embedded) dalam program yang ditulis dalam bahasa pemrograman induk (seperti C, C++, Pascal, PHP, Java, dan lain-lain).

b. User Mahir (Casual User)

Pemakai yang berinteraksi dengan sistem tanpa menulis modul program. Mereka menyatakan query (untuk akses data) dengan bahasa query yang telah disediakan oleh DBMS.

c. User Umum (End User/Naïve user)

Pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan satu program aplikasi permanen (executable program) yang telah disediakan sebelumnya.

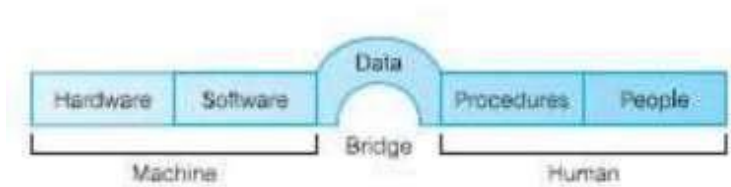
d. User Khusus (Specialized User)

Pemakai yang menulis aplikasi basis data non konvensional, tetapi untuk keperluan-keperluan khusus, seperti aplikasi Artificial Intelligence, Sistem Pakar, Pengolahan citra, dan lain-lain, yang bias saja mengakses basis data dengan atau tanpa DBMS yang bersangkutan.

6. Aplikasi (Perangkat Lunak) lain

Aplikasi (perangkat lunak) ini bersifat optional. Artinya, ada atau tidaknya tergantung pada kebutuhan kita.

1.9 Komponen Sebuah DBMS



Gambar I.4
Komponen DBMS

Menurut (Ladjamudin, 2004), kelima komponen tersebut dapat diklasifikasikan menjadi :

- Hardware dan Software yang berfungsi sebagai mesin.
- People dan Procedure yang merupakan manusia dan tatacara menggunakan mesin.

- c. Data merupakan jembatan penghubung antara manusia dan mesin agar terjadi suatu proses pengolahan data.

1. Hardware (Perangkat Keras)

DBMS dan software aplikasinya membutuhkan perangkat keras (hardware) untuk bias bekerja dan berfungsi dengan baik. Perangkat keras tersebut dapat berupa Personal Computer (PC) yang stand alone sampai kepada suatu mainframe yang terkoneksi dalam suatu jaringan. Perangkat keras umumnya akan menyesuaikan diri dengan kebutuhan suatu organisasi dan DBMS. Beberapa DBMS hanya bias dijalankan pada beberapa perangkat keras dan sistem operasi, sementara lainnya dapat digunakan secara umum oleh perangkat keras dan sistem operasi apa saja.

2. Software (Perangkat Lunak)

Komponen perangkat lunak terdiri dari perangkat lunak DBMS itu sendiri bersama dengan sistem operasi, juga termasuk didalamnya adalah perangkat lunak jaringan. Jika DBMS tersebut digunakan pada suatu jaringan dan program-program aplikasi. Bahasa pemrograman/program aplikasi dapat dikategorikan sebagai berikut :

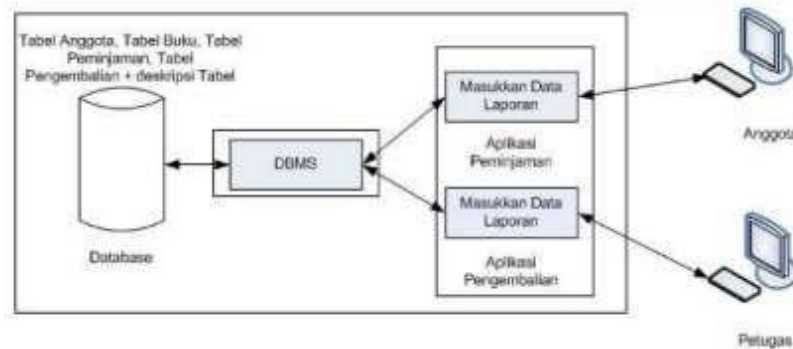
- a. Bahasa pemrograman generasi ketiga, seperti C++, Fortran, PASCAL, dan lain-lain.
- b. Bahasa pemrograman generasi keempat, seperti SQL, Oracle, Sybase, LISP, Informaix, dan lain-lain.

3. Data

Data merupakan komponen yang sangat penting dalam sebuah DBMS. Basis

Data terdiri dari data-data operasional dan meta data (data mengenai data). Struktur basis data dinamakan skema. Data didalam basis data mempunyai sifat

terpadu (integrated) dan berbagi (shared). Terpadu berarti bahwa berkas-berkas data yang ada pada basis data saling terkait, tetapi kemubaziran data tidak akan terjadi atau hanya terjadi sedikit sekali. Sedangkan berbagi data berarti bahwa data dapat dipakai oleh sejumlah pengguna. Artinya sesuatu data dapat diakses oleh sejumlah pengguna dalam waktu bersamaan.



Gambar I.5
Contoh Komputer Mengakses Basis Data

Skema :

- a. Tabel Buku (kode buku, judul buku, dst)
 - b. Tabel Anggota (kode anggota, nama anggota, dst)
 - c. Tabel Peminjaman (Kode pinjam, tgl pinjam, dst)
 - d. Tabel Pengembalian (kode kembali, tgl kembali, dst)
4. Procedure (Prosedur)

Prosedur menghubungkan berbagai perintah, dan aturan-aturan yang akan menentukan rancangan dan penggunaan basis data. User

(pengguna/pemakai) dari sistem dan staff yang akan mengatur dan merancang basis data berdasarkan prosedur-prosedur yang didokumentasikan. Dokumen tersebut berisi tentang bagaimana cara menggunakan dan menjalankan suatu sistem. Instruksi-

instruksi ini dapat terjadi dari :

- a. Log on ke DBMS.
- b. Gunakan fasilitas-fasilitas DBMS yang standar, atau program aplikasi.
- c. Memulai dan mengakhiri DBMS.
- d. Buatlah backup dari basis data tersebut.
- e. Tangani masalah-masalah hardware dan software.
- f. Ubahlah struktur tabel, dan lain-lain.

5. People (Pengguna/User)

People atau pengguna akan berinteraksi dengan mesin (software dan hardware) melalui berbagai prosedur dan aturan-aturan formal yang berlaku. People adalah seseorang yang menginginkan agar persoalannya dikerjakan oleh komputer. Pengguna dapat dikategorikan ke dalam empat golongan, yaitu :

a. Data Administrator (DA) dan Database Administrator (DBA)

1) Data Administrator (DA) akan bertanggungjawab dalam mengelola sumber daya data berupa:

- a) Perencanaan basis data
- b) Pemeliharaan dan peremajaan suatu standarisasi formal yang berlaku
- c) Menentukan kebijakan-kebijakan, prosedur formal, merancang

basis data logic.

Data administrator akan berkonsultasi dengan Manajer Senior, untuk meyakinkan bahwa alur pengembangan basis data akan sangat mendukung dan sejalan dengan target dari enterprise/perusahaan/organisasi tersebut.

2) Database Administrator (DBA) akan bertanggungjawab dalam mengelola sumber daya fisik dari sistem basis data berupa :

- a) Bertanggungjawab terhadap seluruh informasi yang berada didalam database.
- b) Bertanggungjawab terhadap strategi pengaksesan data dan mengorganisasikan file didalam media penyimpanan.
- c) Sebagai media penghubung/perantara dengan user.
- d) Memiliki otoritas pengecekan dan menjalankan prosedur validasi.
- e) Bertanggungjawab terhadap strategi backup dan pemeliharaan data.
- f) Mengontrol performansi data dan berhak memberi tanggapan atas usulan-usulan perubahan dan pemeliharaan data.

b. Database Designer

Dalam suatu proyek basis data yang besar dan kompleks kita dapat membagi database designer menjadi dua kelompok, yaitu:

1) Logical database designer (perancang basis data logic)

Logical database designer bertugas mengidentifikasi data (seperti entitas dan atribut), kardinalitas relasi dari data tersebut dan bagaimana data tersebut disimpan didalam media penyimpanan

sekunder. Logical database designer harus memiliki pemahaman yang menyeluruh

tentang organisasi data dan aturan-aturan bisnis.

2) Physical database designer (perancang basis data fisik)

Physical database designer mengambil model data logical dan memutuskan/menentukan bagaimana model data logic tersebut diimplementasikan ke dalam model data fisik. Pekerjaannya antara lain meliputi:

- a) Melakukan mapping terhadap model data logical ke dalam sekumpulan tabel atau relasi yang terintegrasi.
- b) Melakukan pemilihan atau menentukan struktur media penyimpanan, dan metode pengaksesan data agar dapat menghasilkan tampilan yang menarik dalam kegiatan-kegiatan basis data dan pengolahan data.
- c) Merancang dan menentukan standarisasi keamanan data.

c. Programmer Aplikasi (Application Programmer)

Ketika basis data telah diimplementasikan, programmer aplikasi yang akan membuat berbagai fungsi dan prosedur dalam sistem komputer yang akan digunakan oleh end user. Biasanya programmer aplikasi akan mengerjakan spesifikasi modul yang dirancang oleh sistem analis.

d. End User

1) Naïve User (User Umum)

Naïve user adalah mereka yang dapat mengenali DBMS. Mereka

mengakses basis data melalui program aplikasi yang telah tertulis secara khusus yang memungkinkan untuk dioperasikan sesederhana mungkin.

Mereka mengakses basis data dengan menginput atau memasukkan perintah-perintah sederhana dari sistem menu. Artinya, mereka tidak perlu mengetahui lebih banyak tentang basis data atau DBMS.

2) Sophisticated User (User Mahir)

Sophisticated User adalah mereka yang sangat familiar dengan struktur basis data dan fasilitas-fasilitas yang ditawarkan oleh DBMS. Sophisticated User dapat menggunakan bahasa query tingkat tinggi seperti SQL untuk menampilkan atau membuat operasi-operasi yang diinginkan. Beberapa Sophisticated User akan membuat sendiri program aplikasi yang mereka butuhkan.

1.10 Keuntungan dan Kekurangan DBMS

Keuntungan DBMS, sebagai berikut :

1. Pengontrolan kerangkapan data
2. Konsistensi data
3. Lebih banyak informasi dari jumlah data yang sama
4. Sharing data
5. Peningkatan integrasi data
6. Peningkatan keamanan
7. Penegakan standar layanan

Kekurangan DBMS, sebagai berikut :

1. Kompleksitas
2. Ukuran
3. Biaya DBMS
4. Biaya Perangkat keras tambahan
5. Biaya konversi teknologi
6. Performa
7. Dampak kegagalan yang lebih besar

Kesimpulan dari materi pertemuan ini adalah sebagai berikut:

- a. Data merupakan sesuatu hal yang sangat penting dalam kehidupan manusia. Hal ini tidak dapat dipungkiri, karena setiap harinya kita selalu memerlukan dan menggunakan data dalam merencanakan segala sesuatu, mempertimbangkan hal apapun, dan mengambil keputusan dalam kehidupan kita.
- b. DBMS adalah perangkat lunak yang memungkinkan pemakai untuk mendefinisikan, mengelola, dan mengontrol akses ke basis data.
- c. Tidak setiap sistem operasi mampu bertahan jika tidak mengembangkan kebutuhannya terhadap permintaan pasar.
- d. Setiap Sistem Operasi memiliki kelebihan dan kekurangan

Tugas 1

- | | |
|----|--|
| 1. | Buatlah rangkuman materi berupa makalah terkait basisdata dan pemanfaatannya pada saat ini baik dibidang pendidikan, industry, Perbankan dan kesehatan |
| 2. | Carilah hal-hal apa saja yang dibutuhkan dalam membuat sebuah basis data. |

2.1 Pendahuluan

Salah satu Model data relasional merupakan model basis data yang menggunakan fungsi tabel yang digunakan untuk menyimpan sebuah data. Model data relasional berbentuk sederhana sehingga mudah diakses oleh semua jenis user. Model data relasional juga mudah melakukan berbagai operasi data seperti query, update/edit, delete). Istilah-istilah model data relasional diantaranya Relasi, Atribut, Tuple, Candidate key, Model data berbasis objek merupakan data yang menganut konsep entitas, atribut dan hubungan antar entitas. Merupakan model data yang menjelaskan hubungan antar data dalam basis data berdasarkan persepsi bahwa dunia nyata terdiri dari objek-objek dasar yang saling memiliki hubungan atau relasi

Bahan Kajian

model basis data yang menggunakan fungsi tabel yang digunakan untuk menyimpan sebuah data.

Sub Capaian

Mahasiswa dapat memahami Model data relasional juga mudah melakukan berbagai operasi data seperti query, update/edit, delete). Istilah-istilah model data relasional diantaranya Relasi, Atribut, Tuple, Candidate key,

Indikator

Mahasiswa mengetahui dan melakukan berbagai operasi data seperti query, update/edit, delete).

2.2 Basis Data Relational

1. Pengertian Basis Data Relational

Basis data relasional adalah basis data yang mempresentasikan data dalam bentuk tabel-tabel, dimana tabel-tabel tersebut dihubungkan oleh nilai-nilai yang sama/umum pada kolom-kolom terkait. Menurut (Fathansyah, 2012), menyampaikan bahwa model basis data relasional sering pula disebut dengan model relasional atau basis data relasional. Model basis data ini diperkenalkan pertama kali oleh E.F. Codd. Model basis data menunjukkan suatu mekanisme yang digunakan untuk mengorganisasi data secara fisik dalam disk yang akan berdampak pula pada bagaimana kita mengelompokkan dan membentuk keseluruhan data yang berkaitan dalam sistem yang sedang kita tinjau.

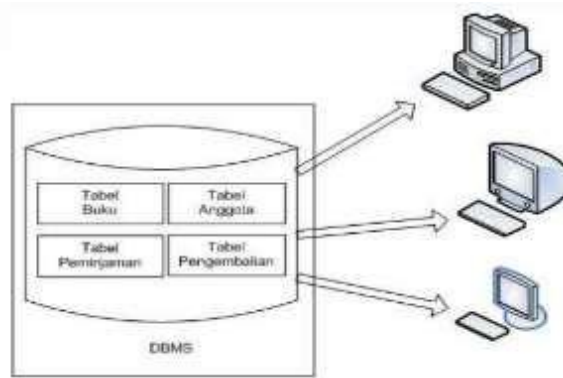
Sedangkan menurut (Hariyanto, 2004), mendefinisikan Basis data relasional merupakan kumpulan sejumlah berhingga relasi. Basis data relasional juga merupakan himpunan skema (metadata) dan himpunan instan skema (data-data).

2. Komponen Penyusun Basis Data

Untuk mengilustrasikan Basis data relasional secara lebih nyata, berikut ini kita ambil sebuah contoh basis data sederhana yang berkaitan dengan perpustakaan. Kita sebut saja basis data tersebut sebagai Basis Data Perpustakaan. Basis Data Perpustakaan terdiri dari empat tabel, yaitu tabel Anggota, tabel Buku, tabel

18

Peminjaman dan tabel Pengembalian. Basis data perpustakaan memiliki skema sebagai berikut :



Gambar 2.1
Skema Tabel

Tabel Anggota

Kode Anggota	Nama
A01	Surya
A02	Fitri
A03	Syahrur

Tabel Buku

Kode Buku	Judul	Stok Buku
B01	Pemrograman C++	10
B02	Membuat Aplikasi 30 Menit	15
B03	Cooking is Easy	15

Skema :

- a. Tabel Buku : kode buku (3) , judul buku (20)
- b. Tabel Anggota : kode anggota (3), nama anggota (25)
- c. Tabel Peminjaman : Kode pinjam (5), tgl pinjam (date), kode anggota (3)
- d. Tabel Pengembalian : Kode kembali (5), tgl kembali (date), kode anggota (3)

Adapun Komponen Penyusun Basis Data ada empat, sebagai berikut:

- a. Tabel

Tabel memiliki nama dan terdiri atas baris dan kolom. Tabel pada suatu basis data tidak boleh memiliki nama yang sama (unik). Tabel disebut juga dengan Relation atau File. Pada gambar diatas terdiri dari 4 tabel yaitu, tabel anggota, tabel buku, tabel peminjaman, tabel pengembalian. Menurut (Ladjamudin, 2004), tabel atau relasi memiliki karakteristik, sebagai berikut:

- 1) Nama relasi yang digunakan dalam suatu basis data haruslah berbeda satu dengan yang lainnya.
- 2) Masing-masing atribut suatu relasi terdiri dari simple attribute dan bernilai tunggal.
- 3) Masing-masing atribut dalam suatu relasi memiliki nama yang unik atau berbeda dengan lainnya.
- 4) Semua nilai dari suatu atribut haruslah berasal dari domain yang sama.
- 5) Tidak ada tuple yang ganda.
- 6) Tuple-tuple boleh tidak berurutan.
- 7) Atribut-atributnya tidak perlu berurutan.
- 8) Semua elemen data pada suatu kolom tertentu dalam relasi yang sama harus mempunyai jenis yang sama.

b. Kolom/Atribut

Kolom memiliki nama. Kolom yang terdapat dalam suatu tabel tidak boleh memiliki nama yang sama. Urutan nama boleh sembarang dan tidak mempengaruhi makna dari tabel. Nama lain kolom adalah Field

atau Atribut. Pada gambar diatas, contoh kolom pada tabel Buku yaitu kode buku dan judul buku.

c. Baris/Tuple

Berisikan data dari sebuah objek. Baris pada sebuah tabel harus unik, dapat diletakkan dalam urutan bebas dan tidak mempengaruhi makna dari tabel. Baris disebut juga dengan Record atau tuple. Pada slide diatas tabel anggota dapat menyimpan tiga obyek (yaitu tiga data anggota).

d. Domain

Domain adalah sekumpulan nilai-nilai yang dapat disimpan pada satu atau lebih kolom. Sebuah domain bisa dimiliki oleh satu kolom atau lebih, tetapi sebuah kolom hanya memiliki satu domain. Karena domain membatasi dan mengatur nilai yang dapat disimpan maka disebut domain constraint. Pada gambar diatas, kolom yaitu kode anggota hanya berisi 3 nilai saja, yaitu "A01".

3. Relational Keys

Relational Keys adalah identifikasi satu atau sekelompok kolom yang nilainya dapat membedakan secara unik tuple-tuple tersebut. Menurut (Ladjamudin, 2005), menyampaikan bahwa Key adalah elemen record yang dipakai untuk menemukan record tersebut pada waktu akses, atau bias juga digunakan untuk mengidentifikasi suatu entity atau record atau baris.

Tabel Anggota

Kode Anggota	Nama
A01	Surya
A02	Fitri
A03	Syahrur

Tabel Pengembalian

Kode kembali	Kode pinjam
KM01	PJ01
KM02	PJ02

Tabel Buku

Kode Buku	Judul	Stok Buku
B01	Pemrograman C++	10
B02	Membuat Aplikasi 30 Menit	15
B03	Cooking is Easy	15

Tabel Peminjaman

Kode pinjam	Tgl pinjam	Kode buku	Kode anggota	Juml	Tgl kembali
PJ01	10-01-2019	B01	A01	1	13-01-2019
PJ01	10-01-2019	B02	A01	1	13-01-2019
PJ01	10-01-2019	B03	A01	1	13-01-2019
PJ02	12-01-2019	B02	A02	1	14-01-2019
PJ02	12-01-2019	B03	A02	1	14-01-2019

Menurut (Pahlevi, 2013) terdapat 5 Relational Keys, sebagai berikut :

a. Superkey

Adalah satu atau kelompok kolom yang nilainya secara unik membedakan tuple-tuple pada suatu tabel. Pada gambar diatas di masing-masing tabel terdapat lebih dari satu superkey, yaitu :

- 1) Tabel anggota : kode anggota, nama anggota
- 2) Tabel buku : kode buku, judul, stok buku
- 3) Tabel peminjaman : kode pinjam, tgl pinjam, kode buku, kode anggota, jumml, tgl kembali
- 4) Tabel pengembalian : kode kembali, kode pinjam

Pada gambar diatas di masing-masing tabel terdapat lebih dri satu superkey, yaitu :

- 1) Tabel anggota :
 - a) Kolom kode anggota,
 - b) Kolom no faktur dan kolom nama anggota
- 2) Tabel buku :

- a) Kolom kode buku
- b) Kolom kombinasi kode buku, judul, stok buku Dst.

b. Candidate Key

Adalah superkey di mana tidak ada satupun himpunan bagian dari superkey tersebut menjadi superkey lagi. Tidak semua superkey menjadi candidate key. Candidate key yang terdiri dari dua kolom atau lebih disebut sebagai composite key. Pada gambar diatas masing-masing tabel terdapat lebih candidate key atau bukan candidate key, yaitu :

1) Tabel anggota :

- a) Kolom kode anggota : candidate key

Kolom kode anggota dan kolom nama anggota = bukan candidate key

- b) Tabel buku :

Kolom kode buku = candidate key

Kolom kombinasi kode buku, judul, stok buku : bukan candidate key

2) Tabel Peminjaman :

- a) Kolom kode pinjam, kode buku, kode anggota : candidate key

- b) Kolom kombinasi kode pinjam, kode buku, kode anggota, jumlah :

bukan candidate key

3) Tabel Buku :

- a) Kolom kode buku merupakan candidate key

b) Kolom kombinasi kode buku, judul buku, stok buku bukan candidate key

4) Tabel pengembalian

a) Kolom kode kembali, kode pinjam merupakan candidate key

b) Kolom kombinasi kode kembali, kode pinjam bukan candidate key

c. Primary Key

Adalah (satu) candidate key yang dipilih (di antara candidate key lain) untuk membedakan tuple-tuple secara unik dalam tabel. Jika dalam satu tabel hanya terdapat satu candidate key (misal tabel anggota dan tabel buku), maka key tersebut menjadi primary key. Tetapi jika terdapat lebih dari satu candidate key (misal tabel penjualan dan tabel pengembalian), maka salah satu candidate key tersebut dapat dijadikan primary key.

Primary key masing-masing tabel pada gambar di atas adalah

- 1) Tabel anggota : kode anggota
- 2) Tabel buku : kode buku
- 3) Tabel peminjaman : kode pinjam
- 4) Tabel pengembalian : kode kembali

d. Alternate Key

Adalah candidate key yang tidak dijadikan sebagai primary key. Misal pada tabel pengembalian jika kita memilih kode kembali sebagai primary key, maka kode pinjam dapat dijadikan alternate key.

e. Foreign Key

Adalah satu atau kelompok kolom yang nilainya sama atau terkait dengan candidate key pada tabel lain atau pada tabel yang sama. Misal pada tabel peminjaman ada kolom kode anggota yang terhubung dengan tabel anggota, maka kode anggota adalah foreign key. Kolom-kolom yang saling terkait ini sangat penting dalam operasi join. Pada tabel pengembalian ada kolom kode pinjam yang terhubung dengan tabel peminjaman, maka kode pinjam disini juga sebagai foreign key.

4. Skema tabel

Adalah informasi dasar yang mendeskripsikan tabel yang terdiri atas nama tabel dan sekumpulan pasangan kolom domain.

Contoh :

Skema Tabel Anggota (kode anggota, nama)

Skema Tabel Buku (kode buku, judul)

Skema Tabel Peminjaman (kode pinjam, tgl pinjam, tgl kembali, jumml, kode anggota, kode buku)

Skema Tabel Pengembalian (kode kembali, kode pinjam)

5. Skema Basis Data

Adalah sekumpulan skema tabel dengan masing-masing tabel memiliki nama yang berbeda.

Contoh : Skema Basis Data Perpustakaan : Tabel anggota (kode anggota, nama)

, Tabel buku (kode buku, judul, stok buku), Tabel peminjaman (kode pinjam, tgl pinjam, kode buku, tgl kembali, kode anggota, jumlah) dan Tabel Pengembalian (kode kembali, kode pinjam).

6. Integrity Constraint

Pada penjelasan di atas telah dibahas mengenai domain constraints. Terdapat empat constraints/batasan lain yang menjaga integritas data yang disimpan pada basis data :

a. Null

Adalah nilai pada suatu kolom (tuple) masih belum diketahui (unknown). Ini bisa berarti nilai tersebut tidak dapat diterapkan pada kolom tersebut. Namun, null tidak sama dengan nilai numerik nol atau string "-"; nol dan spasi adalah nilai, tetapi null menunjukkan tidak adanya nilai. Misal dalam sebuah tabel ada sebuah data yang belum diketahui boleh dituliskan null, akan tetapi hal tersebut tidak berlaku untuk primary key. Karena kolom primary key bersifat unik, jika primary key menyimpan null maka sifat unik dari kolom tersebut akan hilang karena bisa saja beberapa tuple memiliki nilai null.

b. Entity integrity

Adalah batasan atau aturan yang menyatakan bahwa kolom-kolom primary key tidak boleh menyimpan null. Seperti di jelaskan sebelumnya primary key digunakan untuk mendefinisikan secara unik sebuah tuple.

c. Referential integrity

Adalah batasan yang menyatakan jika suatu tabel memiliki kolom foreign key maka nilai pada foreign key tersebut harus sesuai dengan nilai kolom candidate key dan jika tidak demikian maka foreign key dapat dituliskan null. Dua keadaan penulisan null tidak perlu dilakukan :

- 1) Pada saat kolom tersebut diberikan batasan tidak boleh diberikan null.
- 2) Pada saat kolom tersebut juga merupakan bagian dari primary key.

d. General constraints

Adalah batasan /aturan tambahan yang ditetapkan oleh pemakai atau administrator basis data sesuai aturan/batasan yang ada pada suatu organisasi. Contoh :

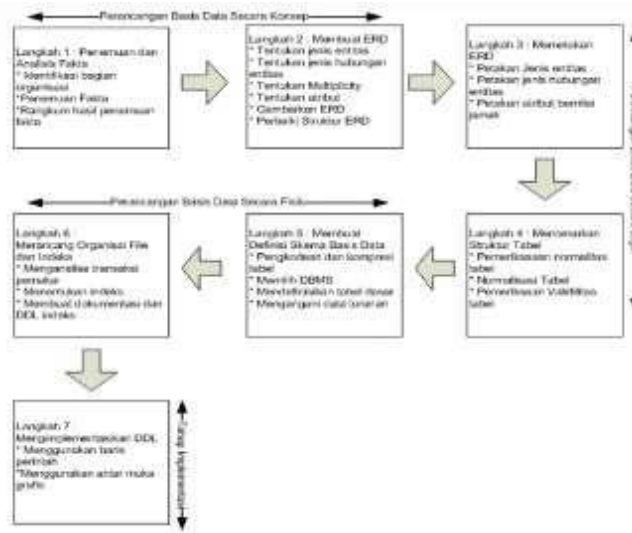
- 1) Peminjaman buku tidak diijinkan jika stok buku hanya satu
- 2) Jika anggota masih memiliki buku yang belum dikembalikan maka tidak di perbolehkan untuk meminjam kembali

2.3 Perancangan Basis Data

Proses pembangunan basis data terdiri dari dua tahapan utama :

1. Tahap analisis dan perancangan

Adalah tahapan pemetaan atau pembuatan model dari dunia nyata menggunakan notasi perancangan basis data tertentu serta pembuatan deskripsi implementasi basis data.



Gambar 2.2
Tahap Analisis dan Perancangan

Tahapan analisis dan perancangan dibagi menjadi tiga, yaitu :

a. Perancangan basis data secara konsep

Merupakan proses pembuatan data model dan tidak bergantung pada seluruh aspek fisik basis data.

b. Perancangan Basis Data Secara Logis

Merupakan proses pembuatan data model berdasarkan data model tertentu, tetapi tidak bergantung pada DBMS tertentu dan implementasi fisik basis data.

c. Perancangan Basis Data Secara Fisik

Merupakan proses pembuatan deskripsi implementasi basis data pada media penyimpanan sekunder (disk). Deskripsi ini menjelaskan tabel-tabel dasar, organisasi file, indeks untuk mendapatkan akses data secara efisien, dan semua integrity constraints, dan langkah-langkah keamanan.

2. Tahap Implementasi

Tahapan ini mengimplementasikan rancangan basis data yang telah dibuat. Implementasi menggunakan aplikasi klien yang disediakan oleh DBMS terpilih.

2.4 Studi Kasus

Perpustakaan Smart adalah perpustakaan umum yang anggotanya pelajar, mahasiswa dan masyarakat yang didirikan oleh Walikota Jakarta Barat. Keberadaan perpustakaan berlokasi di Walikota yang aplikasi pelayanan masih bersifat tradisional.

Prosesnya :

1. Setiap calon anggota yang akan menjadi anggota harus mengisi formulir dengan biaya administrasi Rp.10.000,-
2. Anggota dapat meminjam buku maksimal 3 buku
3. Untuk masa peminjaman selama 1 minggu (7 hari)
4. Keterlambatan pengembalian dikenakan denda sesuai dengan kondisi denda, diantaranya. Diantaranya :
 - a. Denda keterlambatan pengembalian dikenakan biaya administrasi Rp.500 perharinya (bukti surat denda terlampir)
 - b. Denda Buku perpustakaan rusak maka dikenakan biaya revisi buku perpustakaan(biaya ini dikenakan setelah buku diperbaiki).(bukti surat denda terlampir)
 - c. Denda Buku Hilang, maka dikenakan biaya penggantian seharga buku tersebut.(bukti surat denda terlampir)

- d. Perpustakaan smart dapat menerima sumbangan dari donatur statusnya
(anggota atau masyarakat luas)

Buat Database dan Tabel-tabelnya?

3.1 Pendahuluan

Pada bab ini kita akan membahas tentang database relasional, data disimpan dalam bentuk relasi atau tabel dua dimensi, dan antartabel satu dengan tabel yang lainnya terdapat hubungan atau relationship. Komponen-komponen model relasional meliputi : Kumpulan objek yang memiliki keterkaitan atau relasional antar penyimpan data, Set operator yang dapat melakukan relasi untuk membuat relasi yang lainnya, dan Integritas data untuk akurasi dan konsistensi. Untuk membuat struktur tabel, mengisi data ke tabel, mengubah data dan menghapus data dari tabel diperlukan software RDBMS, sedangkan perintah yang digunakan disebut Structure Query Language (SQL) sehingga setiap software RDBMS dapat digunakan untuk menjalankan perintah SQL

Bahan Kajian

Pada bab ini berisi penjelasan secara komprehensif tentang Sebuah database relasional menggunakan hubungan atau tabel dua dimensi untuk menyimpan informasi.

Sub Capaian

Sub capaian pembelajaran mata kuliah dalam bab ini yaitu mahasiswa dapat memahami cara membuat relasi antar table menggunakan hubungan atau tabel dua dimensi untuk menyimpan informasi

Indikator

Mahasiswa mengetahui konsep database relational serta merancang sebuah database.

3.2 Pengertian Model Data

Sebagaimana telah disebutkan di bab sebelumnya, upaya perancangan basis data dapat juga kita tempuh dengan secara langsung membuat sebuah model dari

awal sekali. Langkah ini biasanya ditempuh dari kelangkaan data/fakta yang kita miliki. Di sisi lain, sebuah model dinyatakan dalam bentuk diagram awal akan lebih mudah untuk dievaluasi/dianalisis untuk kemudian dilakukan perbaikan-perbaikan untuk mendapatkan sebuah model data yang lebih permanen dan lebih mendekati kenyataan sesungguhnya.

Menurut (Fathansyah, 2012), model data dapat didefinisikan sebagai kumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantic (makna) data dan batasan data. Oleh karena yang ingin ditunjukkan adalah makna dari data dan keterhubungannya dengan data lain, maka model data ini lebih tepat jika disebut Model Data Logik.

3.3 Jenis-jenis Model Data

a. Model Data Berdasarkan Object

Menurut (Ladjamudin, 2005), Model data berbasis objek menggunakan konsep entitas, atribut dan hubungan antar entitas (relationship). Entity adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama

41

dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu nama orang, benda, lokasi, kejadian (terdapat unsur waktu didalamnya), sedangkan atribut merupakan relasi fungsional dari satu objek set ke objek set yang lain. Relationship adalah hubungan alamiah yang terjadi antara entitas. Model data berbasis objek memiliki beberapa bentuk, sebagai berikut :

1) Model Keterhubungan Entitas (Entity-Relationship Model)

a) Pengertian Model Keterhubungan Entitas (Entity Relationship Model / ER Model)

Model Keterhubungan Entitas (Entity-Relationship Model) merupakan model yang paling populer digunakan dalam perancangan basis data. Entity-Relationship Model merupakan Model untuk menjelaskan hubungan antar data dalam basis data berdasarkan suatu persepsi bahwa real world terdiri dari objekobjek dasar yang mempunyai hubungan atau relasi antara objekobjek tersebut. Komponen utama pembentuk Model EntityRelationship, yaitu: Entitas (Entity) dan Relasi (Relation). Kedua komponen ini dideskripsikan lebih lanjut melalui sejumlah Atribut/Properti.



Dari tabel diatas kita dapat menentukan :

Entitas : Buku, Anggota, Peminjaman.





Atribut : Tabel Buku (kode buku, judul, stok buku), Tabel Anggota (kode anggota, nama) dan Tabel Peminjaman (kode pinjam, tgl pinjam, kode buku, kode anggota, jumlah, tgl

kembali). Relasi : hubungan antara kode buku di tabel buku dengan kode buku di tabel peminjaman. Begitu pula dengan kode anggota.

Model Entity Relationship yang berisi komponen himpunan entitas, relasi, yang dilengkapi atribut-atribut, dapat digambarkan menggunakan Diagram Entity Relationship (Diagram E-R).

b) Simbol dasar yang digunakan :

Tabel III.1
Simbol Dasar Entity Relationship Diagram

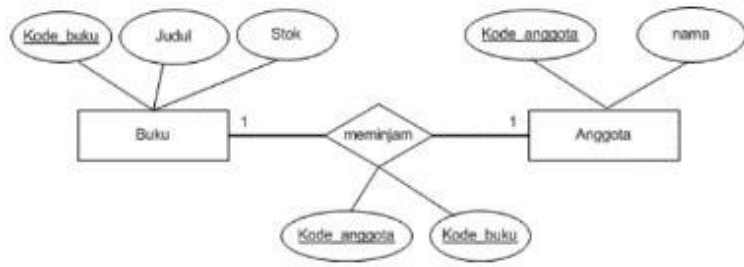
	Menyatakan Himpunan Entitas
	Menunjukkan Himpunan Relasi
	Menyatakan Atribut (Atribut key digaris bawah)
	Penghubung/Link

c) Mapping Kardinalitas

Dalam Diagram E-R aturan terpenting adalah Kardinalitas relasi/ Mapping Cardinalities yang menentukan jumlah entity yang dapat dikaitkan dengan entity lainnya melalui relationship-set.

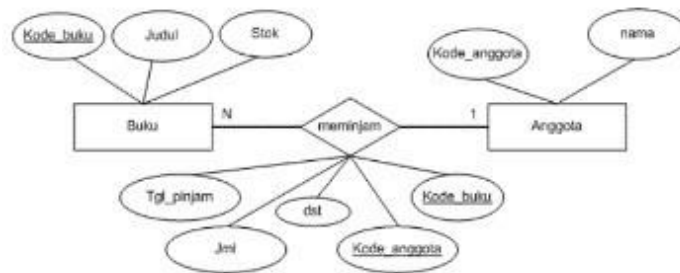
Jenis Mapping Cardinalities:

(1) Relasi satu ke satu (one-to-one) Contoh :



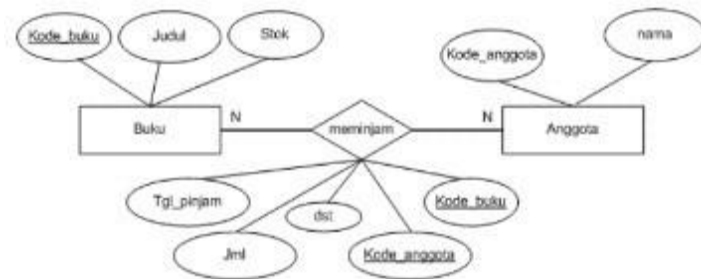
Gambar III.1
Relasi one to one

(2) Relasi satu ke banyak (one-to-Many)



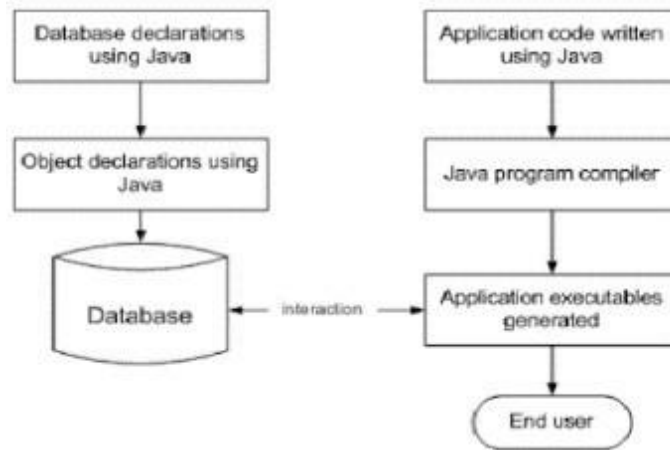
Gambar III.2
Relasi one to many

(3) Relasi banyak ke banyak (many-to-many)



Gambar III.3
Relasi many to many

2) Model Berorientasi Object (Object-Oriented Model)



Gambar III.4
Model Berorientasi Object

Penggambaran model berbasis objek menggunakan UML. UML Digambarkan dengan 2 Jenis, menurut (Sukamto & Shalahuddin, 2018):

a) Structural Diagram

(1) Class Diagram

Menggambarkan struktur sistem dari segi pendefinisian kelas yang akan dibuat untuk membangun sistem.

(2) Object Diagram

Menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem.

(3) Component Diagram

Dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.

(4) Deployment Diagram

Menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.

b) Behaviour Diagram

(1) Use case Diagram

Merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat.

(2) Sequence Diagram

Mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

(3) Communication Diagram

Penyederhanaan dari diagram kolaborasi (collaboration diagram). Collaboration diagram sudah tidak muncul pada UML versi 2.x.

(4) Statechart Diagram

Menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.



(5) Activity Diagram

Menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

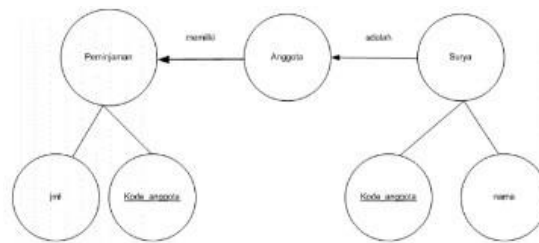
3.4 Model Data Semantik (Semantik Data Model)

Hampir sama dengan Entity Relationship model dimana relasi antara objek dasar tidak dinyatakan dengan simbol tetapi menggunakan katakata (Semantic). Sebagai contoh, dengan masih menggunakan relasi pada Bank X sebagaimana contoh sebelumnya, dalam semantic model adalah seperti terlihat pada gambar di atas. Tanda-tanda yang menggunakan dalam semantic model adalah:

Tabel III.2
Simbol Model Data Semantik

	Menunjukkan adanya relasi
	Menunjukkan atribut

Contoh :



Gambar III.5
Contoh Model Data Semantik

4) Model Data Fungsional (Functional Data Model)

b. Model Data Berdasarkan Record

Model ini berdasarkan pada record untuk menjelaskan kepada user tentang hubungan logic antar data dalam basis data.

Perbedaan dengan model data berbasis objek adalah pada record based data model disamping digunakan untuk menguraikan struktur logika keseluruhan dari suatu database, juga digunakan untuk menguraikan implementasi dari sistem database (higher level description of implementation)

Jenis-Jenis Model Data Berbasis Record :

1) Model Relational

Dimana data serta hubungan antar data direpresentasikan oleh sejumlah tabel dan masing-masing tabel terdiri dari beberapa kolom yang namanya unique. Model ini berdasarkan notasi teori himpunan

(set theory), yaitu relation.

Contoh : database penjualan terdiri dari 3 tabel, sebagai berikut :

- (a) Tabel Supplier
- (b) Tabel Suku Cadang
- (c) Tabel Pengiriman



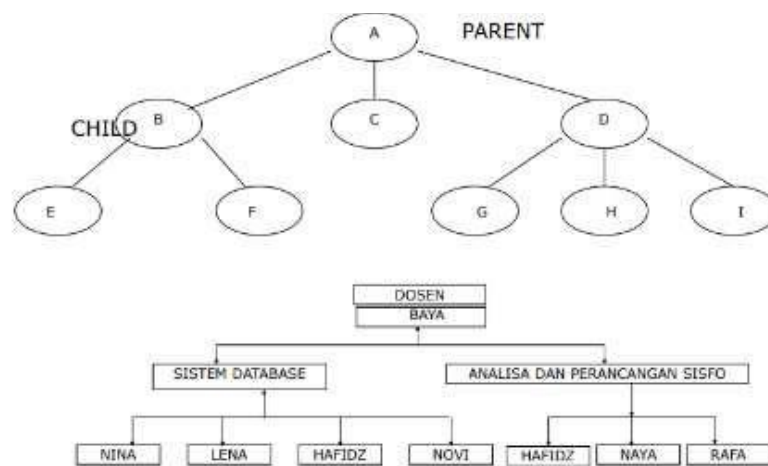
Gambar III.6
Contoh Model Data Relational

2) Model Hirarki

Dimana data serta hubungan antar data direpresentasikan dengan record dan link (pointer), dimana record-record tersebut disusun dalam bentuk tree (pohon), dan masing-masing node pada tree tersebut merupakan record/grup data elemen dan memiliki hubungan

cardinalitas 1:1 dan 1:M.

Contoh :



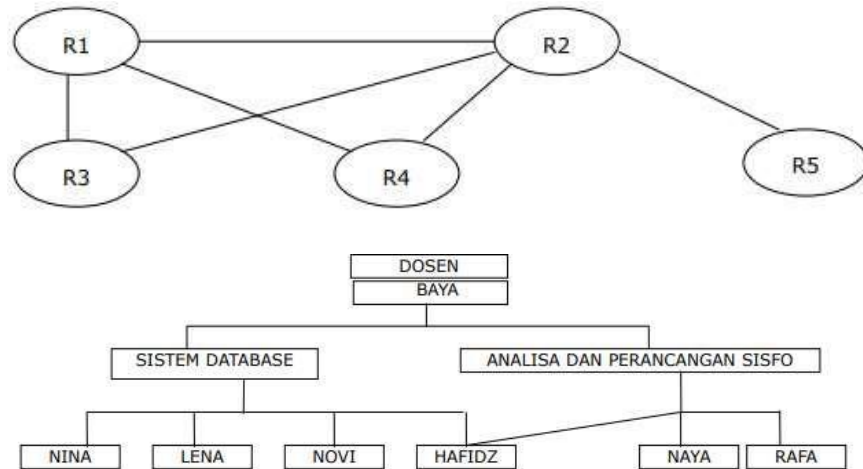
Gambar III.7

Contoh Model Data Hirarki

3) Model Jaringan

Distandarisasi tahun 1971 oleh Database Task Group (DBTG) atau disebut juga model CODASYL (Conference on Data System Language), mirip dengan hirarkical model dimana data dan hubungan antar data direpresentasikan dengan record dan links. Perbedaannya terletak pada susunan record dan linknya yaitu network model menyusun record-record dalam bentuk graph dan menyatakan hubungan cardinalitas 1:1, 1:M dan N:M.

Contoh :



Gambar III.8

Contoh Model Data Jaringan

Kesimpulan dari materi pertemuan ini adalah sebagai berikut:

- a. Data models adalah suatu set konsep yang digunakan untuk menggambarkan struktur dari database, menerangkan hubungan hubungan antar data serta batasan batasan yang sesuai dengan organisasi tersebut. Data models juga digunakan untuk operasi manipulasi struktur data dan mematuhi akan ketentuan dari database rule.

TUGAS

1.

1. Teori:

Buatlah rancangan sebuah database untuk menangani data-data kuliah. Data-data yang akan ditanganinya adalah: data pribadi mengenai mahasiswa, data pribadi mengenai dosen, data mata kuliah dan data ruang kelas. Mahasiswa boleh mengambil lebih dari satu mata kuliah, dan satu mata kuliah boleh diambil oleh lebih dari satu mahasiswa sekaligus (*joint account*).

Buatlah ER Diagram manual untuk kasus tersebut dari tahap 1 sampai tahap 4!

2. Praktek:

1. Ambil contoh sembarang database (harus berbeda untuk setiap mahasiswa). Buatlah rancangan ER Diagram manual database tersebut dari tahap 1 sampai tahap 4, dengan ketentuan database minimal mengandung 4 buah entitas.

4.1 Pendahuluan

Pada bab sebelumnya kita telah mengenal konsep dari layout dan user interface, pada bab ini kita akan mengetahui tentang kontrol input, jenis-jenis kontrol dan pengenalan tentang kontrol input dengan listener.

Bahan Kajian

Pada bab ini berisi penjelasan secara komprehensif tentang kontrol input, jenis-jenis kontrol, serta kita akan mencoba kontrol input dengan penambahan listener sehingga kontrol input dapat memiliki aksi tertentu. Selain itu juga kita akan mencoba kode program sederhana terkait aritmatika dan membuat custom listview.

Sub Capaian

Sub capaian pembelajaran mata kuliah dalam bab ini yaitu mahasiswa dapat memahami cara menggunakan kontrol input/widget view, mahasiswa mampu membuat dan mendesain menggunakan kontrol input yang disertai dengan listener.

Indikator

Mahasiswa mengetahui apa itu kontrol input, mahasiswa dapat mengenal jenis-jenis kontrol input dengan kode xml dan juga dengan desain, mahasiswa dapat membuat aplikasi aritmatika sederhana dan custom listview.

4.2 Pengertian Entity Relationship Diagram (ERD)

4.3 Komponen ERD

4.4 Kardinalitas / Derajat Relasi

4.5 Tahapan Pembuatan ERD

4.6 Logical Record Structured (LRS)

4.7 Membuat ERD

Kontrol

Kesimpulan dari materi pertemuan ini adalah sebagai berikut:

- a. Kontrol User Interface berperan penting dalam interaksi antara pengguna dengan konten aplikasi.
- b. Berbagai jenis-jenis kontrol seperti textview, edittext, button, checkbox dan lain sebagainya memiliki peran dan fungsi masing-masing dalam membentuk interface kontrol user interface.
- c. Untuk menjalankan suatu aksi kontrol input membutuhkan Listener seperti setOnClickListener.

TUGAS

1. **Teori:**
Membuat rangkuman kontrol input, jenis-jenis kontrol, dan listener. Uraikan bagaimana keterkaitan antara masing-masing.
2. **Praktek:**

Buatlah program kalkulator sederhana yang didalamnya terdapat beberapa operasi aritmatika yaitu penjumlahan, pengurangan, perkalian, pembagian dan modulus (sisa bagi).

5.1 Pendahuluan

Pada bab sebelumnya mahasiswa telah mengetahui bagaimana cara mendesain layout dan menggunakan kontrol input yang disertai dengan listener. Pada bab ini kita akan mengenal apa itu activity, bagaimana bentuk / pola dari rutinitas activity lifecycle di android.

Bahan Kajian

Pada bab ini berisi penjelasan secara komprehensif tentang apa itu activity, bagaimana activity dapat berinteraksi, bagaimana siklus hidup dari activity dari dibuat hingga dihancurkan, serta bagaimana penggunaan intent untuk berinteraksi dapat berinteraksi.

Sub Capaian

Sub capaian pembelajaran mata kuliah dalam bab ini yaitu mahasiswa dapat menggunakan activity dan intent, mahasiswa mampu mengimplementasikan activity dan intent pada software pengembangan aplikasi android studio, mahasiswa dapat memahami siklus hidup activity pada aplikasi android.

Indikator

Mahasiswa mengetahui activity, mahasiswa mengetahui bagaimana siklus activity dari dibuat hingga dihancurkan, mahasiswa mampu menggunakan explicit intent dan implicit intent pada activity.

5.2 Pengertian Normalisasi

5.3 Anormly

5.4 Atribut dan Ketergantungan Fungsi

5.5 Bentuk Normalisasi

Kesimpulan :

- a. Sadfasfas
- b. Asdfasdfsaf
- c. Sadfsadf

TUGAS

1. **Teori:**
Membuat rangkuman dari Activity, ActivityLifecycle, intent, intent implicit dan explicit.
2. **Praktek:**
Buatlah program dengan listview. Untuk setiap list item menampilkan menuActivity. Misalnya list pertama bertuliskan nasi goreng, list kedua ayam bakar, list ketiga soto dst.. kemudian dari list tersebut **ketika di klik maka akan berpindah ke Activity** kedua dan membawa pesan menggunakan Extras putExtra. Pesan tersebut ditampilkan pada sebuah TextView didalam Activity kedua tersebut.

6.1 Pendahuluan

Pada bab sebelumnya kita telah memahami bagaimana cara membuat aplikasi sederhana mulai dari layouting, hingga siklus hidup dari activity serta penggunaan intent. Pada bab ini kita akan mengenal bagaimana metode penyimpanan data di android. Mulai dari shared preference, sqlite, file storage, content providers, dan cloud storage. Selain itu bagaimana penyimpanan internal dan penyimpanan external. Dan kita juga akan mengenal database SQLite menggunakan DB Browser for SQLite.

Bahan Kajian

Pada bab ini berisi penjelasan secara komprehensif tentang penyimpanan data, penyimpanan data internal dan external, serta penggunaan database DB Browser for SQLite.

Sub Capaian

Sub capaian pembelajaran mata kuliah dalam bab ini yaitu mahasiswa dapat memahami dan menerapkan penyimpanan data, baik penyimpanan data internal dan external. Mahasiswa mampu memahami bagaimana penggunaan SQLite menggunakan DB Browser for SQLite.

Indikator

Mahasiswa dapat memahami konsep penyimpanan data, mahasiswa mampu menentukan penyimpanan data yang sesuai untuk aplikasi yang akan dibangun, mahasiswa memahami tentang shared preference, Sqlite, File Storage, Content Providers, dan cloud Storage.

6.2 Langkah-langkah pembuatan Normalisasi

6.3 Studi Kasus

Setiap aplikasi ccccc

Kesimpulan dari materi pertemuan ini adalah sebagai berikut:

- a. Penyimpanan data pada aplikasi mobile sangat penting agar aplikasi dapat bersifat dinamis dan ringan.
- b. Dalam Mobile Programming terdapat beberapa metode penyimpanan data yang dapat kita sesuaikan dengan spesifikasi kebutuhannya. Apakah kita harus menggunakan penyimpanan internal (untuk aplikasi personal) atukah perlu untuk akses penyimpanan external dan penggunaan internet dalam model penyimpanan data pada aplikasi mobile dan atukah perlu menggunakan kedua metode penyimpanan tersebut.
- c. Database SQLite merupakan database relasional yang relatif sangat ringat sehingga cocok digunakan dalam pembuatan aplikasi mobile.

TUGAS

1. Teori:

Temukan contoh aplikasi yang menggunakan media penyimpanan data, uraikanlah bagaimana metode penyimpanan data yang ada pada aplikasi tersebut. Anda boleh menjabarkan lebih dari satu model penyimpanan pada aplikasi tersebut jika ada!

2. Praktek:

Buatlah Layout Activity program yang memiliki penyimpanan Database. Tentukanlah rancangan databasenya, mulai dari berapa table, relasinya, hingga atribut apa saja yang ada pada penyimpanan database tersebut menggunakan SQLite.

7.1 Pendahuluan

Pada bab sebelumnya kita telah memahami bagaimana cara pembuatan aplikasi mulai dari rancangan layout, listener hingga database. Pada bab ini kita akan mengimplementasikan aplikasi mobile yang dilengkapi dengan database SQLite. Kita akan membangun aplikasi CRUD yaitu Create, Read, Update dan Delete. Untuk contoh pada kasus pembuatan aplikasi mobile kali ini kita membuat aplikasi untuk menyimpan catatan doa sehari-hari.

Bahan Kajian

Pada bab ini berisi studi kasus dengan menggabungkan pokok bahasan pada bab sebelumnya, mahasiswa akan dibimbing untuk membuat contoh aplikasi mobile CRUD sederhana dan mampu mengimplementasikannya.

Sub Capaian

Sub capaian pembelajaran mata kuliah dalam bab ini yaitu mahasiswa mampu memahami dan mengimplementasikan Create, Read, Update, Delete pada aplikasi mobile.

Indikator

Mahasiswa dapat memahami bagaimana konsep CRUD didalam pemrograman mobile, menggunakan SQLiteOpenHelper dan SQLiteDatabase. Mahasiswa mampu membuat aplikasi yang memiliki fitur Create, Read, Update, Delete.

7.2 Pengertian Bahasa Query Formal

7.3 Operator Aljabar Relational

Kali ini kita

Kesimpulan dari materi pertemuan ini adalah sebagai berikut:

- a. Membuat aplikasi mobile tidak begitu sulit, butuh ketekunan dan ketelitian.

- b. Dalam Aplikasi Mobile Android Studio kita membutuhkan beberapa Class yang saling terhubung untuk dapat saling berkaitan. Seperti untuk membuat recyclerview yang dinamis kita membutuhkan dbhelper, model, recyclerview, dan juga menambahkan beberapa dependencies sebagai pendukung dalam pembuatan aplikasi.
- c. Penggunaan database SQLite merupakan penyimpanan secara client, artinya bersifat internal aplikasi. Namun demikian database SQLite juga dapat dilakukan sinkronisasi ke server.

- | | |
|----|--|
| 1. | Teori:
1. Buatlah sebuah rancangan database aplikasi sederhana menggunakan sqlite. |
| 2. | Praktek:
Implementasikan rancangan database tersebut kedalam bentuk aplikasi CRUD. |

8.1 Pendahuluan

Pada bab sebelumnya kita telah memahami bagaimana membuat aplikasi dari desain layout, penggunaan kontrol input, database hingga penerapan aplikasi dengan CRUD. Pada bab ini kita akan memahami bagaimana cara merancang aplikasi android dengan metode-metode pengembangan sistem yang sering digunakan hingga mempublikasikan (publish) aplikasi (.apk) android.

Bahan Kajian

Pada bab ini kita akan mengetahui bagaimana metode-metode dalam pengembangan perangkat lunak, dan bagaimana cara mempublikasikan aplikasi android.

Sub Capaian

Sub capaian pembelajaran mata kuliah dalam bab ini yaitu mampu memahami metode pengembangan perangkat lunak, mahasiswa mampu menggambarkan UML dari aplikasi yang dirancang. Mahasiswa mampu mempublikasikan aplikasi android dalam bentuk file (.apk)

Indikator

Mahasiswa dapat memahami metode yang sesuai dengan model aplikasi yang akan dibangun, mahasiswa mampu menggambarkan uml dari aplikasi final project matakuliah yang dibangun, seperti use case diagram, activity diagram dan class diagram. Mahasiswa mampu mempublikasikan aplikasi (.apk) android.

8.2 Structured Query Language (SQL)

Kesimpulan dari materi pertemuan ini adalah sebagai berikut:

- a. Dalam pengembangan aplikasi mobile, perlu adanya perancangan aplikasi agar setiap rancangan dan skema yang dibangun dapat diaplikasikan dengan baik.

- b. Terdapat metode-metode pengembangan aplikasi yang dapat digunakan dalam membangun/mendevlop aplikasi.
- c. Ketika aplikasi kita sudah selesai kita dapat menguploadnya ke Google Playstore.

Tugas

- | | |
|----|--|
| 1. | Teori:
Buatlah rancangan lengkap aplikasi program mobile anda bersama dengan kelompok. Gunakan salah satu metode pengembangan yang anda kuasai. Buatlah juga dalam bentuk makalah dan ppt untuk dipresentasikan. |
| 2. | Praktek:
Buatlah program pada tugas teori Anda bersama kelompok. Submit bersama dengan makalah, ppt, dan hasil pembahasan. |

9.1 Pendahuluan

Pada bab sebelumnya kita telah memahami bagaimana membuat aplikasi dari desain layout, penggunaan kontrol input, database hingga penerapan aplikasi dengan CRUD. Pada bab ini kita akan memahami bagaimana cara merancang aplikasi android dengan metode-metode pengembangan sistem yang sering digunakan hingga mempublikasikan (publish) aplikasi (.apk) android.

Bahan Kajian

Pada bab ini kita akan mengetahui bagaimana metode-metode dalam pengembangan perangkat lunak, dan bagaimana cara mempublikasikan aplikasi android.

Sub Capaian

Sub capaian pembelajaran mata kuliah dalam bab ini yaitu mampu memahami metode pengembangan perangkat lunak, mahasiswa mampu menggambarkan UML dari aplikasi yang dirancang. Mahasiswa mampu mempublikasikan aplikasi android dalam bentuk file (.apk)

Indikator

Mahasiswa dapat memahami metode yang sesuai dengan model aplikasi yang akan dibangun, mahasiswa mampu menggambarkan uml dari aplikasi final project matakuliah yang dibangun, seperti use case diagram, activity diagram dan class diagram. Mahasiswa mampu mempublikasikan aplikasi (.apk) android.

9.2 Join

9.3 Fungsi Agregat

9.4 SubQuery

Kesimpulan dari materi pertemuan ini adalah sebagai berikut:

- d. Dalam pengembangan aplikasi mobile, perlu adanya perancangan aplikasi agar setiap rancangan dan skema yang dibangun dapat diaplikasikan dengan baik.
- e. Terdapat metode-metode pengembangan aplikasi yang dapat digunakan dalam membangun/mendevolop aplikasi.
- f. Ketika aplikasi kita sudah selesai kita dapat menguploadnya ke Google Playstore.

Tugas

- | | |
|----|--|
| 1. | Teori:
Buatlah rancangan lengkap aplikasi program mobile anda bersama dengan kelompok. Gunakan salah satu metode pengembangan yang anda kuasai. Buatlah juga dalam bentuk makalah dan ppt untuk dipresentasikan. |
| 2. | Praktek:
Buatlah program pada tugas teori Anda bersama kelompok. Submit bersama dengan makalah, ppt, dan hasil pembahasan. |

10.1 Pendahuluan

Pada bab sebelumnya kita telah memahami bagaimana membuat aplikasi dari desain layout, penggunaan kontrol input, database hingga penerapan aplikasi dengan CRUD. Pada bab ini kita akan memahami bagaimana cara merancang aplikasi android dengan metode-metode pengembangan sistem yang sering digunakan hingga mempublikasikan (publish) aplikasi (.apk) android.

Bahan Kajian

Pada bab ini kita akan mengetahui bagaimana metode-metode dalam pengembangan perangkat lunak, dan bagaimana cara mempublikasikan aplikasi android.

Sub Capaian

Sub capaian pembelajaran mata kuliah dalam bab ini yaitu mampu memahami metode pengembangan perangkat lunak, mahasiswa mampu menggambarkan UML dari aplikasi yang dirancang. Mahasiswa mampu mempublikasikan aplikasi android dalam bentuk file (.apk)

Indikator

Mahasiswa dapat memahami metode yang sesuai dengan model aplikasi yang akan dibangun, mahasiswa mampu menggambarkan uml dari aplikasi final project matakuliah yang dibangun, seperti use case diagram, activity diagram dan class diagram. Mahasiswa mampu mempublikasikan aplikasi (.apk) android.

10.2 Pengertian Basis Data Terdistribusi

10.3 Topologi Distribusi Data

10.4 Fragmentasi Data

11.1 Pendahuluan

Pada bab sebelumnya kita telah memahami bagaimana membuat aplikasi dari desain layout, penggunaan kontrol input, database hingga penerapan aplikasi dengan CRUD. Pada bab ini kita akan memahami bagaimana cara merancang aplikasi android dengan metode-metode pengembangan sistem yang sering digunakan hingga mempublikasikan (publish) aplikasi (.apk) android.

Bahan Kajian

Pada bab ini kita akan mengetahui bagaimana metode-metode dalam pengembangan perangkat lunak, dan bagaimana cara mempublikasikan aplikasi android.

Sub Capaian

Sub capaian pembelajaran mata kuliah dalam bab ini yaitu mampu memahami metode pengembangan perangkat lunak, mahasiswa mampu menggambarkan UML dari aplikasi yang dirancang. Mahasiswa mampu mempublikasikan aplikasi android dalam bentuk file (.apk)

Indikator

Mahasiswa dapat memahami metode yang sesuai dengan model aplikasi yang akan dibangun, mahasiswa mampu menggambarkan uml dari aplikasi final project matakuliah yang dibangun, seperti use case diagram, activity diagram dan class diagram. Mahasiswa mampu mempublikasikan aplikasi (.apk) android.

11.2 K

12.1 Pendahuluan

Pada bab sebelumnya kita telah memahami bagaimana membuat aplikasi dari desain layout, penggunaan kontrol input, database hingga penerapan aplikasi dengan CRUD. Pada bab ini kita akan memahami bagaimana cara merancang aplikasi android dengan metode-metode pengembangan sistem yang sering digunakan hingga mempublikasikan (publish) aplikasi (.apk) android.

Bahan Kajian

Pada bab ini kita akan mengetahui bagaimana metode-metode dalam pengembangan perangkat lunak, dan bagaimana cara mempublikasikan aplikasi android.

Sub Capaian

Sub capaian pembelajaran mata kuliah dalam bab ini yaitu mampu memahami metode pengembangan perangkat lunak, mahasiswa mampu menggambarkan UML dari aplikasi yang dirancang. Mahasiswa mampu mempublikasikan aplikasi android dalam bentuk file (.apk)

Indikator

Mahasiswa dapat memahami metode yang sesuai dengan model aplikasi yang akan dibangun, mahasiswa mampu menggambarkan uml dari aplikasi final project matakuliah yang dibangun, seperti use case diagram, activity diagram dan class diagram. Mahasiswa mampu mempublikasikan aplikasi (.apk) android.

12.2 Konkurensi

12.3 Locking

12.4 Timestamping

12.5 Crass dan Recovery

12.6 Security

12.7 Pemberian Wewenang dan View

12.8 Integrity

DAFTAR PUSTAKA

- [1] A. Developers, “Platform Architecture,” 2020. [Online]. Available: <https://developer.android.com/guide/platform>.
- [2] S. Liu, “Android operating system share worldwide by OS version from 2013 to 2020,” *statista*, 2020. .
- [3] Wikipedia, “Android version history,” 2020. [Online]. Available: https://en.wikipedia.org/wiki/Android_version_history.
- [4] Developer.android.com, “User Interface & Navigation,” 2020. [Online]. Available: <https://developer.android.com/guide/topics/ui?hl=en>.
- [5] Developer.android.com, “Android fundamentals 04.2: Input controls,” 2020.
- [6] Developer.android.com, “Intent,” 2020. [Online]. Available: <https://developer.android.com/reference/android/content/Intent>.
- [7] Developer.android.com, “Intent dan Filter Intent,” 2020. [Online]. Available: <https://developer.android.com/guide/components/intents-filters>.
- [8] Developer.android.com, “Data and file storage overview,” 2020.
- [9] Developer.android.com, “Save key-value data,” 2020. [Online]. Available: <https://developer.android.com/training/data-storage/shared-preferences>.
- [10] Developer.android.com, “Save data using SQLite,” 2020. [Online]. Available: <https://developer.android.com/training/data-storage/sqlite>.
- [11] Developer.android.com, “Data and file storage,” 2020. [Online]. Available: <https://developer.android.com/training/data-storage?hl=en>.
- [12] Developer.android.com, “Content providers,” 2020. [Online]. Available: <https://developer.android.com/guide/topics/providers/content-providers>.
- [13] Sqlitebrowser, “DB Browser for SQLite,” 2020. [Online]. Available: <https://sqlitebrowser.org/>.
- [14] Kevin Collens, “SQLiteOpenHelper dan SQLiteDatabase dalam Android Studio,” 2018. [Online]. Available: <https://socs.binus.ac.id/2018/08/08/sqliteopenhelper-dan-sqlitedatabase-dalam-android-studio/>.

- [15] Developer.android.com, “Publish your app,” 2020. [Online]. Available: <https://developer.android.com/studio/publish?hl=en>.