

1. Mengapa desain database diperlukan?

Desain Database diperlukan menyimpan data data krusial dan memudahkan untuk melakukan pengelolaan data ,Memungkinkan penyimpanan data dalam format yang terorganisir , bisa juga mendapat data secara efisien karena sudah tertata

2. Kesalahan umum dalam desain database dan dampaknya ?

- a) Kesalahan umum bisanya terjadi saat pembuat/programmer menyimpan data secara redundant yang menyebabkan pemborosan ruang penyimpanan database tersebut.
- b) Mengabaikan relasi antar table,Yang menyebabkan keterkaitan antar data
- c) Tidak adanya backup dan recovery plan, menghindari adanya human error dan berjaga jaga untuk membuat backup.
- d) Kurangnya Indexing atau Terlalu Banyak Indexing Dampaknya tanpa index bisa membuat memperlambat saat pencarian,sebaliknya jika terlalu banyak memperlambat operasi INSERT, UPDATE, dan DELETE.

3. Konsep normalisasi (1NF, 2NF, 3NF, BCNF). ?

a) First Normal Form (1NF):

Syarat:

Setiap kolom harus memiliki **nilai atomik** (tidak boleh berisi daftar atau array).

Setiap baris harus unik dengan adanya **primary key**.

Contoh Tidak 1NF:

ID_Pelanggan	Nama	No_Telepon
1	Ali	08123, 08234
2	Budi	08567

Perbaikan ke 1NF:

ID_Pelanggan	Nama	No_Telepon
1	Ali	08123
1	Ali	08234
2	Budi	08567

CONTOH KODE UNTUK 1NF:

```
CREATE TABLE `USER_1NF` (  
  ID_PELANGGAN INT PRIMARY KEY AUTO_INCREMENT,  
  NAME VARCHAR(50) NOT NULL  
)
```

```
CREATE TABLE CONTACT_PERSON (  
  ID_PELANGGAN INT AUTO_INCREMENT,
```

```

NAME VARCHAR(50) NOT NULL,
NO_TELEPON VARCHAR(50) NOT NULL,
EMAIL VARCHAR(50) NOT NULL,
FOREIGN KEY (ID_PELANGGAN) REFERENCES `USER_1NF`(ID_PELANGGAN)
)

```

CONTOH CODE TIDAK 1NF :

```

CREATE TABLE `USER_NOT_1NF` (
  ID_PELANGGAN INT PRIMARY KEY AUTO_INCREMENT,
  NAME VARCHAR(50) NOT NULL
  NO_TELEPON VARCHAR(50) NOT NULL,
  EMAIL VARCHAR(50) NOT NULL,
)

```

b) Second Normal Form (2NF):

Syarat:

Sudah **ternormalisasikan** atau **1NF**

Semua kolom **bergantung sepenuhnya pada primary key**, tidak hanya pada sebagian primary key (jika ada primary key gabungan).

Contoh Tidak 2NF (Primary key: ID_Pesanan + ID_Pelanggan):

ID_Pesanan	ID_Pelanggan	Nama	Produk
101	1	Ali	Laptop
102	2	Budi	HP

Kesalahan: Nama hanya bergantung pada ID_Pelanggan, bukan ID_Pesanan.

Perbaikan ke 2NF (Pisahkan tabel pelanggan dan pesanan):

Tabel Pelanggan:

ID_Pelanggan	Nama
1	Ali
2	Budi

Tabel Pesanan:

ID_Pesanan	ID_Pelanggan	Produk
101	1	Laptop
102	2	HP

Contoh kode untuk 2NF :

```

CREATE Table pelanggan (
  ID_PELANGGAN INT PRIMARY KEY AUTO_INCREMENT,

```

```
NAMA VARCHAR(50) NOT NULL
)
```

```
create table pesanan_pelanggan (
    ID_PESANAN INT PRIMARY KEY AUTO_INCREMENT,
    ID_PELANGGAN INT NOT NULL,
    TANGGAL_PESAN DATE NOT NULL,
    FOREIGN KEY (ID_PELANGGAN) REFERENCES PELANGGAN(ID_PELANGGAN)
)
```

Kode diatas cukup memanggil dari idnya saj tanpa membuat nama pelanggan lagi,supaya tidak berulang ulang

c) Third Normal Form (3NF)

Syarat:

Sudah dalam 2NF.

Tidak ada atribut non-prime (bukan primary key) yang bergantung pada atribut non-prime lainnya.

Contoh Tidak 3NF:

ID_Pelanggan	Nama	Kota	Kode_Pos
1	Ali	Jakarta	10110
2	Budi	Bandung	40234

Kesalahan: Kode_Pos bergantung pada Kota, bukan langsung pada ID_Pelanggan.

Perbaikan ke 3NF:

Tabel Pelanggan:

ID_Pelanggan	Nama	Kota
1	Ali	Jakarta
2	Budi	Bandung

Tabel Kota:

Kota	Kode_Pos
Jakarta	10110
Bandung	40234

4. Boyce-Codd Normal Form (BCNF) – Mengatasi Masalah 3NF yang Masih Bermasalah

Syarat:

Sudah dalam 3NF.

Jika ada dependensi fungsional $X \rightarrow Y$, maka X harus superkey.

Contoh Tidak BCNF:

ID_Dosen	Mata_Kuliah	Ruangan
1	Basis Data	A101
2	Algoritma	B202

Kesalahan: Mata_Kuliah menentukan Ruangan, tetapi ID_Dosen juga bisa menentukan Mata_Kuliah.

Perbaikan ke BCNF:

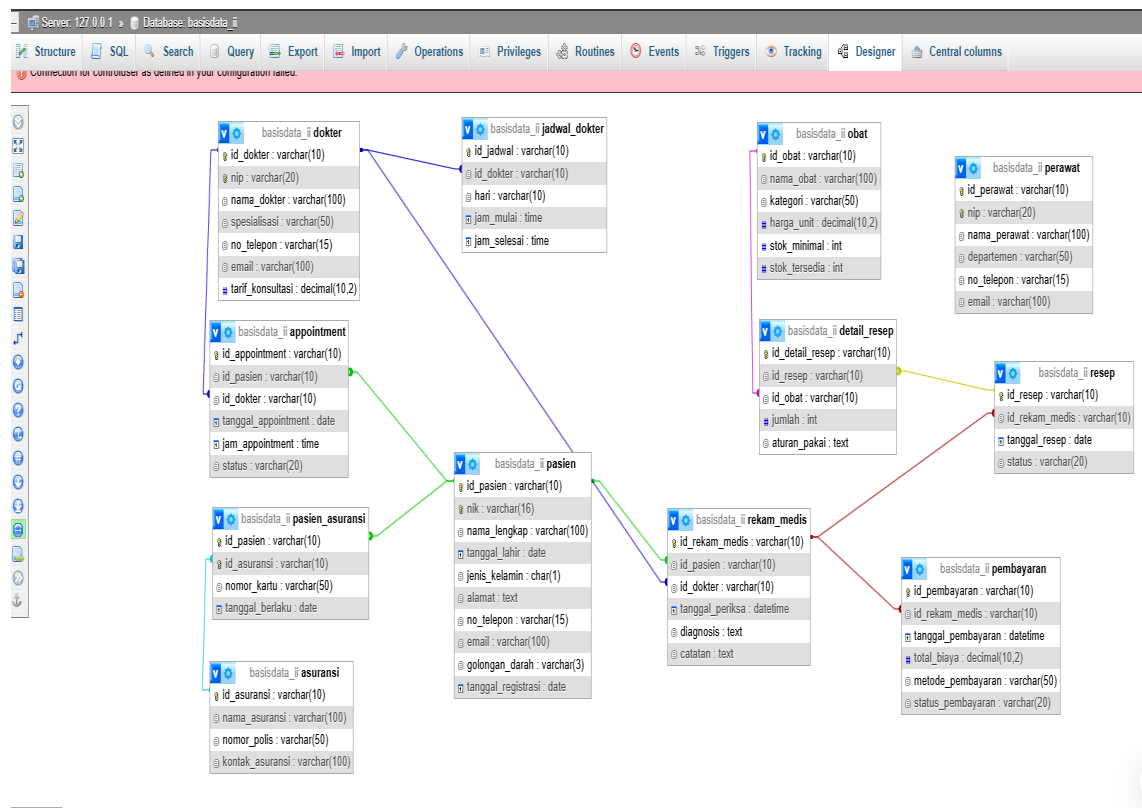
Tabel Dosen-Mata Kuliah:

ID_Dosen	Mata_Kuliah
1	Basis Data
2	Algoritma

Tabel Mata Kuliah-Ruangan :

RUANGAN	MATA KULIAH
A101	Basis Data
B202	Algoritma

TUGAS :



PENJELASAN :

Penjelasan Normalisasi:

1. First Normal Form (1NF):
 - Semua tabel memiliki primary key
 - Tidak ada repeating groups
 - Setiap kolom berisi nilai atomic
 2. Second Normal Form (2NF):
 - Sudah memenuhi 1NF
 - Semua atribut non-key bergantung sepenuhnya pada primary key
 - Dipisahkan tabel untuk menghindari partial dependency
 3. Third Normal Form (3NF):
 - Sudah memenuhi 2NF
 - Tidak ada transitive dependency
 - Setiap atribut non-key harus bergantung langsung pada primary key
- Relasi antar tabel:
- Pasien- Appointment (One to Many)
 - Dokter- Appointment (One to Many)
 - Pasien- Rekam_Medis (One to Many)
 - Rekam_Medis- Resep (One to One)

- Resep- Detail_Resep (One to Many)
- Obat- Detail_Resep (One to Many)
- Pasien- Pasien_Asuransi (One to Many)
- Asuransi- Pasien_Asuransi (One to Many)
- Rekam_Medis- Pembayaran (One to One)

Desain ini sudah memenuhi kebutuhan:

- Manajemen data pasien
- Penjadwalan dokter dan appointment
- Pencatatan rekam medis
- Pengelolaan resep dan obat
- Manajemen pembayaran dan asuransi
- Pelaporan dan administrasi

Desain ini juga mempertimbangkan:

- Data integrity
- Referential integrity
- Minimal redundancy
- Flexibility untuk pengembangan
- Security dan privacy data pasien