

PERANCANGAN BASIS DATA



Penulis :

**Dani Anggoro, Purwita Sari, Lucky Indra Kesuma, Yance Sonatha,
Muh Fahmi Rustan, Lut Faizal**

PERANCANGAN BASIS DATA

**Dani Anggoro
Purwita Sari
Lucky Indra Kesuma
Yance Sonatha
Muh Fahmi Rustan
Lut Faizal**



GET PRESS INDONESIA

PERANCANGAN BASIS DATA

Penulis :

Dani Anggoro
Purwita Sari
Lucky Indra Kesuma
Yance Sonatha
Muh Fahmi Rustan
Lut Faizal

ISBN : 978-623-125-243-2

Editor : Yuliatr Novita., M.Hum

Penyunting : Tri Putri Wahyuni., S.Pd

Desain Sampul dan Tata Letak : Atyka Trianisa, S.Pd

Penerbit : GET PRESS INDONESIA

Anggota IKAPI No. 033/SBA/2022

Redaksi :

Jln. Palarik Air Pacah No 26 Kel. Air Pacah
Kec. Koto Tangah Kota Padang Sumatera Barat

Website : www.getpress.co.id

Email : adm.getpress@gmail.com

Cetakan pertama, Juni 2024

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk dan
dengan cara apapun tanpa izin tertulis dari penerbit.

KATA PENGANTAR

Segala Puji dan syukur atas kehadiran Allah SWT dalam segala kesempatan. Sholawat beriring salam dan doa kita sampaikan kepada Nabi Muhammad SAW. Alhamdulillah atas Rahmat dan Karunia-Nya penulis telah menyelesaikan Buku Perancangan Basis Data ini.

Buku Ini Membahas Konsep Dasar, Istilah Dan Komponen Basis Data Serta Kegunaan System Basis Data, Konsep abstraksi data dan skema basis data; Pengantar tentang pemodelan data, basis data, Normalisasi, Proyek berupa kasus desain basis data dengan pendekatan normalisasi, Studi Kasus Dalam Perancangan Basis Data.

Proses penulisan buku ini berhasil diselesaikan atas kerjasama tim penulis. Demi kualitas yang lebih baik dan kepuasan para pembaca, saran dan masukan yang membangun dari pembaca sangat kami harapkan.

Penulis ucapkan terima kasih kepada semua pihak yang telah mendukung dalam penyelesaian buku ini. Terutama pihak yang telah membantu terbitnya buku ini dan telah mempercayakan mendorong, dan menginisiasi terbitnya buku ini. Semoga buku ini dapat bermanfaat bagi masyarakat Indonesia.

Padang, Juni 2024

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	v
DAFTAR TABEL	vi
BAB 1 KONSEP DASAR, ISTILAH DAN KOMPONEN	
BASIS DATA SERTA KEGUNAAN SISTEM BASIS DATA....	1
1.1 Pendahuluan.....	1
1.2 Istilah-istilah pada Basis Data.....	2
1.2.1 Tabel.....	2
1.2.2 Field	2
1.2.3 Record	2
1.2.4 Key.....	2
1.2.5 Query SQL.....	3
1.3 Komponen Basis Data	3
1.3.1 Perangkat Keras (<i>Hardware</i>).....	3
1.3.2 Sistem Operasi (<i>Operating System</i>)	4
1.3.3 Sistem Pengolahan Basis Data (DBMS).....	4
1.3.4 Data.....	5
1.3.5 Pengguna (<i>User</i>)	5
1.4 Kegunaan Sistem Basis Data	6
DAFTAR PUSTAKA	8
BAB 2 KONSEP ABSTRAKSI DATA DAN SKEMA	
BASIS DATA, PENGANTAR TENTANG	
PEMODELAN DATA.....	9
2.1 Konsep Abstraksi Data dan Skema Basis Data	10
2.2 Pemodelan Data.....	12
2.2.1 Model Data Berbasis Objek - OBDM (<i>Object</i>	
<i>Based Data Model</i>).....	13
2.2.2 Model Data Berbasis Rekaman-RBDM (<i>Record</i>	
<i>Based Data Model</i>).....	14
2.2.3 Model Data Berbasis Fisik-PBDM (<i>Physical</i>	
<i>Based Data Model</i>).....	16
DAFTAR PUSTAKA	17
BAB 3 LINGKUNGAN BASIS DATA	19
3.1 Pendahuluan.....	19

3.2 Kebutuhan Bisnis	20
3.3 Model Data	22
3.4 Sistem Manajemen Basis Data (DBMS)	24
3.5 Desain Skema Basis Data	24
3.6 Bahasa <i>Query</i>	26
3.7 Keamanan Basis Data	26
3.8 Backup dan Pemulihan	27
3.9 Pemeliharaan dan Pengelolaan	27
3.11 Skalabilitas	28
3.12 Integrasi dengan Aplikasi:	28
DAFTAR PUSTAKA	29
BAB 4 NORMALISASI	31
4.1 Konsep Dasar Normalisasi	31
4.1.1 Sejarah Singkat Normalisasi	31
4.1.2 Anomalisasi Data	32
4.1.3 Tujuan Normalisasi Data	32
4.2 Ketergantungan Fungsional	34
4.2.1 Definisi Ketergantungan Fungsional	34
4.2.2 Jenis-Jenis Ketergantungan Fungsional	34
4.3 First Normal Form (1NF)	35
4.4 Second Normal Form (2NF)	37
4.5 Third Normal Form (3NF)	39
4.6 Boyce-Codd Normal Form (BCNF)	40
4.7 <i>Fourth Normal Form</i> (4NF)	42
4.8 Fifth Normal Form (5NF)	43
DAFTAR PUSTAKA	45
BAB 5 PROYEK BERUPA KASUS DESAIN BASIS	
DATA DENGAN PENDEKETAN NORMALISASI	47
5.1 Pendahuluan	47
5.2 Proyek rental mobil	48
5.3 Perekaman Medis Klinik Kesehatan	51
5.4 Aplikasi perpustakaan	55
DAFTAR PUSTAKA	63
BAB 6 STUDI KASUS: BASIS DATA HOTEL	65
6.1 Membuat Database Hotel	65
6.2 Membuat Tabel Dengan Query SQL	67
6.2.1 Membuat Tabel Hotel	68

6.2.2 Membuat Tabel User.....	68
6.2.3 Membuat Tabel Fasilitas.....	70
6.2.4 Membuat Tabel Tipe Kamar.....	71
6.2.5 Membuat Tabel Kamar.....	72
6.2.6 Membuat Tabel Tamu.....	73
6.2.7 Membuat Tabel Check In.....	74
6.3 Membuat Relasi Antar Tabel.....	75
6.4 Menginput Data Dengan Query SQL.....	76
DAFTAR PUSTAKA	79
BIODATA PENULIS	

DAFTAR GAMBAR

Gambar 2.1. Tiga Level Abstraksi Data	10
Gambar 2.2. Contoh ERM di Bank X	13
Gambar 2.3. Contoh Semantical Model di Bank X	14
Gambar 2.4. Contoh Model Hierarki	15
Gambar 2.5. Model Jaringan.....	15
Gambar 4.1. Normalisasi 1NF untuk Pengecekan Nilai Atomik.....	36
Gambar 4.2. Normalisasi 1NF untuk Pengecekan Nama Kolom Secara Unik.....	36
Gambar 4.3. Normalisasi 1NF untuk Pengecekan Setiap Baris Secara Unik.....	37
Gambar 4.4. Normalisasi 2NF untuk Menghilangkan Ketergantungan Partial.....	38
Gambar 4.5. Normalisasi 3NF untuk Menghilangkan Ketergantungan Transitif.....	40
Gambar 4.6. Normalisasi BCNF	41
Gambar 6.1. Xampp Control Panel	65
Gambar 6.2. Dashboard phpMyAdmin	66
Gambar 6.3. List Database.....	67
Gambar 6.4. SQL Editor tbl_hotel.....	68
Gambar 5.6. Informasi Pembuatan tbl_hotel.....	68
Gambar 6.6. SQL Editor tbl_user.....	69
Gambar 6.7. Informasi Pembuatan tbl_user	69
Gambar 6.8. SQL Editor tbl_fasilitas	70
Gambar 6.9. Informasi Pembuatan tbl_fasilitas.....	70
Gambar 6.10. SQL Editor tbl_tipe_kamar	71
Gambar 6.11. Informasi Pembuatan tbl_tipe_kamar.....	71
Gambar 6.12. SQL Editor tbl_kamar	72
Gambar 6.13. Informasi Pembuatan tbl_kamar	73
Gambar 6.14. SQL Editor tbl_tamu.....	73
Gambar 6.15. Informasi Pembuatan tbl_tamu	74
Gambar 6.16. SQL Editor tbl_checkin	74
Gambar 7.17. Informasi Pembuatan tbl_checkin.....	75
Gambar 6.18. Menu Designer.....	75
Gambar 6.19. Relasi Antar Tabel.....	75

DAFTAR TABEL

Tabel 5.1. Tabel member yang memenuhi 2NF	48
Tabel 5.2. Tabel mobil yang memenuhi 2NF	48
Tabel 5.3. Tabel perusahaan yang memenuhi 3NF	49
Tabel 5.4. Tabel mobil yang memenuhi 3NF	49
Tabel 5.5. Tabel Rental.....	50
Tabel 5.6. Tabel Rental 1.....	50
Tabel 5.7. Tabel Rental 2.....	50
Tabel 5.8. Tabel dokter yang memenuhi 2NF	52
Tabel 5.9. Tabel pasien yang memenuhi 2NF.....	52
Tabel 5.10. Tabel obat yang memenuhi 2NF	53
Tabel 5.11. Tabel rekam medis yang memenuhi 2NF	53
Tabel 5.12. Tabel perusahaan obat	55
Tabel 5.13. Tabel Obat	55
Tabel 5.14. Contoh tabel perpustakaan (tidak normal).....	56
Tabel 5.15. Contoh tabel perpustakaan 1NF	57
Tabel 5.16. Tabel buku Hasil normalisasi 2NF	58
Tabel 5.17. Tabel mahasiswa Hasil normalisasi 2NF	58
Tabel 5.18. Tabel peminjaman buku Hasil normalisasi 2NF.....	58
Tabel 5.19. Tabel penerbit buku Hasil normalisasi 3NF	59
Tabel 5.20. Tabel penerbit buku Hasil normalisasi 3NF	60
Tabel 5.21. Tabel pinjam 1 Hasil normalisasi BCNF	61
Tabel 5.22. Tabel pinjam 2 Hasil normalisasi BCNF	62

BAB 1

KONSEP DASAR, ISTILAH DAN KOMPONEN BASIS DATA SERTA KEGUNAAN SISTEM BASIS DATA

Oleh Dani Anggoro

1.1 Pendahuluan

Sebelum membahas lebih jauh tentang basis data, sebaiknya kita mengenal dulu apa itu data dan informasi, karena terkadang kita sulit membedakan antara data dan informasi. Data adalah sebuah fakta mentah yang belum diolah (Carlos Coronel dan Steven Morris, 2016). Menurut KBBI data merupakan kumpulan dari fakta yang diperoleh dari hasil pengamatan, pengukuran ataupun penelitian. Data dapat berupa angka, teks atau gambar yang dapat diolah serta dianalisa guna mendapatkan pemahaman atau pengetahuan lebih lanjut. Sedangkan informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan dapat menjadi penunjang keputusan saat ini ataupun masa depan (Kelly, 2011).

Dapat disimpulkan bahwa perbedaan data dan informasi adalah: data adalah fakta yang belum diolah sedangkan informasi adalah kumpulan dari data yang telah diolah yang mampu dijadikan penunjang keputusan.

Informasi yang akurat, tepat dan cepat sangat penting dalam membantu membuat sebuah keputusan. Untuk menghasilkan informasi tersebut perlu adanya kumpulan data atau basis data. Basis data dalam istilah teknologi juga dikenal dengan nama database. Basis data berisikan data yang saling terhubung yang disimpan secara bersama-sama pada suatu media dengan struktur atau skema tertentu.

1.2 Istilah-istilah pada Basis Data

Dalam basis data terdapat istilah-istilah yang umum digunakan. Berikut dijelaskan beberapa istilah yang sering digunakan dalam basis data:

1.2.3 Tabel

Tabel adalah struktur dasar yang digunakan untuk menyimpan data. Tabel terdiri dari baris dan kolom. Baris digunakan untuk menampung record sementara kolom digunakan untuk penamaan *field*. Pada skema basis data penamaan tabel bersifat unik, penamaan tabel satu dengan tabel yang lain tidak boleh sama. Setiap tabel umumnya memiliki *key* untuk membantu pengelompokan atau pengaksesan *record*. Contoh tabel pada database: tb_pelanggan, tb_barang, tb_pesanan.

1.2.2 Field

Field adalah atribut dari suatu *record* yang menentukan suatu item dari sebuah data. Sama halnya dengan penamaan tabel, penamaan field pada suatu tabel tidak boleh sama atau harus unik. Contoh penamaan field pada tabel tb_pelanggan: id_pelanggan, nama_pelanggan, alamat, no_handphone.

1.2.3 Record

Record adalah rekaman dalam database. *Record* disebut juga dengan tuple, merupakan kumpulan dari element-element yang ada pada field yang saling berkaitan guna memberikan informasi mengenai entitas secara lengkap. Contoh record pada tabel tb_pelanggan: id_pelanggan (001, 002, 003), nama_pelanggan (Ferdinand, Agus, Lusi), alamat (Jakarta, Lampung, Bandung), no_telpon (0878xxx, 0813xxx, 0819xxxx).

1.2.4 Key

Key digunakan untuk mengidentifikasi field pada suatu tabel. Terdapat beberapa *key* yang ada di dalam basis data. Berikut beberapa contoh *key* yang sering digunakan:

1. *Primary Key* (Kunci Utama), biasa digunakan untuk mengidentifikasi *record* yang unik pada tabel. Setiap tabel biasanya hanya mempunyai satu *primary key*. Contoh pada

tabel *tb_pelanggan* *primary key* nya adalah *id_pelanggan*. *Id_pelanggan* dipilih karena setiap *record* pelanggan memiliki id yang tidak sama.

2. *Foreign Key* (Kunci Asing), digunakan untuk membuat relasi tabel yang berkaitan. Key ini biasanya merupakan *primary key* dari tabel lain.

1.2.5 Query SQL

Query SQL (Structured Query Language) adalah pernyataan atau perintah yang digunakan untuk berinteraksi dengan database. Fungsinya adalah untuk mengambil, menyisipkan, memperbarui, atau menghapus data dari tabel dalam database. *Query SQL* memungkinkan pengguna untuk mengelola data dengan melakukan operasi tertentu pada tabel, seperti pencarian data berdasarkan kriteria tertentu atau menggabungkan data dari beberapa tabel.

Contoh *query SQL* untuk mengambil record dari tabel *tb_pelanggan* dengan parameter *no_handphone*:

```
SELECT      id_pelanggan,      nama_pelanggan,      alamat,
no_handphone
FROM tb_pelanggan
WHERE no_handphone LIKE '%081%'
ORDER BY nama_pelanggan ASC;
```

1.3 Komponen Basis Data

Untuk menunjang kinerja basis data supaya efektif, diperlukan beberapa komponen yang harus dipenuhi yaitu (Khadir, 1998) :

1.3.1 Perangkat Keras (*Hardware*)

Perangkat keras adalah perangkat yang terlihat secara fisik. Perangkat ini biasanya terdiri dari perangkat masukan, perangkat keluaran, perangkat pengolahan data, perangkat komunikasi dan perangkat penyimpanan (Anggoro D., 2023). Untuk menunjang kegiatan pada basis data dibutuhkan beberapa perangkat keras, diantaranya :

1. *Central Processing Unit (CPU)*

CPU pada basis data bertugas untuk mengontrol operasi dan melakukan pemrosesan data. Kecepatan perwaktuan (*clock*) pada CPU mempengaruhi kecepatan pemrosesan basis data. Semakin tinggi *clock* pada CPU akan semakin baik. Begitu juga sebaliknya.

2. Harddisk dan RAM

Pada basis data, Harddisk dan RAM dibutuhkan sebagai penunjang penyimpanan data pada database. Hardisk menyimpan data secara permanen, sementara RAM menyimpan data sementara saat pengguna melakukan manipulasi database. Harddisk dengan kapasitas yang besar diperlukan jika data yang ada di databse jumlahnya besar. Begitu juga RAM, untuk melakukan pemrosesan data yang besar, RAM dengan kapasitas besar tentunya akan lebih baik. Besarnya kapasitas Harddisk dan RAM ditentukan oleh jumlah data yang akan ditampung di database yang digunakan.

3. Komponen Penunjang

Untuk memunjang proses basis data diperlukan perangkat keras penunjang seperti mouse dan keyboard untuk membantu prose memasukan data serta manipulasi data di database. Monitor untuk menampilkan data yang ada di database ke layar.

1.3.2 Sistem Operasi (*Operating System*)

Sistem operasi merupakan perangkat lunak utama pada sistem di komputer. Sistem operasi berfungsi sebagai pengelola seluruh sumber daya komputer. Sistem operasi sangat mempengaruhi kinerja sistem pengelolaan basis data (DBMS). Sistem Operasi harus menyesuaikan dengan penggunaan software basis data. Contoh Sistem Operasi yaitu: Windows, Linux, Unix dan lain-lain.

1.3.3 Sistem Pengolahan Basis Data (DBMS)

DBMS adalah perangkat lunak sebagai layanan permintaan pengguna yang posisinya berada diantara basis data yang ada di

harddisk dengan pengguna. DBMS digunakan sebagai aplikasi pengelola basis data. Umumnya DBMS menerapkan pengelolaan pengamanan data, penggunaan data dengan waktu yang bersamaan, konsistensi data dan sebagainya. Beberapa DBMS yang sering digunakan dalam kegiatan basis data adalah *MySQL*, *SQL Server*, *Microsoft Access*, *Oracle* dan lainnya.

1.3.4 Data

Data adalah salah satu komponen penting dalam basis data. Dalam basis data terdapat dua sifat data. Pertama, data yang disimpan secara terintegrasi. Pada sifat ini data akan disusun kemudian data yang rangkap akan dihapus. Kedua, data yang digunakan secara bersamaan. Pada sifat ini data dapat dibagikan dan diakses secara bersamaan melalui aplikasi yang berbeda.

1.3.5 Pengguna (User)

Pengguna adalah orang yang berkaitan langsung dengan basis data. Pada basis data pengguna dapat dibagi menjadi 3 bagian yaitu:

1. Database manajer, merupakan pengguna yang berperan dalam pengelolaan database. Database manajer berperan dalam menjaga keamanan data, melakukan pemulihan data, DBMS yang digunakan.
2. Database administrator (DBA), seseorang yang menjadi pusat pengendali data terhadap sistem basis data. DBA berperan dalam menentukan skema basis data, mengubah skema basis data, mengontrol dan memberikan hak akses data kepada pengguna, melakukan pemeliharaan harian seperti backup data serta melakukan pengawasan terhadap hal-hal yang mengenai penggunaan basis data.
3. Database User, merupakan seseorang yang menggunakan basis data hanya sebagai pengguna saja. Terdapat 4 golongan database user yaitu :
 - a. Programmer, programmer adalah seseorang yang membuat program aplikasi. Supaya aplikasi dapat terhubung dengan database, programmer harus diberikan akses ke dalam database. Biasanya

programmer memerlukan bantuan bahasa pemrograman seperti *Java*, *C++*, PHP, Python dan bahasa pemrograman lainnya.

- b. User mahir, merupakan pengguna basis data yang berinteraksi dengan sistem melalui program aplikasi yang telah dibuat sebelumnya. User ini biasanya menggunakan query SQL untuk menambahkan, mengubah ataupun menghapus data. Query SQL menyesuaikan dengan DBMS yang digunakan pada program aplikasi.
- c. User Umum, merupakan pengguna basis data yang berinteraksi melalui sistem. User umum mengeksekusi query SQL melalui bantuan program aplikasi, sehingga tidak perlu menguasai query SQL. Contoh user umum adalah teller bank, staf administrasi dan lain-lain
- d. Pengguna akhir (End User), adalah seorang pengguna basis data yang berperan dalam pengambilan keputusan pada sebuah perusahaan atau instansi.

1.4 Kegunaan Sistem Basis Data

Basis data memberikan kontribusi signifikan dalam dunia teknologi informasi. Di bawah ini beberapa kegunaan utama dari basis data:

1. Penyimpanan Data Terstruktur

Basis data menyimpan data secara terstruktur. Data diatur dalam tabel, baris dan kolom. Memungkinkan akses dan manipulasi data dengan efisien.

2. Pengelolaan Data Besar

Basis data dapat menyimpan data dengan volume yang besar secara efisien.

3. Keamanan Data

Basis data menyediakan fitur keamanan untuk melindungi data dari akses yang tidak sah seperti otentikasi pengguna, otentikasi hak akses serta kontrol akses yang terpusat.

4. Kemudahan dan Kecepatan Pencarian Data

Dengan optimasi query SQL, pengguna dapat dengan mudah mencari, mengambil dan manipulasi data sesuai dengan

kebutuhan. Kueri SQL menyerhanakan proses interaksi dengan data.

5. Penggunaan Data Bersama

Basis data memberikan kemudahan akses bagi pengguna pada waktu yang bersamaan. Basis data juga memungkinkan akses pada data ke lebih dari satu pengguna.

6. Mendukung Transaksi

Basis data menyediakan data yang dapat digunakan untuk kebutuhan transaksi bisnis seperti analisa, pelaporan sampai pengambilan keputusan.

7. Keseragaamn dan Konsistensi Data

Basis data membantu konsistensi data pada pengguna. Manipulasi data hanya perlu dilakukan sekali dan akan terrefleksikan di semua aplikasi yang menggunakan data tersebut

DAFTAR PUSTAKA

- Coronel, Carlos dan Morris, Steven. 2016. Database Systems: Design, Implementation and Management. Twelve Edition. Boston : Cengage Learning.
- El Hikam, Najib Iqom. Pengantar Basis Data (Teori dan Praktik Menggunakan Microsoft Access, MySQL, dan phpMyAdmin) 2021. Bogor: Guepedia.
- KBBI. (2008). Kamus Besar Bahasa Indonesia Edisi Keempat. Jakarta: PT Gramedia Pustaka Utama.
- Rainer, R. Kelly, dan Cassey G. Cegielski. 2011. Introduction to Information Systems, Third Edition, International Student Version. Asia: John Wiley & Sons, Inc
- Roby, Yanto (2016). Manajemen Basis Data Menggunakan MySQL. Yogyakarta: Deepublish

BAB 2

KONSEP ABSTRAKSI DATA DAN SKEMA BASIS DATA, PENGANTAR TENTANG PEMODELAN DATA

Oleh Purwita Sari

Ide pokok dari basis data adalah sekumpulan catatan atau segmen pengetahuan. Suatu basis data memiliki penjelasan terstruktur mengenai jenis fakta yang tersimpan di dalamnya, yang dikenal sebagai skema. Skema tersebut menggambarkan objek yang direpresentasikan dalam basis data serta relasi di antara objek tersebut (Andaru, 2018).

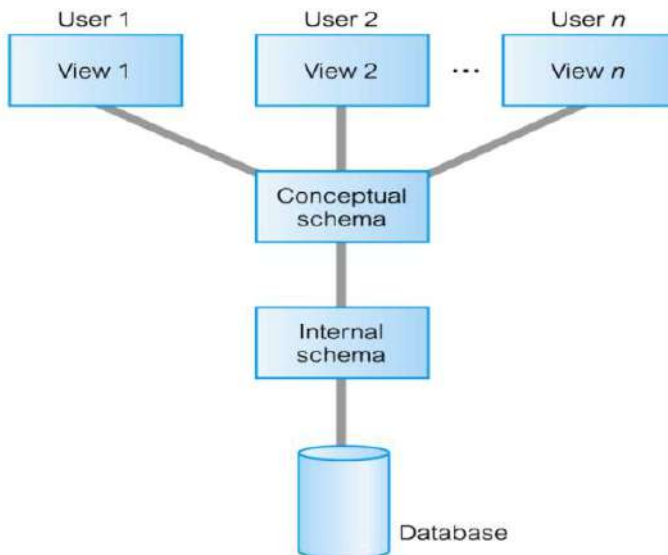
Basis Data merupakan kumpulan item data yang saling terhubung satu sama lain, disusun berdasarkan skema atau struktur tertentu, disimpan di perangkat keras komputer, dan digunakan dengan perangkat lunak untuk memanipulasi data (seperti pembaruan, pencarian, pengolahan dengan perhitungan khusus, dan penghapusan) sesuai dengan tujuan tertentu (Prasetyo and Suharyanto, 2019).

Sistem basis data adalah kumpulan data yang saling terkait, disimpan, dan dikelola oleh sebuah sistem, yang memungkinkan akses data tersebut oleh pengguna sehingga menjadi informasi. Tujuan utama dari sistem basis data adalah menyediakan pandangan data yang bersifat abstrak bagi penggunanya dengan cara menyembunyikan detail-detail tentang penyimpanan dan manipulasi data. Oleh karena itu, tahapan awal dalam merancang basis data adalah menghasilkan abstraksi dan deskripsi umum tentang kebutuhan informasi organisasi yang akan direpresentasikan dalam basis data (I Putu Dody Suarnatha *et al.*, 2023). Bagian ini akan mengulas konsep abstraksi data dan skema basis data, serta memberikan pengantar tentang pemodelan data.

2.1 Konsep Abstraksi Data dan Skema Basis Data

Abstraksi data adalah cara pengguna melihat proses pengolahan data dalam sistem basis data sehingga mirip dengan pengalaman sehari-hari pengguna. Sebuah DBMS seringkali menyembunyikan detail tentang penyimpanan dan pemeliharaan data dalam sistem basis data untuk mempermudah penggunaan DBMS tersebut. Akibatnya, data yang dilihat oleh pengguna mungkin berbeda dengan data yang sebenarnya disimpan secara fisik (Yanto, 2016).

Skema basis data adalah penjelasan tentang basis data yang dibuat secara spesifik pada tahap perancangan. Skema ini berfungsi untuk memisahkan antara struktur fisik basis data dengan program aplikasi pengguna. Representasi skema basis data umumnya ditampilkan dalam bentuk diagram yang mencakup sebagian detail dari deskripsi basis data (Munif, 2013). Deskripsi lengkap dari sebuah basis data dikenal sebagai skema basis data. Terdapat tiga jenis skema yang berbeda dalam sebuah basis data, dan masing-masing didefinisikan sesuai dengan tiga level abstraksi arsitektur tersebut (Elmasri, 2021), (Hasugian, 2020).



Gambar 2.1. Tiga Level Abstraksi Data
(Sumber : Jayanti, Sumiari and others, 2018)

Ada tiga tingkatan abstraksi (Andaru, 2018) yang meliputi :

1. Level Fisik (*Physical Level*)

Level ini merupakan tingkatan terbawah dalam struktur abstraksi data yang menggambarkan penyimpanan sebenarnya dari data. Pada tingkat ini, detail struktur data dapat dilihat dengan rinci, namun hanya dapat diakses oleh pengembang.

2. Level Konseptual (*Conceptual Level*)

Konseptual Level berada pada tingkat yang lebih tinggi dibandingkan dengan Physical Level. Pada tingkat ini, dijelaskan tentang jenis data yang disimpan oleh basis data dan hubungan keseluruhan antara data. Tingkat ini dikelola oleh administrator. Sebagai contoh, seperti transkrip nilai mahasiswa, di mana data yang disimpan dalam tabel saling terkait.

3. Level Pandangan (*View Level*)

Level pandangan adalah lapisan tertinggi dalam hierarki abstraksi data. Pada tingkat ini, pengguna hanya berinteraksi dengan struktur data yang sederhana dan sesuai dengan kebutuhan mereka. Setiap pengguna dapat memiliki pemahaman yang berbeda-beda tentang data, mungkin hanya melihat sebagian dari keseluruhan basis data. Sebagai contoh, bagian keuangan hanya memerlukan akses terhadap data keuangan, sehingga hanya memperhatikan pandangan tentang data keuangan. Demikian pula, bagian akuntansi hanya memerlukan data akuntansi. Oleh karena itu, tidak semua pengguna basis data membutuhkan akses ke seluruh informasi yang terdapat dalam basis data tersebut.

Sebagai contoh, mari kita lihat sebuah struktur data berupa record sebagai berikut:

```
Mahasiswa = RECORD;  
Nama       : STRING;  
Nim        : INTEGER;  
Alamat     : STRING;  
Jurusan    : STRING;
```

End:

Dalam contoh ini, record Mahasiswa terdiri dari empat field (Nama, Nim, Alamat, dan Jurusan). Setiap field memiliki nama dan masing-masing nama memiliki tipe data tertentu.

Pada level fisik, Mahasiswa dapat dijelaskan sebagai blok data yang disusun secara berurutan dalam satuan byte. Pada tingkat konseptual, setiap record dijelaskan dengan definisi tipe data. Di tingkat pandangan, pengguna tertentu hanya memiliki akses ke data yang relevan bagi mereka.

2.2 Pemodelan Data

Pemodelan data bisa dilakukan dalam berbagai jenis proyek dan pada berbagai tahap proyek. Model data tidaklah statis; tidak ada yang namanya model data akhir untuk bisnis atau aplikasi. Sebaliknya, model data harus dianggap sebagai dokumen yang dinamis yang akan mengalami perubahan sebagai respons terhadap perkembangan bisnis. Idealnya, model data harus disimpan dalam repositori sehingga dapat diakses, diperbarui, dan disunting dari waktu ke waktu. (Green, Medlin and Whitten, 2004) mengidentifikasi dua jenis model data sebagai berikut:

Data strategis pemodelan merupakan bagian dari penyusunan

1. Strategi sistem informasi yang menetapkan visi keseluruhan dan arsitektur sistem informasi yang ditetapkan. Teknik Informasi adalah pendekatan yang mencakup proses ini.
2. Pemodelan data dalam analisis sistem melibatkan analisis sistem untuk membuat model data logis sebagai bagian dari pengembangan basis data baru.

Pemodelan data adalah teknik yang digunakan untuk menguraikan kebutuhan bisnis terhadap database. Terkadang disebut sebagai pemodelan database karena model data ini pada akhirnya diimplementasikan dalam basis data.

Ada beberapa jenis model data (Dewantara *et al.*, 2023), (Lubis, 2016), (Nurchahyo, 2019) meliputi:

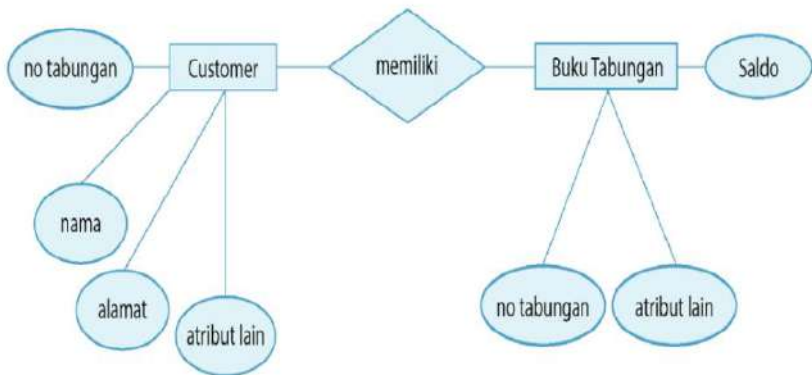
1. OBDM (*Object Based Data Model*) yang berfokus pada objek
2. RBDM (*Record Based Data Model*) yang berpusat pada record
3. PBDM (*Physical Based Data Model*) yang didasarkan pada aspek fisik

2.2.1 Model Data Berbasis Objek - OBDM (*Object Based Data Model*)

Model Data Berbasis Objek adalah kumpulan data, prosedur, atau relasi yang menggambarkan hubungan logis antar data dalam sebuah basis data berdasarkan objek data. Model ini mencakup *Entity Relationship Model* (ERM), model semantik, dan model biner.

Entity Relationship Model (ERM)

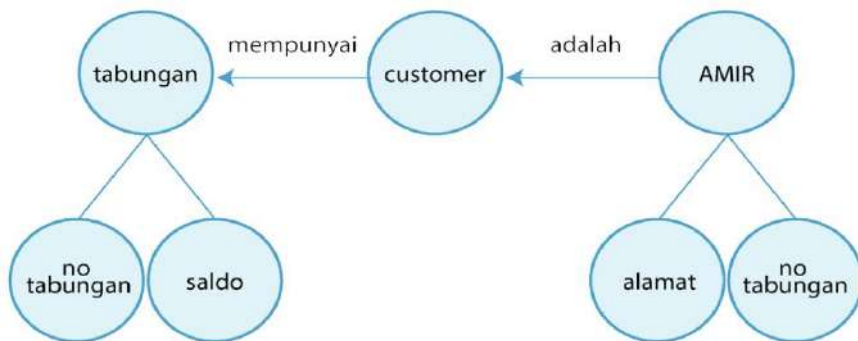
Model ini digunakan untuk menggambarkan interaksi antar data dalam basis data dengan dasar pemahaman bahwa dunia nyata terdiri dari objek-objek dasar yang memiliki keterkaitan atau relasi satu sama lain. ERM mengatur peraturan khusus yang harus dipatuhi oleh isi basis data, seperti mapping cardinalities yang menentukan jumlah entitas yang dapat dihubungkan dengan entitas lainnya melalui relationship set.



Gambar 2.2. Contoh ERM di Bank X
(Sumber : Nurcahyo, 2019)

Semantical Model

Model ini mirip dengan entity relationship model dimana hubungan antara objek dasarnya dijelaskan menggunakan simbol tetapi menggunakan semantic. Sebagai contoh, jika kita menggunakan relasi di Bank X seperti yang disebutkan sebelumnya, dalam model kita dapat melihatnya seperti yang terlihat pada Gambar 2.3. Tanda-tanda yang digunakan dalam semantical model adalah sebagai berikut.



Gambar 2.3. Contoh Semantical Model di Bank X
(Sumber : Nurcahyo, 2019)

Model Biner

Model biner adalah sebuah model data yang meluaskan pengertian tentang entitas. Model ini tidak hanya mencakup atribut-atributnya, tetapi juga tindakan-tindakannya.

2.2.2 Model Data Berbasis Rekaman-RBDM (Record Based Data Model)

Model data berbasis record adalah kumpulan data, prosedur, atau relasi yang menjelaskan keterkaitan logis antar data dalam sebuah model basis data yang berdasarkan record. Ada tiga model basis data yaitu model hierarkis, model jaringan, dan model relasional.

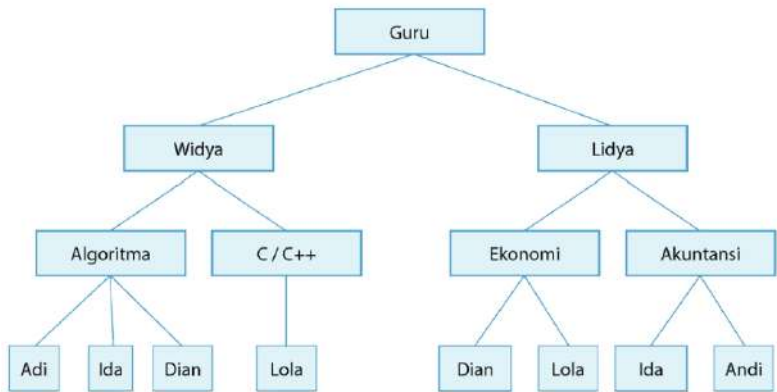
Model Hierarki

Model ini sering disebut sebagai struktur pohon yang menunjukkan hubungan berjenjang. Dalam model ini, unsur-unsurnya disebut sebagai node yang dapat berupa detail data, agregat data, dan record.

Berikut adalah beberapa istilah umum yang digunakan dalam model hirarkis:

1. Root: node yang berada pada level tertinggi dalam hirarki.
2. Parent: node yang berada pada level di atasnya.
3. Child: node yang berada pada level di bawahnya.
4. Leaves: node yang tidak memiliki child.

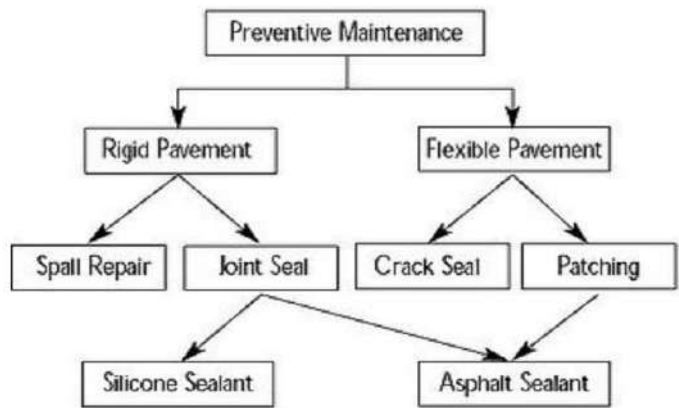
Dalam model hierarki ini, hanya ada satu root, setiap child hanya boleh memiliki satu parent, dan satu parent dapat memiliki lebih dari satu child.



Gambar 2.4. Contoh Model Hierarki
(Sumber : Nurcahyo, 2019)

Model Jaringan

Seperti model hierarki data, model ini juga menggambarkan hubungan antar data menggunakan record dan link. Perbedaannya terletak pada susunan record dan link, di mana model jaringan tersusun dari record-record dalam bentuk grafik. Dalam model ini, sebuah child dapat memiliki lebih dari satu parent, seperti yang ditunjukkan pada Gambar 2.5.



Gambar 2.5. Model Jaringan
(Sumber : Hasanah and Untari, 2019)

Model Relasional

Model ini menggambarkan data dalam bentuk tabel yang terdiri dari baris-baris yang mewakili record dan kolom-kolom yang mewakili atribut-atribut. Model ini digunakan dalam pembuatan dan desain basis data. Konsep dan istilah yang digunakan serupa dengan kondisi yang sebenarnya dialami oleh pengguna, sehingga mudah dipahami.

2.2.3 Model Data Berbasis Fisik-PBDM (*Physical Based Data Model*)

Model data berbasis fisik digunakan untuk menggambarkan bagaimana data disimpan dalam basis data secara internal, atau untuk menjelaskan kepada pengguna cara penyimpanan data secara fisik dalam media penyimpanan. Model ini jarang digunakan karena kompleksitasnya yang dapat membingungkan pengguna. Model data berbasis fisik terdiri dari unifying model dan model frame memory.

DAFTAR PUSTAKA

- Andaru, A. (2018) 'Pengertian database secara umum', *OSF Prepr*, 2.
- Dewantara, R. *et al.* (2023) *PERANCANGAN BASIS DATA*. Penerbit Lakeisha.
- Elmasri, R. (2021) 'Fundamentals of database systems seventh edition'.
- Green, K.W., Medlin, B. and Whitten, D. (2004) 'Developing optimism to improve performance: an approach for the manufacturing sector', *Industrial Management & Data Systems*, 104(2), pp. 106–114.
- Hasanah, F.N. and Untari, R.S. (2019) 'Buku Ajar Mata Kuliah Basis Data', *Umsida Press*, pp. 1–105.
- Hasugian, A.H. (2020) 'Basis Data: Edisi Revisi I'.
- I Putu Dody Suarnatha, D.S. *et al.* (2023) 'Buku Sistem Basis Data'. PT. Mifandi Mandiri Digital.
- Jayanti, N.K.D.A., Sumiari, N.K. and others (2018) *Teori basis data*. Penerbit Andi.
- Lubis, A. (2016) *Basis data dasar*. Deepublish.
- Munif, A. (2013) 'Basis data untuk SMK/MAK kelas XI semester 1'. Direktorat Jenderal Pendidikan Dasar dan Menengah.
- Nurchahyo, R. (2019) 'Basis Data', *Malang: Kuantum Buku Sejahtera* [Preprint].
- Prasetyo, K. and Suharyanto, S. (2019) 'Rancang Bangun Sistem Informasi Koperasi Berbasis Web Pada Koperasi Ikitama Jakarta', *Jurnal Teknik Komputer*, 5(1), pp. 119–126.
- Yanto, R. (2016) *Manajemen Basis Data Menggunakan MySQL*. Deepublish.

BAB 3

LINGKUNGAN BASIS DATA

Oleh Lucky Indra Kesuma

3.1 Pendahuluan

Lingkungan basis data merupakan infrastruktur yang menyediakan platform untuk menyimpan, mengelola, dan mengakses data secara efisien dan aman. Lingkungan ini terdiri dari berbagai komponen yang bekerja sama untuk mendukung operasi basis data. Berikut deskripsi singkat tentang komponen utama dalam lingkungan basis data:

1. Perangkat Keras (*Hardware*):
2. Server : Komputer yang menyediakan sumber daya untuk menjalankan sistem manajemen basis data (DBMS) dan menyimpan data.
3. Penyimpanan : Media penyimpanan seperti hard drive, solid-state drive (SSD), atau penyimpanan berbasis cloud yang digunakan untuk menyimpan data.
4. Memori (RAM) : Kapasitas memori yang digunakan oleh server untuk menyimpan data yang aktif dan mengaksesnya dengan cepat.
5. Perangkat Lunak (*Software*) : Sistem Manajemen Basis Data (DBMS): Perangkat lunak yang mengelola operasi basis data seperti penyimpanan, pencarian, pembaruan, dan penghapusan data. Contoh DBMS termasuk MySQL, PostgreSQL, Oracle, SQL Server, MongoDB, dan lain-lain.
6. Sistem Operasi : Program yang mengatur sumber daya perangkat keras dan menyediakan antarmuka antara pengguna dan perangkat keras. Sistem operasi seperti Linux, Windows Server, dan Unix sering digunakan dalam lingkungan basis data.
7. Aplikasi Klien : Perangkat lunak yang digunakan oleh pengguna untuk berinteraksi dengan basis data, baik untuk memasukkan, mengubah, menghapus, atau mengambil data.

Aplikasi klien bisa berupa aplikasi desktop, aplikasi web, atau aplikasi mobile.

8. Jaringan : Infrastruktur jaringan yang menghubungkan perangkat keras dan perangkat lunak dalam lingkungan basis data. Ini mencakup kabel, router, switch, dan protokol jaringan seperti TCP/IP. Jaringan ini memungkinkan komunikasi antara klien dan server basis data, serta antara server dalam sistem basis data terdistribusi.
9. Data : Data yang disimpan dalam basis data, termasuk berbagai jenis informasi seperti teks, angka, gambar, suara, atau video.
10. Data bisa terstruktur (dalam bentuk tabel, kolom, dan baris), semi-terstruktur (seperti dokumen XML atau JSON), atau tak terstruktur (seperti file teks atau gambar). Metadata, yang menyediakan informasi tentang struktur, definisi, dan karakteristik data, juga merupakan bagian penting dari lingkungan basis data.
11. Pengguna : Pengguna yang berinteraksi dengan basis data, termasuk administrator basis data, pengembang aplikasi, analis data, dan pengguna akhir. Setiap pengguna memiliki peran dan kewenangan yang berbeda dalam lingkungan basis data, dan mereka menggunakan perangkat lunak klien untuk mengakses data sesuai dengan kebutuhan dan kewenangan mereka.

Berikut adalah contoh kerangka dasar yang bisa digunakan untuk membuat atau memahami lingkungan basis data.

3.2 Kebutuhan Bisnis

Identifikasi kebutuhan bisnis yang mendasari penggunaan basis data merupakan langkah kritis dalam merancang lingkungan basis data yang efektif. Berikut adalah beberapa cara untuk melakukan identifikasi tersebut:

1. Wawancara dengan Pihak Terkait:

Langkah awal yang penting adalah melakukan wawancara dengan pemangku kepentingan utama dalam organisasi. Ini termasuk manajemen tingkat atas, pengguna

- akhir, pengembang aplikasi, analis bisnis, dan staf IT. Wawancara ini bertujuan untuk memahami tujuan bisnis, kebutuhan informasi, dan tantangan yang dihadapi.
2. **Analisis Dokumen Bisnis:**
Tinjau dokumen bisnis seperti rencana strategis, laporan keuangan, kebijakan perusahaan, dan analisis persaingan. Dokumen-dokumen ini dapat memberikan wawasan tentang tujuan bisnis, proses operasional, dan kebutuhan informasi yang relevan.
 3. **Observasi Langsung Proses Bisnis:**
Melakukan observasi langsung terhadap proses bisnis yang ada dapat membantu dalam mengidentifikasi kebutuhan informasi secara langsung. Dengan mengamati bagaimana orang bekerja dan berinteraksi dengan data, Anda dapat mengidentifikasi kelemahan dalam aliran informasi dan peluang untuk meningkatkan efisiensi.
 4. **Survei dan Kuesioner:**
Menggunakan survei atau kuesioner kepada pengguna potensial dapat membantu dalam mengumpulkan masukan tentang kebutuhan mereka terkait informasi dan data. Pertanyaan dapat mencakup kebutuhan akses, jenis data yang diperlukan, frekuensi penggunaan, dan preferensi terkait antarmuka pengguna.
 5. **Analisis Gap:**
Lakukan analisis gap antara keadaan saat ini dan keadaan yang diinginkan. Identifikasi di mana kekurangan dalam akses data atau kebutuhan informasi yang tidak terpenuhi. Ini dapat membantu dalam menentukan area-area di mana basis data dapat memberikan nilai tambah bagi bisnis.
 6. **Mengidentifikasi Kebutuhan Analisis dan Pelaporan:**
Tinjau kebutuhan analisis dan pelaporan untuk memahami jenis informasi yang diperlukan untuk pengambilan keputusan yang efektif. Ini termasuk pemahaman tentang jenis laporan yang dibutuhkan, frekuensi pelaporan, dan tingkat detail yang diinginkan.

7. Konsultasi dengan Profesional IT:

Dalam beberapa kasus, mungkin perlu berkonsultasi dengan profesional IT yang berpengalaman dalam merancang dan mengimplementasikan basis data. Mereka dapat memberikan wawasan tentang praktik terbaik dalam mengidentifikasi kebutuhan bisnis dan merancang solusi yang tepat.

Dengan menggunakan kombinasi metode di atas, Anda dapat mengidentifikasi kebutuhan bisnis yang mendasari penggunaan basis data dengan lebih efektif, sehingga memastikan bahwa solusi yang diimplementasikan memenuhi harapan dan memperkuat operasi bisnis. Lalu tentukan tujuan utama penggunaan basis data dan kebutuhan fungsional yang harus dipenuhi.

3.3 Model Data

Membuat model data yang mencerminkan struktur informasi yang akan disimpan dan hubungan antara entitas merupakan langkah penting dalam merancang basis data yang efisien dan sesuai dengan kebutuhan bisnis. Berikut adalah langkah-langkah untuk membuat model data:

1. Identifikasi Entitas:

Identifikasi entitas utama yang relevan dalam domain bisnis Anda. Entitas adalah objek atau konsep yang dapat disimpan dalam basis data. Contohnya bisa berupa karyawan, produk, pesanan, pelanggan, dan sebagainya.

2. Identifikasi Atribut:

Untuk setiap entitas yang diidentifikasi, tentukan atribut-atribut yang mendefinisikan entitas tersebut. Atribut adalah karakteristik atau properti dari entitas. Misalnya, untuk entitas "karyawan", atributnya bisa berupa nama, alamat, tanggal lahir, dan sebagainya.

3. Definisikan Kunci Primer:

Pilih satu atau beberapa atribut yang dapat digunakan sebagai kunci primer untuk setiap entitas. Kunci primer adalah atribut (atau kombinasi atribut) yang unik mengidentifikasi setiap baris (rekord) dalam tabel entitas. Misalnya, nomor identitas karyawan bisa menjadi kunci primer untuk entitas "karyawan".

4. Tentukan Hubungan Antara Entitas:

Identifikasi hubungan antara entitas. Hubungan antara entitas dapat berupa "*one-to-one*", "*one-to-many*", atau "*many-to-many*". Misalnya, satu karyawan dapat memiliki banyak pesanan ("*one-to-many*" antara entitas "karyawan" dan "pesanan").

5. Model Hubungan:

Gambarkan hubungan antara entitas menggunakan diagram model entitas-relasi (ERD). ERD menggambarkan entitas sebagai kotak dan hubungan antara entitas sebagai garis yang menghubungkannya. Anda juga dapat menambahkan kardinalitas (misalnya, "1" atau "N") untuk menunjukkan banyaknya entitas yang terlibat dalam hubungan.

6. Normalisasi:

Normalisasi adalah proses untuk mengorganisasi tabel-tabel dalam basis data sehingga mengurangi redundansi data dan meningkatkan integritas data. Normalisasi membantu memastikan bahwa basis data Anda diorganisasi dengan baik dan dapat diakses dengan efisien.

7. Validasi dan Iterasi:

Validasi model data dengan pemangku kepentingan bisnis dan tim teknis Anda. Pastikan bahwa model mencerminkan dengan akurat struktur informasi dan hubungan antara entitas dalam domain bisnis Anda. Jika diperlukan, lakukan iterasi untuk memperbaiki dan memperbaiki model berdasarkan umpan balik yang diterima.

8. Implementasi dalam DBMS:

Setelah model data disetujui, implementasikan struktur yang didefinisikan dalam basis data menggunakan perangkat lunak DBMS yang sesuai. Buat tabel-tabel berdasarkan entitas dan hubungan yang ditentukan dalam model, serta terapkan kunci primer dan kunci asing sesuai kebutuhan.

Dengan mengikuti langkah-langkah ini, Anda dapat membuat model data yang mencerminkan struktur informasi yang akan disimpan dan hubungan antara entitas dengan akurat dan efisien, yang menjadi dasar bagi pengembangan basis data yang sukses.

3.4 Sistem Manajemen Basis Data (DBMS)

Sistem Manajemen Basis Data (DBMS) adalah perangkat lunak yang dirancang khusus untuk mengelola penyimpanan, pengambilan, dan pembaruan data dalam basis data. DBMS bertanggung jawab atas tugas-tugas seperti penciptaan, pengelolaan, dan penghapusan tabel; pelaksanaan operasi kueri untuk mengambil data; menjaga integritas data dengan menerapkan aturan referensial; mengamankan akses data dengan menerapkan kontrol akses; dan menyediakan mekanisme pemulihan data dalam kasus kegagalan sistem.

DBMS menyediakan antarmuka untuk pengguna dan aplikasi untuk berinteraksi dengan basis data, memungkinkan mereka untuk menyimpan, mengambil, dan memperbarui informasi dengan cara yang terstruktur dan terkontrol. DBMS juga menyediakan berbagai fitur dan fungsi, termasuk dukungan transaksi, keamanan data, pemulihan bencana, pemantauan kinerja, dan banyak lagi. Pilih DBMS yang paling sesuai dengan kebutuhan bisnis dan model data. Instal dan konfigurasi DBMS sesuai dengan kebutuhan.

3.5 Desain Skema Basis Data

Desain skema basis data adalah proses merancang struktur dasar dari basis data, termasuk tabel, kolom, kunci, dan hubungan antara tabel. Desain skema yang baik sangat penting untuk memastikan basis data dapat menyimpan data dengan efisien, memungkinkan akses data yang cepat, dan menjaga integritas data. Berikut adalah langkah-langkah umum untuk desain skema basis data:

1. **Identifikasi Kebutuhan Bisnis:**

Langkah pertama adalah memahami kebutuhan bisnis yang akan diakomodasi oleh basis data. Ini termasuk pemahaman tentang jenis data yang akan disimpan, cara data akan digunakan, dan kebutuhan fungsional bisnis lainnya.

2. **Model Data:**

Berdasarkan kebutuhan bisnis, buat model data yang mencerminkan struktur informasi yang akan disimpan dalam

- basis data. Identifikasi entitas, atribut, dan hubungan antara entitas menggunakan diagram model entitas-relasi (ERD).
3. **Normalisasi:**
Normalisasi adalah proses untuk mengorganisasi tabel-tabel dalam basis data sehingga mengurangi redundansi data dan meningkatkan integritas data. Normalisasi memastikan bahwa basis data diorganisasi dengan baik dan meminimalkan anomali data.
 4. **Definisikan Tabel dan Kolom:**
Berdasarkan model data, tentukan tabel-tabel yang diperlukan dan kolom-kolom dalam setiap tabel. Pastikan setiap tabel memiliki kunci primer yang unik untuk mengidentifikasi setiap baris.
 5. **Tentukan Kunci Asing:**
Tentukan kunci asing untuk menentukan hubungan antara tabel. Kunci asing memungkinkan Anda untuk membuat hubungan antara baris dalam satu tabel dengan baris dalam tabel lainnya.
 6. **Optimalkan Skema:**
Optimalkan skema untuk performa dan efisiensi. Ini bisa termasuk pemilihan tipe data yang tepat, pembuatan indeks untuk kolom yang sering digunakan dalam operasi pencarian, dan pemilihan strategi penyimpanan yang tepat.
 7. **Validasi dan Pemantauan:**
Validasi skema dengan pemangku kepentingan bisnis dan tim teknis Anda. Pastikan bahwa skema memenuhi kebutuhan bisnis dan dapat mendukung operasi aplikasi dengan baik. Selain itu, lakukan pemantauan terus-menerus terhadap kinerja skema basis data dan siap untuk melakukan penyesuaian jika diperlukan.
 8. **Dokumentasi:**
Dokumentasikan skema basis data dengan baik. Ini termasuk deskripsi tabel, kolom, kunci, hubungan, serta aturan-aturan dan konvensi yang diterapkan dalam desain skema.
Dengan mengikuti langkah-langkah ini, Anda dapat merancang skema basis data yang efisien, sesuai dengan kebutuhan bisnis, dan dapat diandalkan untuk menyimpan dan mengelola data

dengan baik. Tentukan skema basis data, termasuk tabel, kolom, kunci, dan hubungan antara tabel. Optimalkan skema untuk performa dan keamanan.

3.6 Bahasa *Query*

Pelajari bahasa *Query* yang digunakan oleh DBMS yang dipilih (misalnya, SQL).

Latih pengguna dalam menggunakan bahasa kueri untuk memanipulasi dan mengambil data.

3.7 Keamanan Basis Data

Keamanan basis data sangat penting untuk melindungi data sensitif dan mencegah akses yang tidak sah. Berikut adalah beberapa langkah yang dapat diambil untuk meningkatkan keamanan basis data:

1. Kontrol Akses:
Terapkan kontrol akses yang ketat untuk memastikan bahwa hanya pengguna yang sah yang memiliki izin untuk mengakses data tertentu. Ini dapat dicapai dengan memberikan hak akses yang tepat kepada pengguna berdasarkan peran dan tanggung jawab mereka.
2. Enkripsi Data:
Gunakan enkripsi untuk melindungi data yang disimpan dalam basis data. Enkripsi dapat digunakan untuk mengamankan data yang disimpan di repositori, selama transmisi data melalui jaringan, atau dalam penyimpanan cadangan data.
3. Audit dan Pemantauan:
Terapkan mekanisme audit dan pemantauan untuk melacak aktivitas pengguna dalam basis data. Audit log dapat digunakan untuk merekam aktivitas seperti login, perubahan skema, atau akses data yang tidak sah.
4. Pengelolaan Kredensial:
Lindungi kredensial akses ke basis data dengan baik. Pastikan bahwa kata sandi pengguna kuat dan terenkripsi, dan gunakan teknologi otentikasi ganda jika memungkinkan.
5. Pemulihan Bencana:

Tetapkan strategi pemulihan bencana yang solid untuk melindungi data dari kehilangan atau kerusakan. Ini termasuk pembuatan cadangan reguler dan pengujian rutin proses pemulihan.

6. Pembaruan dan Pemeliharaan:

Pastikan bahwa perangkat lunak DBMS dan sistem operasi yang digunakan dalam lingkungan basis data selalu diperbarui dengan patch keamanan terbaru. Lakukan pemeliharaan rutin dan perbaikan keamanan untuk menjaga basis data tetap aman.

7. Kontrol Fisik:

Lindungi perangkat keras dan infrastruktur fisik yang mendukung basis data dari akses yang tidak sah. Ini termasuk pengamanan fisik server, pengendalian akses ke ruang server, dan perlindungan terhadap kebakaran atau bencana alam lainnya.

Dengan mengimplementasikan langkah-langkah ini, Anda dapat meningkatkan keamanan basis data Anda dan melindungi data sensitif dari ancaman keamanan yang berpotensi merugikan. Selain itu, selalu penting untuk terus memantau tren dan teknologi keamanan terbaru untuk tetap berada di depan ancaman keamanan yang berkembang.

Terapkan kontrol akses yang sesuai untuk melindungi data sensitif.

Pertimbangkan langkah-langkah keamanan fisik dan logis untuk melindungi integritas data.

3.8 Backup dan Pemulihan

Rencanakan strategi backup reguler untuk melindungi data terhadap kehilangan.

Buat rencana pemulihan bencana untuk memulihkan basis data dalam skenario kegagalan sistem.

3.9 Pemeliharaan dan Pengelolaan

Jadwalkan pemeliharaan rutin seperti optimisasi kinerja, pembaruan perangkat lunak, dan pemantauan kapasitas.

Tetapkan tim atau individu yang bertanggung jawab atas pengelolaan dan pemeliharaan basis data.

3.11 Skalabilitas

Pertimbangkan kemungkinan pertumbuhan data di masa depan dan persiapkan basis data untuk skala yang lebih besar jika diperlukan.

3.12 Integrasi dengan Aplikasi:

Pastikan basis data dapat diintegrasikan dengan aplikasi bisnis yang ada atau yang akan dikembangkan.

Uji integrasi antara aplikasi dan basis data untuk memastikan konsistensi dan kinerja yang baik.

Ini hanya merupakan kerangka dasar dan terdapat banyak aspek tambahan yang bisa ditambahkan tergantung pada kebutuhan spesifik proyek dan lingkungan pengembangan.

DAFTAR PUSTAKA

- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Database System Concepts. McGraw-Hill Education.
- Ramakrishnan, R., & Gehrke, J. (2003). Database Management Systems. McGraw-Hill Education.
- Connolly, T., & Begg, C. (2014). Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson.
- Elmasri, R., & Navathe, S. B. (2016). Fundamentals of Database Systems. Pearson.
- Date, C. J. (2012). An Introduction to Database Systems. Pearson.
- Atzeni, P., & Ceri, S. (2019). Database Systems: Concepts, Languages, and Architectures. McGraw-Hill Education.
- Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). Database Systems: The Complete Book. Pearson.
- Ozsu, M. T., & Valduriez, P. (2011). Principles of Distributed Database Systems. Springer.
- Kimball, R., & Ross, M. (2013). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. John Wiley & Sons.
- White, T. (2015). Hadoop: The Definitive Guide. O'Reilly Media.

BAB 4

NORMALISASI

Oleh Yance Sonatha

4.1 Konsep Dasar Normalisasi

Normalisasi Basis Data adalah suatu metode pendekatan dalam mengembangkan desain logika basis data relasional. Pendekatan ini tidak secara langsung terkait dengan model data, melainkan melibatkan penerapan aturan dan kriteria standar tertentu. Prinsip utama dari normalisasi ini adalah menghasilkan struktur tabel yang terorganisir dengan baik. Secara umum, proses pembentukan relasi atau tabel basis data ke dalam bentuk normal bertujuan untuk mengurangi ambiguitas atau bahkan menghilangkannya (Silberschatz et al,2019).

4.1.1 Sejarah Singkat Normalisasi

Normalisasi Basis Data dimulai pada tahun 1970 bersamaan dengan perkembangan model data relasional oleh Edgar F. Codd, seorang ilmuwan komputer. Melalui makalah terkenalnya, "A Relational Model of Data for Large Shared Data Banks," Codd memperkenalkan konsep kunci utama (primary key) untuk mengidentifikasi setiap baris dalam tabel secara unik. Tingkat normalisasi pertama yang diusulkan adalah bentuk normal pertama/ First Normal Form (1NF), yang menetapkan bahwa setiap sel dalam tabel harus menyimpan nilai atomik dan setiap baris harus unik (Connolly & Strachan, 2001).

Berlanjut dari 1NF, Codd dan peneliti lainnya merumuskan tingkat-tingkat normalisasi berikutnya, seperti Second Normal Form (2NF) dan Third Normal Form (3NF), untuk mengatasi redundansi data dan meningkatkan integritas struktur basis data. Kemudian, tingkat-tingkat normalisasi lanjutan seperti Boyce-Codd Normal Form (BCNF), Fourth Normal Form (4NF), dan Fifth Normal Form (5NF) muncul dengan aturan dan kriteria khusus. Normalisasi bukan hanya konsep teoretis, tetapi juga mendapatkan penerapan

praktis dalam pengembangan basis data relasional, didukung oleh sistem manajemen basis data (DBMS) seperti IBM's System R dan Oracle Database yang mulai berkembang pada periode ini (Korth, 2002).

4.1.2 Anomalisasi Data

Anomali data merujuk pada situasi atau kondisi yang tidak biasa atau tidak diinginkan dalam suatu set data. Dalam konteks basis data, anomali data sering kali terkait dengan struktur basis data yang tidak terorganisir dengan baik atau tidak dinormalisasi dengan benar.

Berikut adalah beberapa jenis anomali data yang umumnya terjadi :

1. Anomali *Insertion* (Anomali Penyisipan): Terjadi ketika kita mencoba menyisipkan data baru ke dalam tabel, tetapi ada ketergantungan antar atribut yang tidak terpenuhi.
2. Anomali *Update* (Anomali Pembaruan): Terjadi ketika kita mencoba memperbarui nilai suatu atribut dalam satu baris tabel, tetapi tidak konsisten di semua baris yang seharusnya memiliki nilai yang sama.
3. Anomali *Deletion* (Anomali Penghapusan): Terjadi ketika kita mencoba menghapus data dari tabel, dan sebagai hasilnya, kita kehilangan informasi yang sebenarnya masih relevan untuk data lain di dalam tabel.
4. Redundansi data merujuk pada pengulangan atau duplikasi informasi dalam suatu basis data. Lebih spesifik, ini terjadi ketika informasi yang sama disimpan lebih dari satu kali dalam sistem.

4.1.3 Tujuan Normalisasi Data

Tujuan normalisasi data dalam basis data adalah mengurangi redundansi, meningkatkan integritas data, dan menghindari anomali data. Normalisasi membantu merancang struktur basis data sedemikian rupa sehingga data disimpan dengan cara yang efisien, konsisten, dan dapat dielola dengan baik. Berikut adalah beberapa tujuan khusus dari normalisasi data:

1. Mengurangi Redundansi
Normalisasi membantu menghindari duplikasi data yang tidak perlu. Dengan membagi tabel-tabel besar menjadi tabel-tabel kecil dan mengorganisir data sesuai dengan aturan normalisasi, kita dapat meminimalkan pengulangan informasi. Ini membantu menghemat ruang penyimpanan dan mengurangi kerumitan dalam pemeliharaan data.
2. Meningkatkan Integritas Data
Normalisasi membantu memastikan integritas referensial dan integritas data secara umum. Dengan mengurangi ketergantungan fungsional antar atribut dalam tabel, risiko terjadinya ketidaksesuaian data atau inkonsistensi dapat diminimalkan. Ini menjadikan data lebih dapat diandalkan dan akurat.
3. Menghindari Anomali Data
Anomali data terjadi ketika terjadi penyisipan, pembaruan, atau penghapusan data dan dapat menyebabkan ketidakseimbangan atau ketidaksesuaian dalam basis data. Normalisasi membantu menghindari anomali data dengan merinci aturan dan struktur yang mengatur cara data disimpan dan diorganisir.
4. Memudahkan Pemeliharaan dan Pembaruan
Dengan struktur yang terorganisir, normalisasi membuat pemeliharaan dan pembaruan data menjadi lebih mudah. Jika perubahan perlu dilakukan, perubahan tersebut hanya perlu dilakukan di satu tempat, dan tidak perlu memperbarui banyak salinan data yang sama.
5. Meningkatkan Kinerja Kueri
Normalisasi dapat meningkatkan kinerja kueri karena data yang terorganisir dengan baik memungkinkan kueri untuk dieksekusi dengan lebih efisien. Ketergantungan fungsional yang baik dan struktur yang terurai membantu meningkatkan kecepatan pencarian dan pengambilan data.

Dengan mencapai tujuan-tujuan ini, normalisasi membantu menciptakan basis data yang lebih efisien, mudah dikelola, dan dapat diandalkan, memastikan bahwa data yang disimpan

mencerminkan dunia yang diwakilinya dengan akurat dan konsisten.

4.2 Ketergantungan Fungsional

Ketergantungan fungsional adalah konsep kunci dalam perancangan basis data yang merujuk pada hubungan antara atribut dalam tabel. Ketergantungan fungsional dapat juga dijelaskan sebagai keterkaitan antara dua atribut atau kolom dalam suatu tabel atau relasi, umumnya mencerminkan hubungan antara Primary Key (PK) dan atribut non-kunci lainnya.

4.2.1 Definisi Ketergantungan Fungsional

Definisi formal Ketergantungan Fungsional (*Functional Dependency*) adalah sebagai berikut :

Diberikan dua baris data (r_1 dan r_2) dalam sebuah tabel X, atribut A dikatakan memiliki ketergantungan terhadap B ($A \rightarrow B$) jika, ketika $r_1(A) = r_2(A)$, maka juga $r_1(B) = r_2(B)$. Dalam konteks ini, A disebut sebagai determinan, sedangkan B disebut sebagai dependen.

Secara lebih rinci, ini mengindikasikan bahwa atribut A memiliki ketergantungan fungsional terhadap atribut B jika dan hanya jika setiap kali nilai atribut A pada dua baris data sama ($r_1(A) = r_2(A)$), maka nilai atribut B pada dua baris data tersebut juga harus sama ($r_1(B) = r_2(B)$). Dengan kata lain, nilai atribut B secara fungsional bergantung pada nilai atribut A.

Dalam notasi, ketergantungan fungsional ini dapat diwakili sebagai $A \Rightarrow B$, menandakan bahwa atribut A secara fungsional menentukan atribut B, dan atribut B bergantung pada atribut A.

4.2.2 Jenis-Jenis Ketergantungan Fungsional

Berikut ini adalah beberapa jenis ketergantungan fungsional yang sering ditemukan dalam perancangan basis data (Efendy, 2018) :

1. Ketergantungan Fungsional Penuh (*Full Functional Dependency*)

Suatu atribut B dikatakan memiliki ketergantungan fungsional penuh pada atribut A dalam suatu tabel jika B tergantung

sepenuhnya pada A dan tidak tergantung pada bagian dari A. Dengan kata lain, jika A menentukan nilai unik B dan tidak ada subbagian dari A yang juga dapat menentukan B.

Contoh :

KodeMK → Nama_MK

(artinya NamaMK bergantung penuh pada nilai dari atribut KodeMK)

dari contoh di atas dapat disimpulkan bahwa 1 atribut menentukan 1 atribut lainnya

2. Ketergantungan Fungsional Parsial (*Partial Functional Dependency*)

Terjadi ketika atribut non-kunci (non-prime) tergantung hanya pada sebagian dari kunci utama, bukan pada keseluruhan kunci utama. Dalam desain basis data yang baik, seharusnya tidak ada ketergantungan fungsional parsial untuk mencegah redundansi data.

Contoh : NIM, nama → idRuang

(artinya idRuang tergantung pada NIM dan nama)

sehingga apabila nama dihilangkan, NIM masih terdapat ketergantungan dengan idRuang.

3. Transitive Dependency

Ketergantungan fungsional transitive terjadi ketika ada tiga atribut A, B, dan C dalam tabel, di mana A mempengaruhi B dan B mempengaruhi C. Dalam hal ini, A secara tidak langsung mempengaruhi C melalui B. Normalisasi basis data bertujuan untuk mengurangi ketergantungan fungsional transitive dengan memecah tabel menjadi tabel-tabel yang lebih kecil.

Contoh :

No_Pengirim → Nama_Pengirim, Kd_Barang

Kd_Barang → Nama_Barang, Jenis

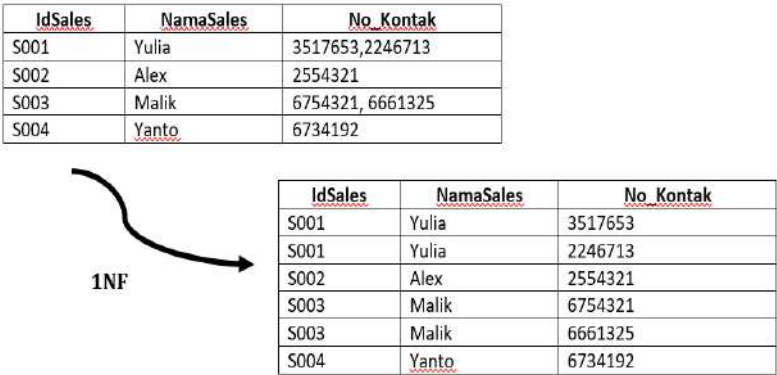
4.3 First Normal Form (1NF)

Normalisasi tingkat 1 (First Normal Form atau 1NF) adalah tingkat pertama dalam proses normalisasi basis data. Konsep ini dirancang untuk menghilangkan duplikasi data dan memastikan bahwa setiap kolom dalam tabel hanya berisi nilai atomik atau nilai

yang tidak dapat dipecah lebih lanjut. Dengan kata lain, 1NF menetapkan bahwa setiap sel dalam tabel harus berisi satu nilai tunggal, bukan kumpulan nilai atau struktur data yang kompleks (Fathansyah, 2018).

Syarat Normalisasi 1 NF :

1. Nilai Atomik. Setiap sel dalam tabel harus berisi nilai atomik atau nilai tunggal yang tidak dapat dipecah lagi. Ini berarti setiap kolom harus berisi satu nilai, dan tidak ada kolom yang memiliki nilai yang terdiri dari sejumlah nilai terpisah.



Gambar 4.1. Normalisasi 1NF untuk Pengecekan Nilai Atomik

2. Semua Kolom dengan Nama Unik. Setiap kolom dalam tabel harus memiliki nama yang unik dan berisi jenis data yang sesuai. Tidak boleh ada dua kolom dengan nama yang sama. Contoh :



Gambar 4.2. Normalisasi 1NF untuk Pengecekan Nama Kolom Secara Unik

3. Setiap Baris Unik. Tidak boleh ada baris yang sama atau duplikat dalam tabel. Setiap baris harus unik.

Contoh :

NIM	KodeMK	Nilai
101	SD	A
	BD	B
	SO	A
102	SD	B
	BD	C
	SI	B



NIM	KodeMK	Nilai
101	SD	A
101	BD	B
101	SO	A
102	SD	B
102	BD	C
102	SI	B

Gambar 4.3. Normalisasi 1NF untuk Pengecekan Setiap Baris Secara Unik

4.4 Second Normal Form (2NF)

Suatu tabel disebut memenuhi bentuk normal kedua (2NF) jika dan hanya jika, memenuhi ketentuan sebagai berikut (Suryadi, 2019) :

1. Tabel tersebut harus sudah memenuhi 1NF
Jadi proses normalisasi 2NF baru bisa dilakukan jika tabel tersebut sudah memenuhi kriteria 1NF.
2. Setiap atribut yang bukan kunci utama(non key) tergantung penuh secara fungsional terhadap semua atribut kunci dan bukan hanya sebagian atribut kunci.

Berarti jika terdapat kolom non-kunci yang bergantung pada hanya sebagian dari kunci utama, maka tabel tersebut tidak memenuhi 2NF. Dengan kata lain proses normalisasi 2NF menghilangkan **ketergantungan parsial** yang mungkin ada pada tabel.

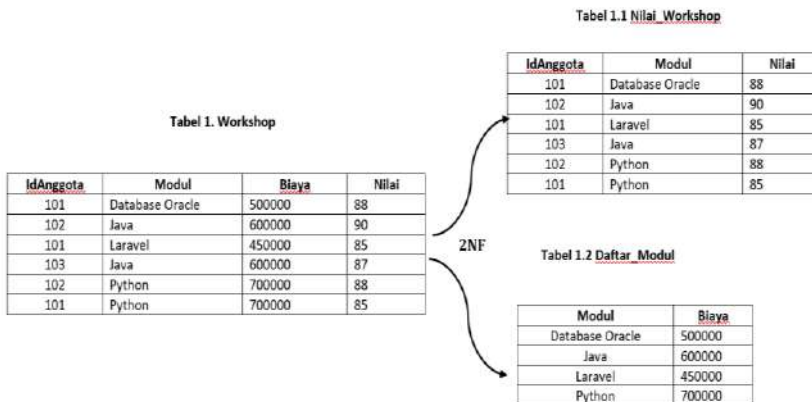
Contoh :

Pada Gambar 4.4 berikut terdapat sebuah tabel yakni Tabel Workshop yang memiliki atribut/field IdAnggota, Modul, Biaya dan Nilai. Primary Key dari tabel tersebut adalah gabungan IdAnggota dengan Modul (*Composite Field*).

Ketergantungan fungsional yang ada adalah :

- IdAnggota,Modul → Nilai
Isi atribut nilai tergantung pada gabungan data idAnggota dan Modul (berarti atribut Nilai bergantung penuh pada atribut kunci tabel)
- Modul → Biaya
Ternyata atribut Biaya cukup hanya ditentukan oleh Modul saja tanpa melibatkan IdAnggota. Berarti terdapat ketergantungan parsial yakni Nilai hanya ditentukan oleh sebagian dari atribut kunci.

Hal ini menyebabkan tabel Workshop belum memenuhi 2NF sehingga perlu dilakukan dekomposisi tabel, yakni memecah tabel tersebut untuk menghilangkan ketergantungan parsial sebagai berikut :



Gambar 4.4. Normalisasi 2NF untuk Menghilangkan Ketergantungan Partial

Dari hasil dekomposisi tabel tersebut diperoleh 2 tabel yakni :

1. Tabel Nilai_Workshop, dengan atribut IdAnggota, Modul dan Nilai. Primary Key dari tabel ini adalah IdAnggota,Modul.
2. Tabel Daftar_Modul, dengan atribut Modul dan Biaya. Primary key dari tabel ini adalah Modul.

Dengan kata lain, suatu tabel yang memiliki kunci berupa field tunggal (bukan composite field), maka tidak mungkin terdapat ketergantungan parsial dan tabel tersebut sudah pasti memenuhi 2NF.

4.5 Third Normal Form (3NF)

Suatu tabel disebut memenuhi bentuk normal ketiga (3NF) jika dan hanya jika, memenuhi ketentuan sebagai berikut (Mulyati et al, 2018) :

1. Tabel tersebut harus sudah memenuhi 2NF
Jadi proses normalisasi dilakukan bertingkat, setelah dipastikan tabel memenuhi 2NF baru dilakukan pengecekan ke 3NF.
2. Tidak terdapat **ketergantungan transitif** pada atribut bukan kunci.
Ketergantungan Transitif (*Transitive Dependency*) adalah ketergantungan fungsional antara 2 (atau lebih) atribut bukan Key (Kunci/*Primary Key*). Atau dengan kata lain ketergantungan field-field yang bukan Primary Key adalah harus secara mutlak (*Full-Dependent*).

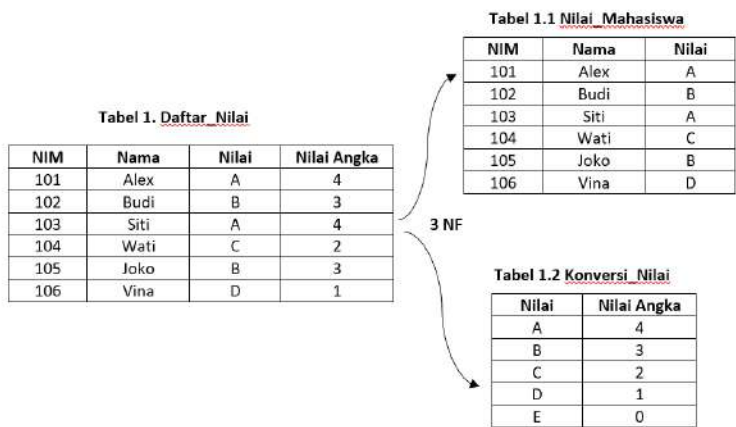
Contoh :

Pada Gambar 4.5 terdapat sebuah tabel yakni Daftar_Nilai yang terdiri dari atribut NIM, Nama, Nilai dan Nilai_Angka. Primary key dari tabel tersebut adalah NIM.

Ketergantungan fungsional yang ada adalah :

- NIM → Nama
- NIM → Nilai
Atribut Nama dan Nilai bergantung penuh pada NIM sebagai primary key, dalam arti perubahan isi field keduanya akan mengikuti perubahan NIM.
- Nilai → Nilai_Angka
Ternyata field Nilai_Angka tidak ditentukan oleh field NIM, melainkan field Nilai. Sehingga terdapat ketergantungan antar atribut yang bukan merupakan kunci tabel (*non key column*). Ini yang disebut dengan ketergantungan transitif.

Dari jabaran ketergantungan fungsional tersebut dapat disimpulkan bahwa tabel Daftar_Nilai belum memenuhi 3 NF sehingga perlu dilakukan dekomposisi tabel seperti terlihat pada Gambar 4.5 berikut.



Gambar 4.5. Normalisasi 3NF untuk Menghilangkan Ketergantungan Transitif

Dari hasil dekomposisi tabel tersebut diperoleh 2 tabel sebagai berikut :

- 1. Tabel Nilai_Mahasiswa, dengan atribut NIM, Nama dan Nilai. Primary key dari tabel ini adalah NIM.
- 2. Tabel Konversi_Nilai, dengan atribut Nilai dan Nilai_Angka. Primary key dari tabel ini adalah Nilai.

Jika sebuah tabel hanya memiliki satu atribut yang bukan merupakan kunci tabel, maka pasti tabel tersebut sudah memenuhi 3NF.

4.6 Boyce-Codd Normal Form (BCNF)

Normalisasi BCNF (*Boyce-Codd Normal Form*) adalah tingkat normalisasi yang lebih tinggi dibandingkan dengan tingkat Normal Form ketiga (3NF). BCNF dirancang untuk mengatasi anomali fungsional yang mungkin tetap ada di tingkat 3NF. Anomali fungsional tersebut disebut anomali dependensi fungsional dan

melibatkan situasi di mana sebuah atribut non-kunci tergantung pada atribut non-kunci lainnya, yang dapat memengaruhi integritas data (Elmasri & Navathe, 2001). Berikut adalah syarat BCNF :

- Tabel sudah memenuhi 3 NF
- Untuk semua ketergantungan fungsional pada sebuah tabel $A \rightarrow B$, maka A harus merupakan superkey dari tabel tersebut. Jika tidak, maka harus dilakukan dekomposisi berdasarkan ketergantungan fungsional yang ada hingga A menjadi super key dari tabel-tabel hasil dekomposisi.

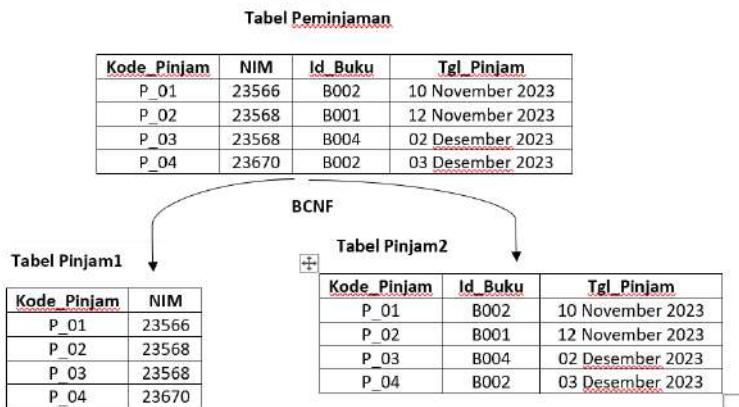
Kriteria BCNF :

Sebuah tabel dikatakan berada dalam BCNF jika, untuk setiap dependensi fungsional non-trivial $X \rightarrow Y$, X adalah superkunci.

- Dependensi fungsional non-trivial: X dan Y adalah himpunan atribut, dan Y tidak ada di dalam X atau sebaliknya.
- Superkunci: Kunci kandidat yang dapat mengidentifikasi secara unik setiap baris dalam tabel.

Contoh :

Terdapat tabel Peminjaman seperti terlihat pada Gambar 4.6, tabel tersebut terdiri dari atribut Kode_pinjam, NIM, Id_Buku dan Tgl_Pinjam.



Gambar 4.6. Normalisasi BCNF

Pada tabel Peminjaman terlihat adanya duplikasi pada NIM yang sama dengan id_buku dan tgl_pinjam yang berbeda. Setiap kode_pinjam adalah kunci kandidat, dan ada ketergantungan fungsional non-trivial yakni Kode_Pinjam → NIM. Sehingga perlu kita lakukan dekomposisi menjadi :

- Tabel Pinjam1 yang berisikan data NIM mahasiswa yang melakukan peminjaman, dengan atribut Kode_Pinjam dan NIM
- Tabel Pinjam2 yang berisikan data yang berhubungan dengan peminjaman buku, dengan atribut Kode_Pinjam, Id_Buku dan Tgl_Pinjam.

Kedua tabel ini memenuhi kriteria BCNF karena setiap dependensi fungsional non-trivial memiliki superkunci di sebelah kiri panah.

Normalisasi BCNF membantu memastikan integritas data lebih lanjut dan mengurangi risiko anomali fungsional yang mungkin terjadi pada tingkat normalisasi yang lebih rendah. Namun, perlu diingat bahwa normalisasi yang berlebihan juga dapat meningkatkan kompleksitas struktur database dan query. Oleh karena itu, perlu mempertimbangkan kebutuhan aplikasi dan keseimbangan antara normalisasi dan kinerja.

4.7 Fourth Normal Form (4NF)

Fourth Normal Form (4NF) adalah tahapan normalisasi yang lebih lanjut, setelah Third Normal Form (3NF). 4NF fokus pada ketergantungan multivalued dan hubungan antara atribut dalam tabel. Sebuah tabel dikatakan memenuhi 4NF jika tidak ada ketergantungan multi value (*multivalued dependency*) yang dapat diuraikan menjadi ketergantungan fungsional.

Contoh :

Misalkan kita memiliki tabel "Pegawai" dengan kolom-kolom berikut:

NIP (Nomor Induk Pegawai) - Primary Key
Nama_Pegawai
Daftar_Projek (Multivalued)

Tabel ini tidak memenuhi 4NF karena atribut "Daftar_Projek" adalah multivalued, dan itu dapat diurai menjadi tabel terpisah.

Tabel "Projek" baru:

NIP (Primary Key)

Nama_Projek

Tabel "Pegawai" yang diperbarui:

NIP (Primary Key)

Nama_Pegawai

Dengan normalisasi ini, kita menghilangkan dependensi multivalued, dan data disimpan secara terpisah sesuai dengan aturan 4NF.

4.8 Fifth Normal Form (5NF)

Fifth Normal Form (5NF) memperluas konsep ketergantungan lebih lanjut. Tabel dikatakan memenuhi 5NF jika tidak ada ketergantungan join (*join dependency*) yang dapat diuraikan.

Contoh :

Misalkan kita memiliki tabel "Pesanan" dengan kolom-kolom berikut:

Nomor_Pesanan (Primary Key)

Produk

Gudang (Lokasi Gudang)

Jumlah_Pesanan

Dalam hal ini, kita dapat memiliki dependensi join di antara "Produk" dan "Gudang." Jika "Produk" bergantung pada "Nomor_Pesanan," dan "Gudang" bergantung pada "Nomor_Pesanan" juga, maka kita bisa memecahnya menjadi dua tabel terpisah.

Tabel "Pesanan_Produk":

Nomor_Pesanan (Primary Key)

Produk
Jumlah_Pesanan

Tabel "Lokasi_Gudang":

Nomor_Pesanan (Primary Key)
Gudang

Dengan cara ini, kita menghilangkan ketergantungan join antara "Produk" dan "Gudang," dan tabel tersebut memenuhi 5NF.

DAFTAR PUSTAKA

- Connolly, T., Begg, C., & Strachan, A. (2001). Database Systems: A Practical Approach to Design, Implementation and Management, 3rd edition. Addison Wesley.
- Efendy, Z. (2018). Normalisasi dalam desain database. Jurnal CoreIT, 4(1), 34–43. ISSN: 2460-738X (Print). ISSN: 2599-3321 (Online).
- Elmasri, R., & Navathe, S. B. (2001). Fundamentals of Database Systems. The Benjamin/Cummings Publishing Company, Inc., California.
- Fathansyah. (2018). Basis Data Revisi III. Informatika Bandung.
- Korth, H. (2002). Database System Concepts, 4th edition. McGraw-Hill.
- Mulyati, S., Sujatmoko, B. A., Wira, T. I. M., Afif, R., & Pratama, R. A. (2018). Normalisasi database dan migrasi database untuk memudahkan manajemen data. Sebatik, 22(2), 124–129.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Database System Concepts, Seventh Edition. The McGraw-Hill Companies.
- Suryadi, S. (2019). Implementasi normalisasi dalam perancangan database relasional. U-NET: Jurnal Teknik Informatika, 3(2), 1–5. E-ISSN: 2622-8238.

BAB 5

PROYEK BERUPA KASUS DESAIN BASIS DATA DENGAN PENDEKETAN NORMALISASI

Oleh Muh Fahmi Rustan

5.1 Pendahuluan

Dalam era digital yang terus berkembang, peran data sebagai aset utama suatu organisasi semakin mendominasi. Desain basis data yang solid menjadi kunci untuk mengelola informasi dengan efisien dan efektif. Salah satu aspek penting dalam pengembangan basis data adalah normalisasi dimana normalisasi merupakan sebuah metodologi yang mengorganisir atribut pada suatu tabel sehingga di antara atribut-atribut non-kunci dapat dihilangkan sehingga dapat menghindari anomali dan meningkatkan integritas informasi (Gillenson, 1990)

Pada bab ini penulis mengajak pembaca untuk memahami proyek desain basis data melalui pendekatan normalisasi. Seperti kita ketahui Normalisasi basisdata bertujuan untuk menghilangkan ketidakteraturan, Mengurangi Redundansi, Mencegah Anomali Pengolahan Data, Meningkatkan Integritas Data, Mengurangi Risiko Kesalahan. Fokus utama kita adalah eksplorasi kasus-kasus desain basis data yang memberikan tantangan nyata, menggambarkan perjalanan dari konsepsi hingga implementasi. Seiring dengan itu, buku ini juga membahas prinsip-prinsip dasar normalisasi, mengurai kompleksitas struktur data, dan memberikan panduan praktis bagi para profesional IT, mahasiswa, dan pembaca yang tertarik dalam menguasai seni merancang basis data yang efisien.

Dengan menggabungkan teori dan aplikasi praktis, bab ini bertujuan memberikan wawasan tentang proyek desain basis data dengan mempertimbangkan aspek-aspek normalisasi. Mari bersama-sama menjelajahi dunia desain basis data yang tak hanya

menarik, tetapi juga esensial bagi kesinambungan operasional dan keberlanjutan inovasi di era teknologi informasi saat ini.

5.2 Proyek rental mobil

Dalam proyek rental mobil, normalisasi diterapkan untuk mengurangi duplikasi data, meminimalkan ruang penyimpanan, dan meningkatkan konsistensi data serta mengefisienkan update data (Trisyanto, 2018). Pada pembahasan normalisasi database rental mobil ini akan menjelajahi berbagai aspek dan manfaat terkait kasus rental mobil. Melalui penelusuran dan studi kasus yang relevan, diharapkan pembaca akan memperoleh pemahaman tentang bagaimana normalisasi basisdata yang dapat menjadi pondasi kuat untuk mengoptimalkan kinerja basis data rental mobil secara keseluruhan.

Dalam contoh kasus rental mobil kita sudah mendapatkan sebuah tabel yang telah memenuhi bentuk normalisasi 2NF, dimana setiap atribut non-kunci sepenuhnya bergantung pada kunci primer.

Tabel 5.1. Tabel member yang memenuhi 2NF

Id member	Nama member	Alamat	No telepon
M01	Noviar	Tande	081234202111
M02	Opi	Galung	082331293482
M03	Amirul	Mamuju	083998318221
M04	Sadly	Majene	084993728321

Tabel 5.2. Tabel mobil yang memenuhi 2NF

Id_mobil	Merk_mobil	Nomor_plat	Warna	Id_perusahaan	Nama perusahaan
Mb1	Innova	DC 1 B	Hitam	P1	Toyota
Mb2	Xenia	DC 2 B	Silver	P2	Daihatsu
Mb3	Brio	DC 3 B	putih	P3	Honda
Mb4	Brio	DC 4 B	merah	P3	Honda
Mb5	Rush	DC 5 B	Hitam	P1	Toyota

Pada tabel mobil terdapat atribut id mobil, merk mobil, nomor plat, warna, dan nama Perusahaan kendaraan. tabel tersebut

belum memenuhi Bentuk normal ketiga (3NF) karena atribut non-key (bukan kunci) masih memiliki ketergantungan fungsional terhadap atribut non-key lainnya. Langkah yang harus dilakukan untuk mendapatkan Bentuk normal ketiga (3NF) adalah:

Merk_mobil->	Id_perusahaan->	Nama perusahaan
		↑

Merk_mobil dan nama perusahaan memiliki ketergantungan transitif sehingga harus dipecah menjadi dua tabel yaitu : perusahaan dan mobil dengan atribut sebagai berikut:

Id_perusahaan -> nama perusahaan

Id mobil-> merk mobil, nomor plat, warna, id perusahaan

Id Perusahaan pada tabel perusahaan akan dimasukkan pada tabel mobil agar informasi Perusahaan mobil tetap ada. Berikut hasil 3NF:

Tabel 5.3. Tabel perusahaan yang memenuhi 3NF

Id_perusahaan	Nama perusahaan
P1	Toyota
P2	Daihatsu
P3	Honda

Tabel 5.4. Tabel mobil yang memenuhi 3NF

Id_mobil	Merk mobil	Nomor plat	Warna	Id_perusahaan
Mb1	Innova	Dc 1 B	Hitam	P1
Mb2	Xenia	Dc 2 B	Silver	P2
Mb3	Brio	Dc 3 B	putih	P3

Untuk semua ketergantungan fungsional pada sebuah tabel dengan notasi A ->, maka A harus merupakan super key pada tabel tersebut. Jika tidak maka tabel harus di dekomposisi berdasarkan KF yang ada hingga A menjadi super key dari tabel-tabel hasil dekomposisi. Tujuan dari BCNF ini adalah menghindari pembaruan anomaly yang masih mungkin terjadi pada 3NF (S. Sumathi & Esakkirajan, 2007). Suatu relasi memenuhi BCNF jika dan hanya jika setiap determinan yang ada pada relasi tersebut adalah candidate

key. Determinan adalah atribut dimana satu atau lebih atribut lain tergantung secara fungsional.

Berikut adalah tabel yang belum terdekomposisi yaitu tabel rental Dimana super key dari tabel rental adalah id rental dengan atribut yang masih bergantung pada super key sehingga dapat dibuat dalam Bentuk BCNF dengan id rental akan menjadi super key dari tabel tabel hasil dekomposisi.

Tabel 5.5. Tabel Rental

Id_rental	Id member	Id mobil	Tanggal rental
R01	M01	Mb2	10/12/2023
R02	M02	Mb3	23/23/2023
R03	M03	Mb1	05/01/2024
R04	M04	Mb3	10/02/2024

Tb rental 1 dan tb rental 2 merupakan hasil dari Bentuk BCNF dengan super key yaitu id rental. Dimana tb rental 1 dengan atribut id rental dan id member kemudian pada tb rental 2 dengan atribut id rental.

Tabel 5.6. Tabel Rental 1

Id_rental	Id member
R01	M01
R02	M02
R03	M03
R04	M04

Tabel 5.7. Tabel Rental 2

Id_rental	Id mobil	Tanggal rental
R01	Mb2	10/12/2023
R02	Mb3	23/23/2023
R03	Mb1	05/01/2024
R04	Mb3	10/02/2024

Perubahan normalisasi ke dalam Bentuk BCNF ini diperlukan karena :

1. **Menghindari Ketergantungan Fungsional yang Tidak Diinginkan:** Dengan membagi tabel Rental menjadi dua tabel terpisah, Anda memastikan bahwa setiap tabel hanya memiliki ketergantungan fungsional yang jelas dan relevan dengan kunci utamanya.
2. **Mengurangi Redundansi Data:** Dengan memisahkan atribut yang berbeda ke dalam tabel yang berbeda, Anda mengurangi redundansi data. Misalnya, informasi tentang member tidak perlu disimpan ulang di setiap entri Rental.
3. **Mempermudah Pemeliharaan dan Perubahan Data:** Dengan struktur yang lebih terorganisir, pemeliharaan dan perubahan data menjadi lebih mudah. Jika suatu saat nanti Anda perlu menambahkan atau mengubah informasi tentang member atau mobil, Anda hanya perlu melakukannya di satu tempat tanpa mempengaruhi tabel Rental lainnya.
4. **Meningkatkan Kinerja dan Penggunaan Ruang Penyimpanan:** Struktur yang lebih terstruktur dan efisien dapat meningkatkan kinerja basis data dan mengoptimalkan penggunaan ruang penyimpanan.

5.3 Perekaman Medis Klinik Kesehatan

Di dunia klinik kesehatan, keandalan dan keteraturan data medis adalah elemen penting dalam menyediakan pelayanan administrasi kesehatan yang berkualitas. Basis data yang terstruktur dengan baik menjadi tulang punggung dari sistem perekaman medis, memastikan bahwa setiap informasi pasien tersedia secara akurat dan mudah diakses oleh penyedia layanan kesehatan.

Dalam contoh kasus yang kedua dalam buku ini, kami akan membahas pentingnya normalisasi basis data dalam konteks perekaman medis di klinik kesehatan. Dengan menerapkan prinsip-prinsip normalisasi, kita dapat memastikan integritas data yang tinggi dan mengurangi risiko kesalahan atau inkonsistensi. Mari kita lihat contoh kasus sederhana dari penerapan normalisasi dalam perekaman medis klinik kesehatan:

Normalisasi basis data perekaman medis klinik kesehatan, Suatu relasi memenuhi 2NF jika dan hanya jika memenuhi 1NF. Setiap atribut yang bukan kunci utama tergantung secara fungsional terhadap semua atribut kunci dan tidak hanya sebagian atribut kunci. Bentuk 2NF bertujuan untuk menghilangkan ketergantungan transitif

Berikut adalah Bentuk 2NF

Id dokter -> nama dokter, spesialis, Alamat

Id pasien -> nama pasien, Alamat, jenis kelamin

Id obat -> nama obat, tgl kadaluarsa, Perusahaan obat, Alamat Perusahaan

Id rekam -> tgl periksa, id pasien, diagnosa, id dokter, id obat

Tabel 5.8. Tabel dokter yang memenuhi 2NF

Id_dokter	Nama_dokter	Spesialis	Alamat
D011	Rafiuddin	P. Dalam	Majene
D012	Udin	P. Hati	Polewali
D013	Yeni	P. Kulit	Makassar

Pada tabel dokter memiliki 4 atribut yaitu id dokter, nama dokter, spesialis dan Alamat Dimana id dokter sebagai primary key nya.

Tabel 5.9. Tabel pasien yang memenuhi 2NF

Id_pasien	Nama_pasien	Alamat	Jenis_kelamin
P01	Anto	Majene	Pria
P02	Firman	Mamuju	Pria
P03	Lina	makassar	Wanita
P04	Rani	polewal	Wanita

Pada tabel pasien memiliki 4 atribut yaitu id pasien, nama pasien, Alamat dan jenis kelamin. Dimana id pasien sebagai atribut kunci.

Tabel 5.10. Tabel obat yang memenuhi 2NF

Id_obat	Nama_obat	Jenis obat	Perusahaan_obat	Alamat_perusahaan
B01	Paracetamol	Tabelt	Kimia_farma	Jakarta
B02	Ibuprofen	Tabelt	Ap.fadila	Majene
B03	Antasida	Sirup	Farma_obat	lembang
B04	Hydrocortisone	Salep	Kimia farma	Jakarta

Pada tabel obat memiliki 5 atribut yaitu id obat, nama obat, jenis obat, Perusahaan obat dan Alamat perusahaan. Dimana id obat sebagai atribut kunci.

Tabel 5.11. Tabel rekam medis yang memenuhi 2NF

Id_rekam	Tgl_periksa	Id_pasien	Diagnosa	Id_dokter	Id_obat
R01	22-12-2023	P02	Demam	D012	B01
R02	02-01-2024	P03	Aslam	D011	B03
R03	04-01-2024	P01	Luka kulit	D013	B04
R04	10-02-2024	P04	Sakit dada	D012	B02

Pada tabel rekam medis memiliki 6 atribut yaitu id rekam, tgl_periksa, id_pasien, diagnosa, id dokter dan id obat. Dimana id rekam sebagai primary key nya dan ada 3 foreign key yang bergantung pada tabel ini yaitu id pasien, id dokter dan id obat yang tentunya ini saling berkaitan dalam menampilkan data rekam medis.

Pada normalisasi ini diperlukan perubahan ke Bentuk 3NF karena

1. **Menghindari Redundansi Data:** Dengan memisahkan informasi perusahaan obat ke dalam tabel terpisah, menghindari redundansi data yang dapat terjadi jika informasi perusahaan disimpan bersama dengan informasi obat di setiap entri obat.
2. **Integritas Data yang Lebih Baik:** Memastikan integritas data yang lebih baik dengan hanya perlu memperbarui

informasi perusahaan obat di satu tempat, yaitu dalam tabel perusahaan, untuk menghindari kesalahan atau inkonsistensi data.

3. **Fleksibilitas dalam Perubahan dan Pemeliharaan:** Memudahkan perubahan dan pemeliharaan karena struktur basis data yang ternormalisasi, sehingga penambahan atribut tambahan untuk perusahaan obat bisa dilakukan tanpa mengubah struktur tabel obat.
4. **Optimalisasi Penyimpanan dan Kinerja:** Dengan memisahkan informasi perusahaan obat ke dalam tabel terpisah, mengoptimalkan penggunaan ruang penyimpanan dan meningkatkan kinerja basis data karena tabel perusahaan biasanya lebih kecil dan efisien dalam pencarian dan akses data.

Untuk memperoleh 3NF kita harus memastikan telah memenuhi syarat berikut

1. Telah memenuhi Bentuk normal kedua (2NF)
2. Tabel yang dimiliki tidak ada ketergantungan transitif dengan kata lain jika A bergantung pada B dan B bergantung pada C, maka terdapat ketergantungan transitif yang mengakibatkan A bergantung pada C

Pada tabel obat terdapat atribut id obat, nama obat, tgl kadaluarsa, nama perusahaan dan alamat Perusahaan. Tabel tersebut belum memenuhi Bentuk normal ketiga (3NF) karena atribut non-key (bukan kunci) masih memiliki ketergantungan fungsional terhadap atribut non-key lainnya. Langkah yang harus dilakukan untuk mendapatkan Bentuk normal ketiga (3NF) adalah

Id_perusahaan -> Perusahaan obat, Alamat Perusahaan

Id obat -> nama obat, jenis obat

Id Perusahaan pada tabel obat akan dimasukkan pada tabel obat agar informasi Perusahaan obat tetap ada.

Tabel 5.12. Tabel perusahaan obat

Id_perusahaan	Perusahan_obat	Alamat_perusahaan
Pr01	Kimia_farma	Jakarta
Pr02	Ap.fadila	Majene
Pr03	Farma_obat	lembang
Pr04	Kimia farma	Jakarta

Tabel 5.13. Tabel Obat

Id_obat	Nama_obat	Jenis obat	Id Perusahaan
B01	Paracetamol	Tabelt	Pr01
B02	Ibuprofen	Tabelt	Pr02
B03	Antasida	Sirup	Pr03
B04	Hydrocortisone	Salep	Pr01

5.4 Aplikasi perpustakaan

Studi kasus yang akan dibahas adalah tentang aplikasi perpustakaan dengan kasus sebagai berikut:

Anda bekerja sebagai seorang database administrator di perpustakaan universitas. Perpustakaan tersebut memiliki basis data untuk mengelola informasi tentang buku, mahasiswa, penerbit, peminjaman, dan pengembalian. Basis data awal memiliki beberapa masalah terkait redundansi data dan anomali. Tugas Anda adalah melakukan normalisasi untuk memperbaiki masalah-masalah tersebut.

Contoh Tabel Perpustakaan :
Tabel 5.14. Contoh tabel perpustakaan (tidak normal)

Tgl Peminjaman	Kode Buku	Judul Buku	Kode penerbit	Nama Penerbit	Nama Peminjam	NIM	Tgl Pengembalian
21/01/2024	BK001	Basis Data	P123	Celebes Ekspres	Fahmi	D0215001	26/01/2024
	BK002	Matematika Diskrit					
22/01/2024	BK001	Basis Data	P123	Celebes Ekspres	Indra	D0214021	27/01/2024
	BK003	Mobile Programming	P111	Jakarta Press			
23/01/2024	BK004	Sistem Operasi	P111	Jakarta Press	Nurul Wafiqah	D0217018	28/01/2024
	BK005	Algoritma Pemrograman					
	BK002	Matematika Diskrit	P123	Celebes Ekspres			
27/02/2024	BK001	Basis Data	P123	Celebes Ekspres	Fahmi	D0215001	4/03/2024

Masalah:

- Informasi mahasiswa terduplikasi dalam tabel peminjaman.
- Terdapat redundansi informasi penerbit.
- Terdapat redundansi informasi judul buku
- Memiliki atribut yang memiliki makna yang ganda
- Belum ditentukan kunci primer
- Anomali Pengolahan Data: Jika ada perubahan pada informasi nama peminjam atau lainnya, perubahan harus dilakukan di beberapa tempat, dan kesalahan atau ketidaksesuaian bisa terjadi.

Untuk memperoleh bentuk 1NF kita harus memastikan telah memenuhi syarat berikut ini:

- Tidak terdapat data yang berulang pada setiap atribut yang ada
- Pada setiap data yang ada disetiap cell memiliki nilai tunggal/ tidak ada perulangan group
- Data yang berada pada kolom yang sama memiliki tipe data yang sama pula

Tabel 5.15. Contoh tabel perpustakaan 1NF

Tgl Peminjaman	Kode Buku	Judul Buku	Kode penerbit	Nama Penerbit	Nama Mahasiswa	NIM	Tgl Pengembalian
21/01/2024	BK001	Basis Data	P123	Celebes Ekspres	Fahmi	D0215001	26/01/2024
21/01/2024	BK002	Matematika Diskrit	P123	Celebes Ekspres	Fahmi	D0215001	26/01/2024
22/01/2024	BK001	Basis Data	P123	Celebes Ekspres	Indra	D0214021	27/01/2024
22/01/2024	BK003	Mobile Programming	P111	Jakarta Press	Indra	D0214021	27/01/2024
23/01/2024	BK004	Sistem Operasi	P111	Jakarta Press	Nurul Wafiqah	D0217018	28/01/2024
23/01/2024	BK005	Algoritma Pemrograman	P111	Jakarta Press	Nurul Wafiqah	D0217018	28/01/2024
23/01/2024	BK002	Matematika Diskrit	P123	Celebes Ekspres	Nurul Wafiqah	D0217018	28/01/2024
27/02/2024	BK001	Basis Data	P123	Celebes Ekspres	Fahmi	D0215001	4/03/2024

Pada tabel 5.15 telah dilakukan normalisasi pada atribut tanggal peminjaman, judul buku yang dipinjam, nama penerbit, nama mahasiswa yang meminjam buku, nim, dan tanggal pengembalian. Namun tabel tersebut belum memenuhi bentuk normal yang kedua (2NF).

Untuk memperoleh bentuk 2NF kita harus memastikan telah memenuhi syarat berikut ini:

1. Telah memenuhi bentuk normal yang pertama
2. Tidak ada ketergantungan parsial, artinya setiap kolom yang bukan kunci utama bergantung sepenuhnya pada kunci utama dalam hal ini kita akan menghilangkan ketergantungan sebahagian, contoh 1: nama mahasiswa memiliki dependensi dengan atribut nim, contoh 2: Nama penerbit memiliki dependensi dengan kode penerbit.

Langkah yang harus dilakukan untuk memperoleh bentuk normal yang kedua(2NF) adalah dengan menentukan fungsional dependensi dari masing-masing atribut yang ada (Joseph S. P. Fong, 2021).

Kode Buku -> Judul Buku, Kode Penerbit, Nama Penerbit

NIM -> Nama Mahasiswa

NIM, kode buku -> tgl_peminjaman, tgl_pengembalian

Hasil normalisasi (2NF) akan memperoleh tabel berikut ini:

1. Tabel buku

Tabel 5.16. Tabel buku Hasil normalisasi 2NF

Kode Buku	Judul Buku	Kode penerbit	Nama Penerbit
BK001	Basis Data	P123	Celebes Ekspres
BK002	Matematika Diskrit	P123	Celebes Ekspres
BK003	Mobile Programming	P111	Jakarta Press
BK004	Sistem Operasi	P111	Jakarta Press
BK005	Algoritma Pemrograman	P111	Jakarta Press

2. Tabel mahasiswa

Tabel 5.17. Tabel mahasiswa Hasil normalisasi 2NF

NIM	Nama Mahasiswa
D0215001	Fahmi
D0214021	Indra
D0217018	Nurul Wafiqah

3. Tabel peminjaman buku

Tabel 5.18. Tabel peminjaman buku Hasil normalisasi 2NF

id_pinjam	NIM	Kode Buku	Tgl Peminjaman	Tgl Pengembalian
1	D0215001	BK001	21/01/2024	26/01/2024
2	D0215001	BK002	21/01/2024	26/01/2024
3	D0214021	BK001	22/01/2024	27/01/2024
4	D0214021	BK003	22/01/2024	27/01/2024
5	D0217018	BK004	23/01/2024	28/01/2024

id_pinjam	NIM	Kode Buku	Tgl Peminjaman	Tgl Pengembalian
6	D0217018	BK005	23/01/2024	28/01/2024
7	D0217018	BK002	23/01/2024	28/01/2024
8	D0215001	BK001	27/02/2024	04/03/2024

Untuk memperoleh bentuk 3NF kita harus memastikan telah memenuhi syarat berikut ini:

1. Telah memenuhi bentuk normal kedua (2NF)
2. Tabel yang dimiliki tidak ada ketergantungan transitif, dengan kata lain jika A bergantung pada B dan B bergantung pada C, maka terdapat ketergantungan transitif, yang mengakibatkan A bergantung pada C.

Pada tabel buku terdapat atribut kode buku, nama buku, kode penerbit dan nama penerbit, tabel tersebut belum memenuhi bentuk normal ketiga (3NF) karena atribut non-key (bukan kunci) masih memiliki ketergantungan fungsional terhadap atribut non-key lainnya. Dimana atribut nama penerbit masih memiliki ketergantungan fungsional terhadap kode penerbit. Langkah yang harus dilakukan untuk mendapatkan bentuk normal ke tiga (3NF) adalah:

Kode penerbit -> nama penerbit

Kode buku-> judul buku, kode penerbit

Tabel 5.19. Tabel penerbit buku Hasil normalisasi 3NF

Kode penerbit	Nama Penerbit
P123	Celebes Ekspres
P111	Jakarta Press

Kode penerbit pada tabel buku tetap dimasukkan pada tabel buku agar informasi penerbit tetap ada, atau dengan kata lain informasi penebit pada tabel buku itu tidak hilang.

Tabel 5.20. Tabel penerbit buku Hasil normalisasi 3NF

Kode Buku	Judul Buku	Kode penerbit
BK001	Basis Data	P123
BK002	Matematika Diskrit	P123
BK003	Mobile Programming	P111
BK004	Sistem Operasi	P111
BK005	Algoritma Pemrograman	P111

Bentuk Normal Boyce-Codd (BCNF):

BCNF adalah tingkat normalisasi yang lebih ketat daripada 3NF, yang memperbaiki beberapa keterbatasan yang masih ada dalam 3NF(Pamungkas, 2017)

Untuk memenuhi BCNF, setiap dependensi fungsional non-trivial harus bergantung pada kunci utama, Dengan kata lain, jika sebuah tabel memenuhi BCNF, tidak boleh ada dependensi fungsional non-trivial antara atribut, yang berarti setiap atribut non-kunci secara eksklusif bergantung pada kunci utama. BCNF memastikan bahwa setiap anomali penyimpanan yang mungkin terjadi dalam 3NF dapat diatasi, sehingga meningkatkan konsistensi dan keamanan data.

Perbedaan utama antara 3NF dan BCNF terletak pada ketatnya aturan dependensi fungsional. BCNF memerlukan kondisi tambahan bahwa setiap dependensi fungsional non-trivial harus bergantung pada kunci utama, sedangkan 3NF hanya memerlukan eliminasi dependensi fungsional transitif. Dengan demikian, BCNF sering dianggap sebagai bentuk normalisasi yang lebih tinggi dan lebih ketat, tetapi bisa juga memerlukan pemodelan basis data yang lebih kompleks. Mari kita lihat kembali tabel 5.5 dimana tabel tersebut memiliki:

id_pinjam-> NIM, kode_buku, tgl_peminjaman, tgl_pengembalian
Pinjam=(id_pinjam, NIM, kode_buku, tgl_peminjaman, tgl_pengembalian)

FD : id_pinjam -> kode_buku, tgl_peminjaman, tgl_pengembalian

• Bukan BCNF, sebab id_pinjam bukan superkey

id_pinjam $\nrightarrow$ NIM

Sebuah dependensi fungsional non-trivial adalah yang tidak bisa disimpulkan dari kunci utama saja. Dalam hal ini, ID_Pinjam adalah kunci utama, dan kita memiliki dependensi fungsional non-trivial yang mengaitkan semua atribut lainnya dengan ID_Pinjam.

Namun, masalahnya adalah, bagaimana jika satu peminjam meminjam lebih dari satu buku dalam satu transaksi? Dalam skenario seperti itu, atribut NIM, Kode_Buku, Tgl_Peminjaman, dan Tgl_Pengembalian akan tergantung pada ID_Pinjam saja, karena itu menjadi kunci utama. Ini menyebabkan kelemahan pada desain, karena atribut-atribut lainnya juga memiliki dependensi fungsional yang signifikan terhadap atribut kunci selain ID_Pinjam.

Untuk memperbaiki ini, kita harus memecah tabel ini menjadi dua tabel: satu untuk informasi peminjaman umum dan satu untuk detail setiap buku yang dipinjam dalam setiap transaksi. Dengan cara ini, kita dapat memastikan bahwa setiap tabel mematuhi BCNF, dan tidak ada dependensi fungsional yang tidak trivial yang tergantung pada atribut non-kunci. sehingga tabel 5.18 didekomposisi menjadi :

Tabel 5.21. Tabel pinjam 1 Hasil normalisasi BCNF

id_pinjam	NIM
1	D0215001
2	D0215001
3	D0214021
4	D0214021
5	D0217018
6	D0217018
7	D0217018
8	D0215001

Tabel 5.22. Tabel pinjam 2 Hasil normalisasi BCNF

id_pinjam	Kode Buku	Tgl Peminjaman	Tgl Pengembalian
1	BK001	21/01/2024	26/01/2024
2	BK002	21/01/2024	26/01/2024
3	BK001	22/01/2024	27/01/2024
4	BK003	22/01/2024	27/01/2024
5	BK004	23/01/2024	28/01/2024
6	BK005	23/01/2024	28/01/2024
7	BK002	23/01/2024	28/01/2024
8	BK001	27/02/2024	04/03/2024

DAFTAR PUSTAKA

- GILLENSON, M. L. (1990). *FUNDAMENTALS OF DATABASE MANAGEMENT SYSTEMS* (2nd ed.).
- Joseph S. P. Fong, K. W. T. Y. (2021). *Information Systems Reengineering, Integration and Normalization* (4th ed.). Springer.
- Pamungkas, C. A. (2017). *Pengantar dan Implementasi Basis Data*. deepublish.
- S. Sumathi, & Esakkirajan, S. (2007). *Fundamentals of Relational Database Management Systems*. Springer.
- Trisyanto. (2018). *Analisis & Perancangan Sistem Basis Data*. Garuda Mas Sejahtera.

BAB 6

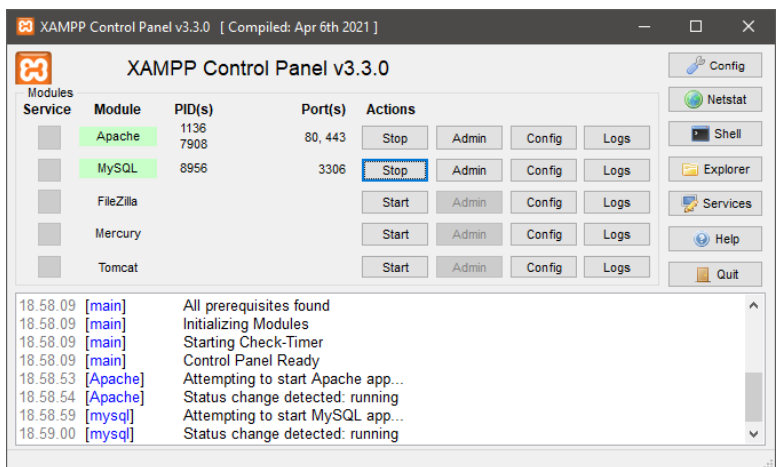
STUDI KASUS: BASIS DATA HOTEL

Oleh Lut Faizal

6.1 Membuat Database Hotel

Pada bab ini, kita akan mempelajari bagaimana cara membuat database menggunakan PHPMyAdmin dengan memanfaatkan editor skrip *structured query language* (SQL) dengan studi kasus database Hotel. Untuk lebih jelas, silahkan ikuti langkah-langkah berikut:

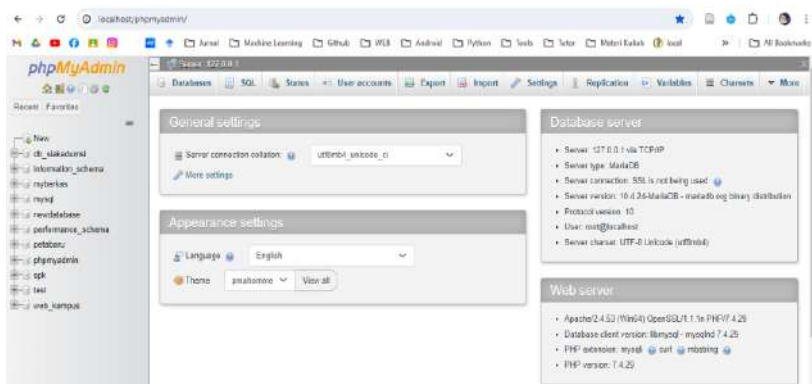
1. Buka aplikasi XAMPP Control Panel yang ada pada start menu, kemudian aktikan server / modul Apache dan MySQL dengan mengklik tombol Start di sebelah masing-masing modul. Pastikan status modul berubah menjadi hijau, menandakan bahwa server sedang berjalan seperti gambar berikut:



Gambar 6.20. Xampp Control Panel

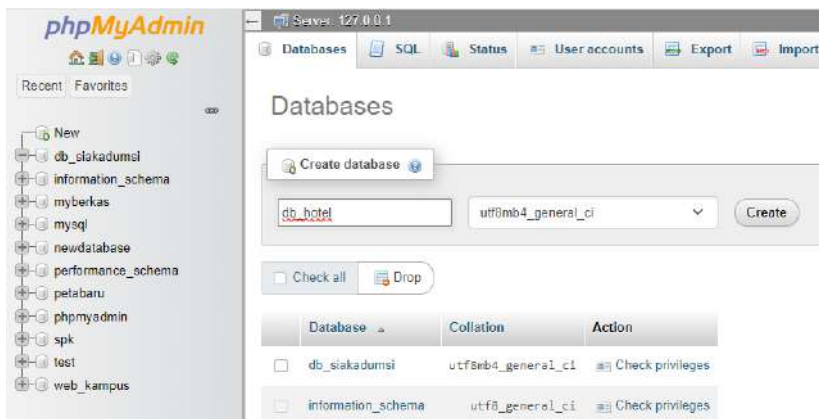
2. Buka salah satu browser (Google Chrome, Mozilla Firefox, Microsoft Edge) di komputer Anda, kemudian ketikkan

alamat **http://localhost/phpmyadmin** di bilah alamat. Tampilan yang muncul akan seperti gambar di bawah ini:



Gambar 6.21. Dashboard phpMyAdmin

3. Buat database dengan mengklik tombol new lalu isi nama data base nama db_hotel dan klik Create



Gambar 6.22. Pembuatan Database Hotel

4. Database db_hotel akan dibuat dan akan muncul dalam daftar database di PHPMyAdmin, sebagaimana terlihat pada tampilan berikut:



Gambar 6.23. List Database

Rancangan database hotel terdiri dari beberapa tabel yang menyimpan berbagai jenis data terkait operasi hotel. Tabel-tabel utama antara lain: `tbl_hotel`: Menyimpan informasi umum tentang hotel, seperti nama hotel, alamat, rating, telepon, dan informasi lainnya.

1. `tbl_user`: Menyimpan data pengguna, seperti username, password, nama lengkap, dan alamat.
2. `tbl_fasilitas`: Menyimpan informasi fasilitas yang ada di hotel, seperti nama dan deskripsi fasilitas.
3. `tbl_tipe_kamar`: Menyimpan informasi tentang tipe-tipe kamar yang tersedia, seperti nama tipe kamar dan deskripsi.
4. `tbl_kamar`: Menyimpan informasi tentang kamar hotel, termasuk nomor kamar, tipe kamar, status, fasilitas, dan harga.
5. `tbl_tamu`: Menyimpan data tamu hotel, termasuk nama, alamat, nomor telepon, dan jenis kelamin.
6. `tbl_checkin`: Menyimpan data check-in tamu, termasuk tamu, kamar, pembayaran, waktu check-in, dan waktu check-out.

6.2 Membuat Tabel Dengan Query SQL

Sejauh ini, kita telah berhasil membuat database bernama "`db_hotel`". Selanjutnya, kita akan membuat beberapa tabel di dalam database tersebut menggunakan query SQL. Tabel-tabel yang akan dibuat antara lain adalah "`tbl_hotel`", "`tbl_user`", "`tabel_fasilitas`", "`tbl_kamar`", "`tbl_tipekamar`", "`tbl_checkin`", dan "`tbl_tamu`". Tabel-tabel ini akan digunakan untuk menyimpan berbagai jenis data

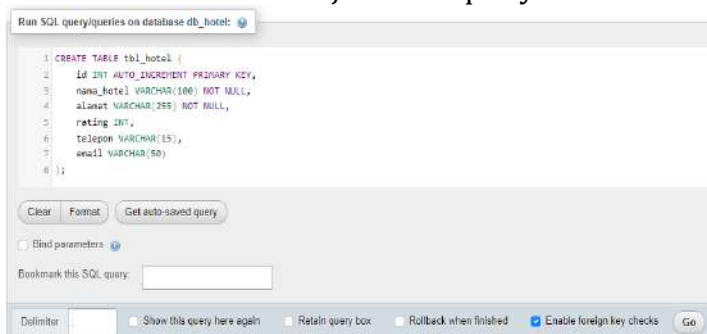
terkait hotel, seperti informasi hotel, pengguna, fasilitas, kamar, jenis kamar, data check-in, dan data tamu. Mari kita lanjutkan dengan membuat tabel-tabel tersebut:

6.2.1 Membuat Tabel Hotel

1. Klik tab "SQL" untuk membuka editor query. Masukkan query SQL berikut ke dalam editor untuk membuat tbl_hotel:

```
CREATE TABLE tbl_hotel ( id INT AUTO_INCREMENT PRIMARY KEY, nama_hotel VARCHAR(100) NOT NULL, alamat VARCHAR(255) NOT NULL, rating INT, telepon VARCHAR(15), email VARCHAR(50) );
```

2. Klik tombol "Go" untuk menjalankan query tersebut.



Gambar 6.24. SQL Editor tbl_hotel

3. Tabel "tbl_hotel" sekarang sudah dibuat dan akan menampilkan informasi berikut:



Gambar 25.6. Informasi Pembuatan tbl_hotel

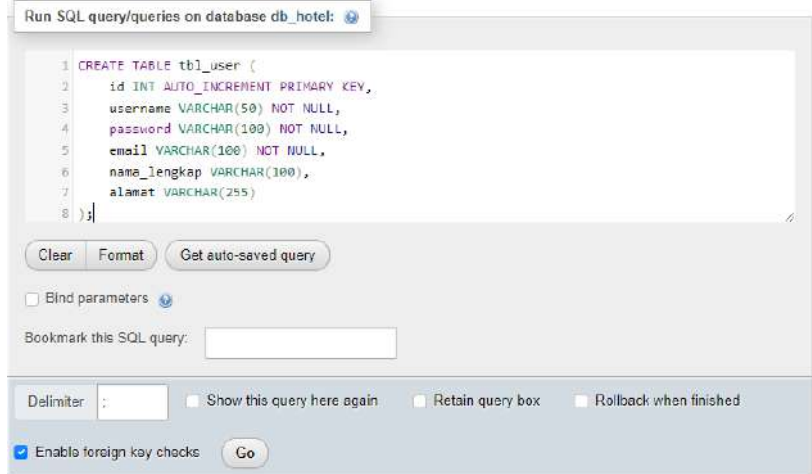
6.2.2 Membuat Tabel User

1. Masih di dalam PHPMyAdmin dengan database "db_hotel" yang dipilih, buka tab "SQL". Masukkan query SQL berikut ke dalam editor:

```
CREATE TABLE tbl_user (id INT AUTO_INCREMENT PRIMARY KEY, username VARCHAR(50) NOT NULL,
```

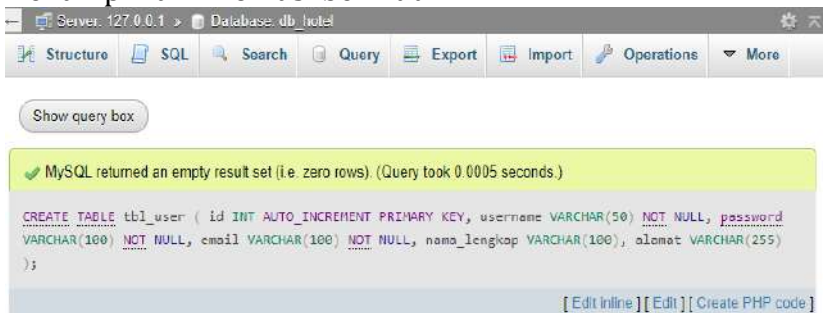
```
password VARCHAR(100) NOT NULL, email
VARCHAR(100) NOT NULL, nama_lengkap
VARCHAR(100), alamat VARCHAR(255) );
```

2. Klik tombol "Go" untuk menjalankan query tersebut.



Gambar 6.26. SQL Editor tbl_user

3. Tabel "tbl_user" sekarang sudah dibuat dan akan menampilkan informasi berikut:



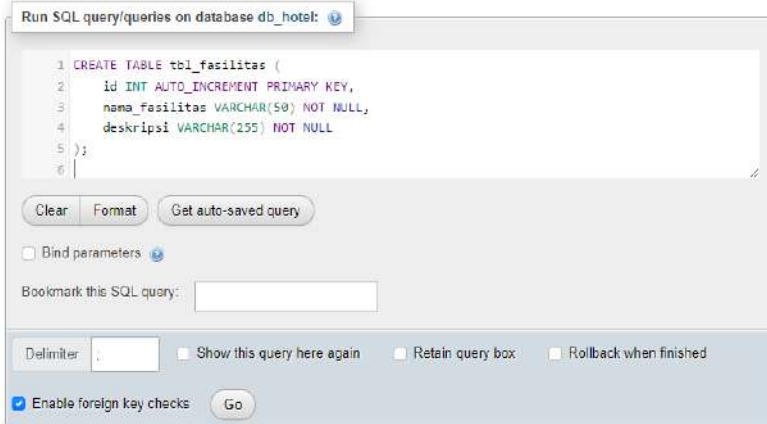
Gambar 6.27. Informasi Pembuatan tbl_user

6.2.3 Membuat Tabel Fasilitas

1. Klik tab "SQL" untuk membuka editor query. Masukkan query SQL berikut ke dalam editor untuk membuat tbl_fasilitas:

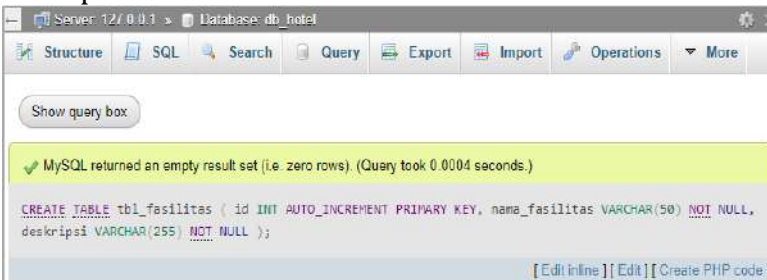
```
CREATE TABLE tbl_fasilitas ( id INT AUTO_INCREMENT  
PRIMARY KEY, nama_fasilitas VARCHAR(50) NOT NULL,  
deskripsi VARCHAR(255) NOT NULL );
```

2. Klik tombol "Go" untuk menjalankan query tersebut.



Gambar 6.28. SQL Editor tbl_fasilitas

3. Tabel "tbl_fasilitas" sekarang sudah dibuat dan akan menampilkan informasi berikut:



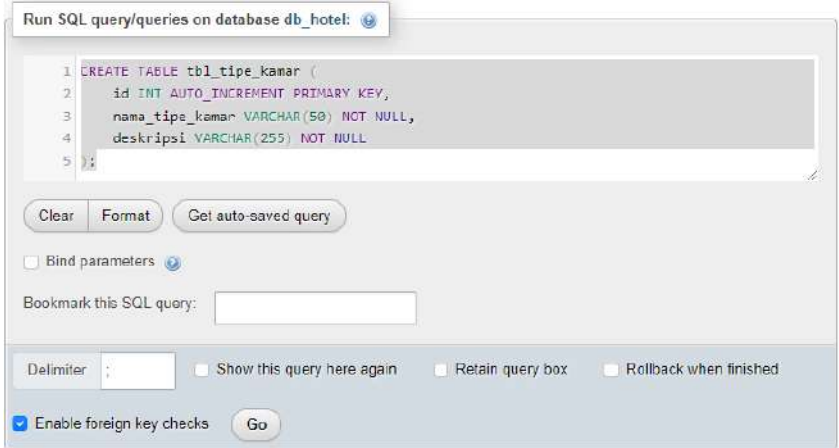
Gambar 6.29. Informasi Pembuatan tbl_fasilitas

6.2.4 Membuat Tabel Tipe Kamar

1. Klik tab "SQL" untuk membuka editor query. Masukkan query SQL berikut ke dalam editor untuk membuat `tbl_tipe_kamar`:

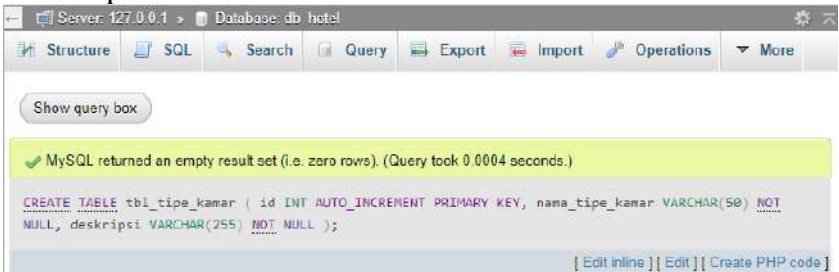
```
CREATE TABLE tbl_tipe_kamar ( id INT AUTO_INCREMENT  
PRIMARY KEY, nama_tipe_kamar VARCHAR(50) NOT NULL,  
deskripsi VARCHAR(255) NOT NULL );
```

2. Klik tombol "Go" untuk menjalankan query tersebut.



Gambar 6.30. SQL Editor `tbl_tipe_kamar`

3. Tabel "`tbl_tipe_kamar`" sekarang sudah dibuat dan akan menampilkan informasi berikut:



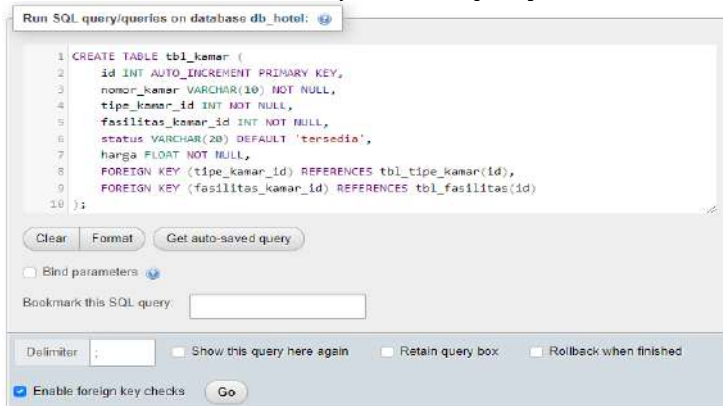
Gambar 6.31. Informasi Pembuatan `tbl_tipe_kamar`

6.2.5 Membuat Tabel Kamar

1. Klik tab "SQL" untuk membuka editor query. Masukkan query SQL berikut ke dalam editor untuk membuat `tbl_kamar`:

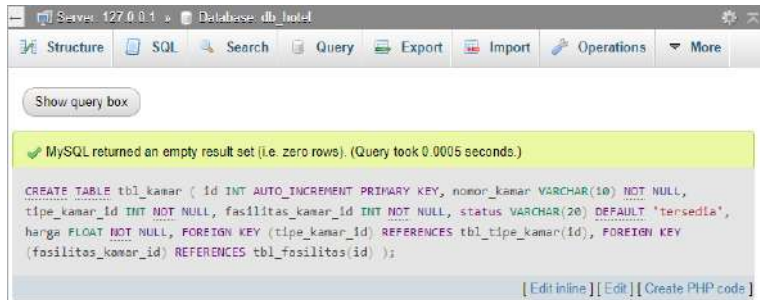
```
CREATE TABLE tbl_kamar ( id INT AUTO_INCREMENT  
PRIMARY KEY, nomor_kamar VARCHAR(10) NOT NULL,  
tipe_kamar_id INT NOT NULL, fasilitas_kamar_id INT NOT  
NULL, status VARCHAR(20) DEFAULT 'tersedia', harga FLOAT  
NOT NULL, FOREIGN KEY (tipe_kamar_id) REFERENCES  
tbl_tipe_kamar(id), FOREIGN KEY (fasilitas_kamar_id)  
REFERENCES tbl_fasilitas(id) );
```

2. Klik tombol "Go" untuk menjalankan query tersebut.



Gambar 6.32. SQL Editor `tbl_kamar`

3. Tabel "`tbl_kamar`" sekarang sudah dibuat dan akan menampilkan informasi berikut:



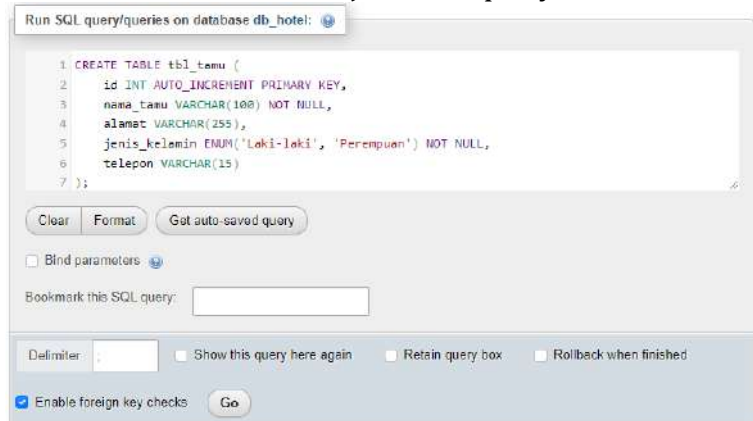
Gambar 6.33. Informasi Pembuatan tbl_kamar

6.2.6 Membuat Tabel Tamu

1. Klik tab "SQL" untuk membuka editor query. Masukkan query SQL berikut ke dalam editor untuk membuat tbl_tamu:

```
CREATE TABLE tbl_tamu ( id INT AUTO_INCREMENT
PRIMARY KEY, nama_tamu VARCHAR(100) NOT NULL, alamat
VARCHAR(255), jenis_kelamin ENUM('Laki-laki',
'Perempuan') NOT NULL, telepon VARCHAR(15) );
```

2. Klik tombol "Go" untuk menjalankan query tersebut.



Gambar 6.34. SQL Editor tbl_tamu

3. Tabel "tbl_tamu" sekarang sudah dibuat dan akan menampilkan informasi berikut:



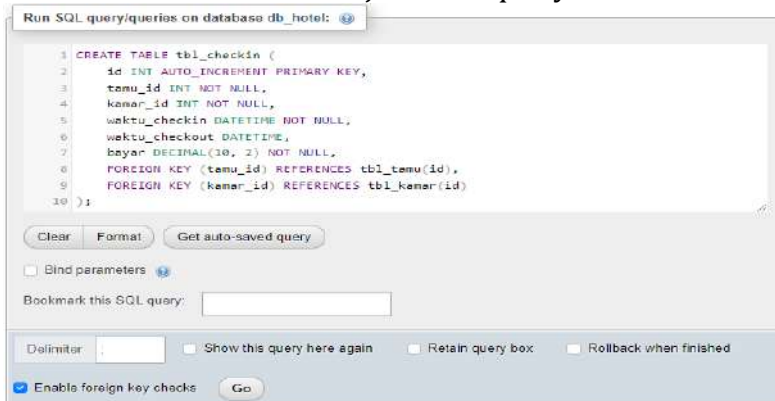
Gambar 6.35. Informasi Pembuatan tbl_tamu

6.2.7 Membuat Tabel Check In

1. Klik tab "SQL" untuk membuka editor query. Masukkan query SQL berikut ke dalam editor untuk membuat tbl_hotel:

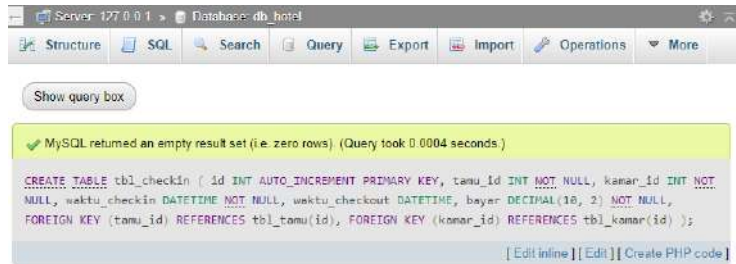
```
CREATE TABLE tbl_checkin ( id INT AUTO_INCREMENT
PRIMARY KEY, tamu_id INT NOT NULL, kamar_id INT NOT
NULL, waktu_checkin DATETIME NOT NULL, waktu_checkout
DATETIME, bayar DECIMAL(10, 2) NOT NULL, FOREIGN KEY
(tamu_id) REFERENCES tbl_tamu(id), FOREIGN KEY
(kamar_id) REFERENCES tbl_kamar(id) );
```

2. Klik tombol "Go" untuk menjalankan query tersebut.



Gambar 6.36. SQL Editor tbl_checkin

3. Tabel "tbl_checkin" sekarang sudah dibuat dan akan menampilkan informasi berikut:



Gambar 7.37. Informasi Pembuatan tbl_checkin

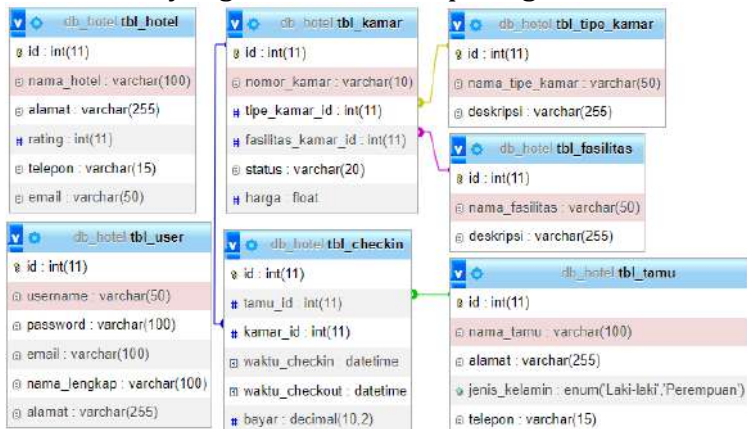
6.3 Membuat Relasi Antar Tabel

1. Untuk membuat relasi antara tabel, Anda bisa memilih menu lain-lain -> designer, seperti yang ditunjukkan pada gambar berikut:



Gambar 6.38. Menu Designer

2. Klik pada designer, maka akan tampil relasi dari tabel-tabel pada database yang telah dibuat, seperti gambar berikut:



Gambar 6.39. Relasi Antar Tabel

6.4 Menginput Data Dengan Query SQL

Untuk memasukkan data ke dalam tabel menggunakan query SQL, Anda dapat menggunakan perintah INSERT INTO. Perintah ini memungkinkan Anda menambahkan baris baru ke tabel tertentu dengan nilai-nilai yang anda tentukan. Mari kita bahas cara memasukkan data ke dalam beberapa tabel dalam database hotel. Berikut Query SQL yang dapat digunakan untuk menyimpan data ke dalam beberapa tabel yang telah dibuat:

1. Query untuk menambahkan data pada tbl_hotel

```
INSERT INTO tbl_hotel (nama_hotel, alamat, rating, telepon, email) VALUES ('Hotel Bintang', 'Jl. Mawar No.123', 4, '08123456789', 'hotelbintang@example.com')
```

2. Query untuk menambahkan data pada tbl_user

```
INSERT INTO tbl_user (username, password, email, nama_lengkap, alamat) VALUES ('user1', 'password1', 'user1@example.com', 'Nama User 1', 'Jl. Mawar No.12'), ('user2', 'password2', 'user2@example.com', 'Nama User 2', 'Jl. Melati No.34');
```

3. Query untuk menambahkan data pada tbl_fasilitas

```
INSERT INTO tbl_fasilitas (nama_fasilitas, deskripsi) VALUES ('Minibar', 'Lemari pendingin kecil berisi minuman dan makanan ringan'), ('Safe', 'Brankas pribadi untuk menyimpan barang berharga'), ('Pengering Rambut', 'Pengering rambut untuk keperluan tamu'), ('Setrika', 'Setrika dan papan setrika disediakan untuk tamu'), ('Telepon', 'Telepon untuk komunikasi internal dan eksternal'), ('TV', 'Televisi layar datar dengan saluran kabel'), ('AC', 'Pendingin udara untuk kenyamanan tamu'), ('Bathtub', 'Bathtub untuk relaksasi mandi'), ('Shower', 'Shower air panas dan dingin'), ('WiFi', 'Koneksi internet nirkabel gratis');
```

4. Query untuk menambahkan data pada tbl_tipe_kamar

```
INSERT INTO tbl_tipe_kamar (nama_tipe_kamar, deskripsi)
VALUES
('Single', 'Kamar untuk satu orang dengan fasilitas dasar'),
('Double', 'Kamar untuk dua orang dengan satu tempat tidur
besar'),
('Twin', 'Kamar dengan dua tempat tidur terpisah'),
('Suite', 'Kamar mewah dengan ruang tamu terpisah'),
('Family', 'Kamar untuk keluarga dengan dua atau lebih
tempat tidur');
```

5. Query untuk menambahkan data pada tbl_kamar

```
INSERT INTO tbl_kamar (nomor_kamar, tipe_kamar_id,
fasilitas_kamar_id, status, harga) VALUES
('101', 1, 5, 'tersedia', 500000.00),
('102', 2, 4, 'tersedia', 600000.00),
('201', 3, 3, 'tersedia', 550000.00),
('202', 4, 2, 'terisi', 750000.00),
('301', 5, 1, 'tersedia', 800000.00);
```

6. Query untuk menambahkan data pada tbl_tamu

```
INSERT INTO tbl_tamu (nama_tamu, alamat, telepon,
jenis_kelamin) VALUES
('Andi', 'Jl. Merpati No.12', '08123456789', 'Laki-laki'),
('Hilda', 'Jl. Cendrawasih No.45', '08198765432', 'Perempuan'),
('Faizal', 'Jl. Melati No.89', '08155667788', 'Laki-laki'),
('Emily Davis', 'Jl. Kenanga No.21', '08122334455',
'Perempuan'),
('Edi', 'Jl. Mawar No.67', '08166778899', 'Laki-laki');
```

7. Query untuk menambahkan data pada tbl_checkin

```
INSERT INTO tbl_checkin (tamu_id, kamar_id, waktu_checkin,
waktu_checkout, bayar) VALUES
(1, 5, '2024-04-20 14:00:00', '2024-04-22 12:00:00',
1000000),
(2, 4, '2024-04-21 15:00:00', '2024-04-23 11:00:00',
1200000),
(3, 3, '2024-04-22 13:00:00', '2024-04-24 10:00:00',
1100000),
(4, 2, '2024-04-23 14:30:00', '2024-04-25 12:30:00',
```

```
1400000),  
(5, 1, '2024-04-24 16:00:00', '2024-04-26 14:00:00',  
1600000);
```

DAFTAR PUSTAKA

- Awaludin, F., Sari, W. L., Hidayat, G. A., Rizky, M., & Saifudin, A. (2023). Analisa Dan Perancangan Sistem Informasi Reservasi Kamar Hotel Menggunakan Model View Controller (MVC) Pada Aplikasi Web. *Jurnal Manajemen*, 1(04).
- Harrison, G., & Feuerstein, S. (2006). *MySQL stored procedure programming* (1st ed). O'Reilly.
- Muliadi, M., Andriani, M., & Irawan, H. (2020). PERANCANGAN SISTEM INFORMASI PEMESANAN KAMAR HOTEL BERBASIS WEBSITE (WEB) MENGGUNAKAN DATA FLOW DIAGRAM (DFD). *JISI: Jurnal Integrasi Sistem Industri*, 7(2), 111. <https://doi.org/10.24853/jisi.7.2.111-122>
- Rohim, A., & Prihatin, A. (2022). Perancangan Sistem Informasi Pada Hotel Desa Wisata Taman Mini Indonesia Indah (TMII) Jakarta Berbasis Web.

BIODATA PENULIS



Dani Anggoro, S.Kom., M.Kom.

Dosen Program Studi Ilmu Komputer

Fakultas Ilmu Komputer, Universitas Muhammadiyah Metro

Penulis lahir di Metro (Lampung). Penulis adalah dosen tetap pada Program Studi Ilmu Komputer Universitas Muhammadiyah Metro (UM Metro). Menyelesaikan pendidikan S1 pada Program Studi Sistem Informasi Universitas Budi Luhur Tahun 2011 dan melanjutkan S2 Ilmu Komputer di universitas yang sama di tahun 2014. Selain dosen, penulis adalah programmer di Direktorat Teknologi Informasi Universitas Budi Luhur dari tahun 2007 sampai 2021. Konsentrasi penelitian di bidang Basis Data, Data Warehouse, Big Data, Sistem Informasi dan Kecerdasan Buatan (AI).

BIODATA PENULIS



Purwita Sari, S.SI., M.Kom.

Dosen Program Studi Manajemen Informatika
Fakultas Ilmu Komputer Universitas Sriwijaya

Penulis lahir di Palembang tanggal 09 Juni 1992. Penulis adalah dosen tetap pada Program Studi Manajemen Informatika Fakultas Ilmu Komputer, Universitas Sriwijaya. Menyelesaikan pendidikan S1 pada Jurusan Sistem Informasi dan melanjutkan S2 pada Jurusan Teknik Informatika. Penulis menekuni bidang riset tentang Sistem Informasi.

BIODATA PENULIS



Dr. Lucky Indra Kesuma, S.SI., M.Kom

Dosen Program Studi Sistem Informasi
Fakultas Ilmu Komputer Universitas Sjakhyakirti Palembang

Penulis Lahir di Palembang, 25 September 1990. Menyelesaikan pendidikan sebagai Ahli Madya pada program studi komputerisasi Akuntansi pada tahun 2011 di Universitas Sriwijaya. Melanjutkan Pendidikan sebagai sarjana S1 program studi Sistem Informasi lulus pada tahun 2014 di Universitas Sriwijaya, menempuh pendidikan kembali S2 Magister Komputer Program Studi Teknik Informatika menyelesaikan studi pada tahun 2017 di Universitas Binadarma, Pada tahun 2023 telah selesai menempuh pendidikan S3 Program Doktor program studi Ilmu Teknik (Informatika). Penulis adalah Dosen tetap pada program studi Sistem Informasi Fakultas Ilmu Komputer Universitas Sjakhyakirti Palembang.

BIODATA PENULIS



Yance Sonatha, S.Kom.,MT

Dosen Jurusan Teknologi Informasi
Politeknik Negeri Padang

Penulis lahir di Bukittinggi Sumatera Barat pada tanggal 29 Desember 1980. Penulis adalah dosen tetap pada Program Studi Manajemen Informatika Jurusan Teknologi Informasi Politeknik Negeri Padang. Menyelesaikan pendidikan S1 pada Program Studi Ilmu Komputer FMIPA Universitas Gadjah Mada kemudian melanjutkan Pendidikan S2 pada Program Studi Informatika Institut Teknologi Bandung.

BIODATA PENULIS



Muh. Fahmi Rustan., S.Kom., M.T

Dosen Program Studi Informatika
Universitas Sulawesi Barat

Penulis lahir di Majene pada tanggal 27 Desember 1991. Penulis mengawali pendidikan dasarnya di SDN 6 Majene, kemudian melanjutkan ke SMP 3 Majene. Setelah menyelesaikan pendidikan menengahnya di SMA 1 Majene, penulis menempuh pendidikan tinggi di STMIK Dipanegara Makassar, di mana penulis meraih gelar Sarjana (S1). Selama masa kuliahnya, penulis juga pernah berperan sebagai asisten laboratorium, memperkaya pengalaman dan keterampilannya dalam bidang TIK.

Tidak berhenti di situ, penulis melanjutkan pendidikannya ke jenjang pascasarjana di Universitas Hasanuddin dan berhasil meraih gelar Magister (S2). Dengan dedikasi yang tinggi terhadap dunia pendidikan, penulis kini berperan sebagai dosen tetap di Universitas Sulawesi Barat, berkontribusi dalam mencetak generasi penerus yang berkompeten. Dalam perannya sebagai dosen, penulis merupakan salah satu pengajar mata kuliah Basisdata dan Pemrograman Web.

Sebagai seorang penulis buku, Muh. Fahmi Rustan memanfaatkan pengalamannya dalam bidang informatika untuk menginspirasi dan memberikan wawasan baru bagi para pembacanya. Melalui karya-karyanya, penulis berharap dapat terus berkontribusi dalam perkembangan ilmu pengetahuan dan pendidikan di Indonesia.

BIODATA PENULIS

Lut Faizal, S.Kom., M.Kom.

Dosen Program Studi Teknologi Rekayasa Instalasi Listrik
Fakultas Teknik
Universitas Muhammadiyah Sinjai

Penulis lahir di Sinjai pada tanggal 21 Mei 1996, merupakan dosen tetap di Universitas Muhammadiyah Sinjai pada Fakultas Teknik, program studi Teknologi Rekayasa Instalasi Listrik. Menyelesaikan pendidikan S1 pada jurusan Teknik Informatika dan melanjutkan pendidikannya ke jenjang S2 pada jurusan Sistem Komputer.