

UJIAN TENGAH SEMESTER



**DIAJUKAN UNTUK MEMENUHI UJIAN TENGAH SEMESTER
MATA KULIAH BASIS DATA I**

Dosen Pengampu:

Syafrial Fachri Pane, S.T., M.T.I., EBDP

Disusun Oleh:

Muhammad Fahri Ramadhan 1194055 1B

PROGRAM DIPLOMA IV TEKNIK INFORMATIKA

POLITEKNIK POS INDONESIA

BANDUNG

2020

JAWABAN SOAL A

1. **Database** terdiri dari 2 kata yaitu data yang berarti nilai yang merepresentasikan objek atau kejadian di dunia nyata. Base berarti tempat penyimpanan atau markas. Secara istilah maka database dapat diartikan sebagai himpunan data yang saling berhubungan tanpa adanya redundansi (pengulangan) yang disimpan dalam media elektronik dan diorganisir agar dapat dimanfaatkan lagi dengan efisien.
Sistem Informasi terdiri dari 2 kata yaitu sistem yang berarti kumpulan elemen yang saling berhubungan dan terorganisir untuk mencapai suatu tujuan. Informasi adalah hasil pemrosesan data yang diperoleh sehingga menjadi bentuk yang mudah dipahami bagi setiap orang.
2. **Abstraksi Data** adalah gambaran umum untuk melihat data dalam database. Abstraksi data memiliki 3 level diantaranya:
 - a. Level fisik (Physical Level)
 - Level terendah, menunjukkan bagaimana sesungguhnya data disimpan
 - Pemakai melihat data sebagai gabungan dari struktur dan datanya.
 - Pemakai mengetahui bagaimana representasi fisik dari simpanan/pengorganisasian data sebagai teks, angka bahkan bit data.
 - b. Level logik/konseptual (Conceptual Level)
 - Level yang menggambarkan data apa yang sebenarnya (secara fungsional) disimpan dalam basis data.
 - Pemakai mengetahui bahwa data pegawai disimpan dalam beberapa tabel seperti tabel identitas, tabel pendidikan, tabel keluarga dsb.
 - c. Level penampakan (View Level)
 - Level tertinggi yang menunjukkan sebagian data dari basis data.
 - Tampilan data dimata pemakai diatur oleh aplikasi end user sehingga data pada level penampakan sudah berbentuk data siap saji.
3. Pengertian dari hirarki /jenjang data untuk setiap itemnya yaitu:
 - a. **Characters** : bagian data yang terkecil, dapat berupa karakter numerik (angka 0 - 9), huruf (A - Z, a - z) ataupun karakter-karakter khusus, seperti *, &. %, # dan lain-lain.

- b. Field** : merepresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat, dsb. Setiap field harus mempunyai :
 - Field Name : Harus diberi nama yang unik
 - Field Representation : Tipe field (karakter, teks, tanggal, angka, dsb), lebar field (ruang maksimum yang dapat diisi dengan data).
 - Field Value: Isi dari field tersebut.
- c. Record / Baris Data** : Kumpulan dari field membentuk suatu record. Sebuah record menggambarkan suatu unit data individu tertentu.
- d. File/Tabel** : File terdiri dari record-record yang menggambarkan satu kesatuan data yang sejenis.

4. **Dalam pendekatan Normalisasi** perancang basis data bertitik tolak dari situasi yang nyata dimana ia telah memiliki item-item data yang siap ditempatkan dalam baris dan kolom pada tabel-tabel relasional dan ia telah mengetahui sejumlah aturan tentang keterhubungan antara item-item data tersebut.

Dalam pendekatan Model E-R perancang basis data dengan langsung membuat model data jika yang telah diketahui baru prinsip-prinsip sistem secara keseluruhan karena adanya kelangkaan data yang dimiliki.

Faktanya di lapangan, kedua pendekatan ini dilakukan bersama-sama, berganti-ganti, dan dapat saling memperkuat satu sama lain, dari contoh data (fakta) yang telah dimiliki dilakukan Normalisasi kemudian hasilnya diwujudkan dalam Model E-R, setelah itu diimplementasikan dalam bentuk sejumlah struktur tabel dalam sebuah basis data, struktur tabel ini dapat diuji kembali dengan menerapkan aturan-aturan Normalisasi, hingga akhirnya diperoleh sebuah struktur basis data yang benar-benar efektif dan efisien.

5. Langkah – langkah perancangan database:
- a. Mengumpulkan data-data yang diperlukan, lalu dianalisis.
 - b. Merancang database secara konseptual.
 - c. Memilih Data Base Management System (DBMS), lalu mengimplementasikannya.

6. 5 Macam Atribut:

- a. **Atribut Bernilai Tunggal** : Ditujukan pada atribut-atribut yang memiliki paling banyak satu nilai untuk setiap baris data.
- b. **Atribut Bernilai Banyak** : Ditujukan pada atribut-atribut yang dapat diisi dengan lebih dari 1 (satu) nilai, tetapi jenisnya sama.
- c. **Atribut Harus Bernilai** : Atribut pada sebuah tabel yang harus berisi data (nilainya tidak boleh kosong).
- d. **Atribut Bernilai Null** : Atribut yang nilainya boleh kosong. Nilai (konstanta) Null digunakan untuk mengisi atribut yang nilainya memang belum siap atau tidak ada. Nilai Null tidak sama dengan spasi.
- e. **Atribut turunan** : Atribut yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari atribut atau tabel lain yang berhubungan.

7. **Ketergantungan Fungsional** adalah hubungan antar atribut dalam suatu relasi. Jadi nilai dari satu atribut bisa sama dengan atribut lainnya karena ketergantungan fungsional tersebut.

Diberikan sebuah tabel T berisi paling sedikit 2 buah atribut, yaitu A dan B. Kita dapat menyatakan notasi berikut ini :

$$A \Rightarrow B$$

yang berarti A secara fungsional menentukan B atau B secara fungsional tergantung pada A, jika dan hanya jika untuk setiap kumpulan baris data (row) yang ada di tabel T, pasti ada 2 baris data (row) di tabel T dengan nilai untuk A yang sama, maka nilai untuk B pasti juga sama.

Definisi yang lebih formal untuk hal diatas yaitu : diberikan 2 row r_1 dan r_2 dalam tabel T dimana $A \Rightarrow B$, Jika $r_1(A) = r_2(A)$, maka $r_1(B) = r_2(B)$

8. - **Model Data Berdasarkan Object** terdiri dari: Model ketergantungan entitas, Model orientasi object, Model data semantic, Model data fungsional.
- **Model Data Berdasarkan Record** terdiri dari: Model relasional, Model hirarki, Model jaringan.

9. **Kardinalitas Relasi** : Menunjukkan jumlah hubungan maksimum yang terjadi dari setiap entitas pada himpunan entitas yang satu ke setiap entitas pada himpunan entitas yang lain dan begitu juga sebaliknya.

- a. **Satu ke Satu (One to One)**, yaitu setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A
- b. **Satu ke Banyak (One to Many)**, yaitu setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak satu entitas pada himpunan entitas A.
- c. **Banyak ke Satu (Many to One)**, yaitu setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.
- d. **Banyak ke Banyak (Many to Many)**, yaitu setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, begitu juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A

10. Sebuah tabel dikatakan baik (efisien) atau normal, jika telah memenuhi 3 kriteria berikut :

- a. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (Lossless-Join Decomposition)
- b. Terpeliharanya ketergantungan fungsional pada saat perubahan data (Dependency Preservation)
- c. Tidak melanggar Boyce Code Normal Form (BCNF) 3. Tidak melanggar Boyce-Code Normal Form (BCNF)

Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak, tabel tersebut tidak melanggar Bentuk Normal tahap Ketiga (3rd Normal Form / 3NF)

JAWABAN SOAL B

Berikut langkah-langkah yang harus dilakukan dalam melakukan normalisasi tabel yang ada pada soal:

1. Tentukan atribut yang menjadi primary key

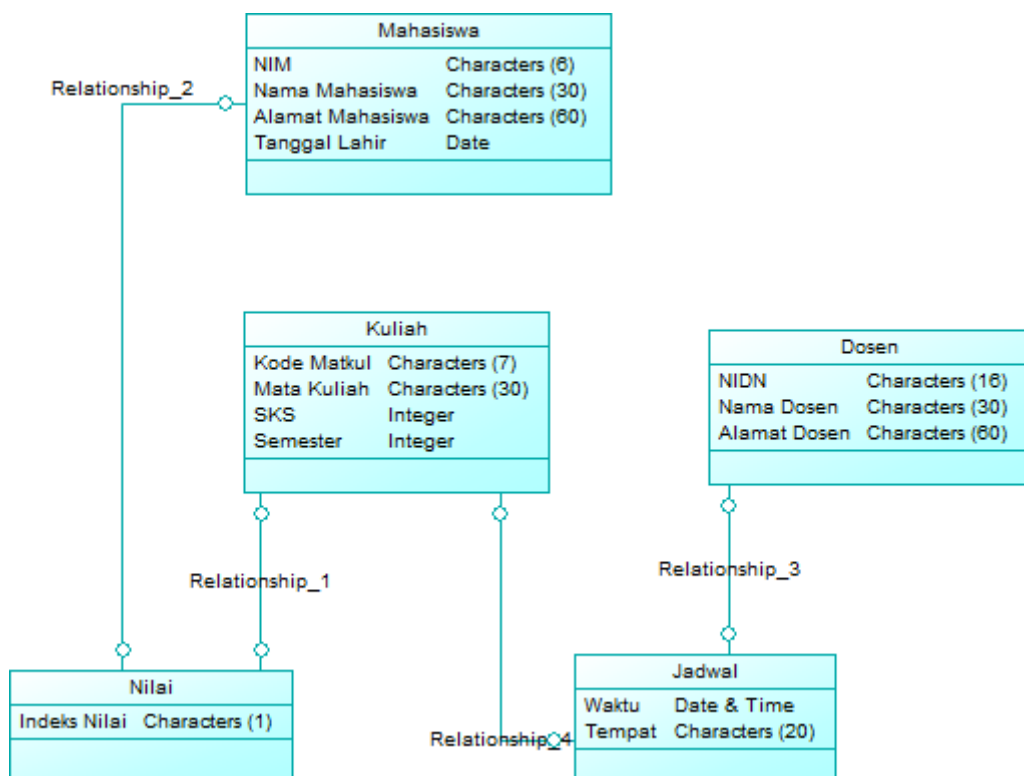
- Pada tabel Mahasiswa sudah ada atribut NIM sebagai primary key.
- Pada tabel dosen harus ditambahkan atribut NIDN sebagai Primary key.
- Pada tabel kuliah sudah ada Kode sebagai primary key.

2. Normalisasi tabel Nilai

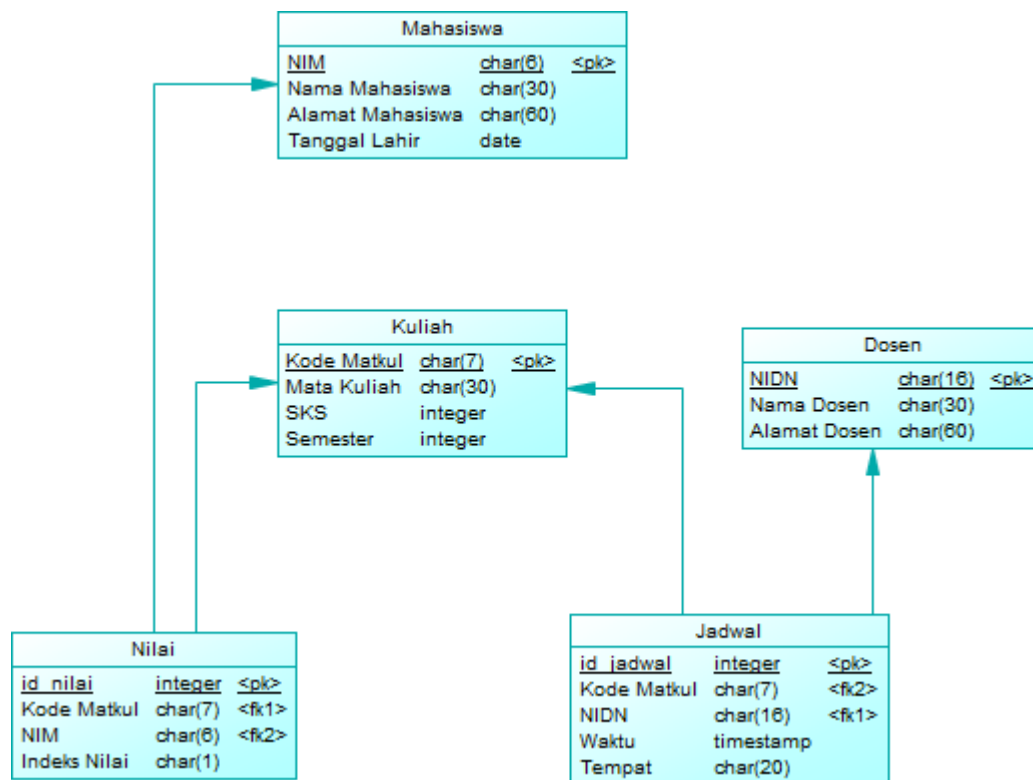
- Atribut Mata Kuliah diganti menjadi Kode Mata Kuliah, yaitu primary key pada tabel Kuliah dan menjadi foreign key saat di tabel Nilai
- Atribut Nama Mahasiswa dihapus karena sudah ada NIM sebagai foreign key pada tabel Nilai

3. Normalisasi tabel Jadwal

- Atribut Mata Kuliah diganti menjadi Kode Mata Kuliah, yaitu primary key pada tabel Kuliah dan menjadi foreign key saat di tabel Jadwal
- Atribut Nama Dosen diganti menjadi NIDN, yaitu primary key pada tabel Dosen dan menjadi foreign key saat di tabel Jadwal



Jika sudah dinormalisasi lalu dibentuk CDM maka akan terlihat tabel berelasi seperti gambar diatas.



Jika digenerate ke PDM maka akan terlihat primary key dan foreign keynya, terlihat arah relasi, dan membuat tabel lebih mudah untuk dipahami. Selain itu tabel Nilai dan Jadwal ditambahkan atribut berupa id_namatabel dan dijadikan sebagai primary key. Id ini sangat penting mengingat primary key adalah sebuah kewajiban bagi tiap tabel, agar tabel mudah direlasikan kembali, data mudah dicari, dan tidak terjadi redudansi.