**NPM : 1194059**

**Nama : Muhammad Rizal Supriadi**

**Kelas : D4TI 1B**

**UTS Basisdata1**

**Soal A**

1. **Sistem Database** merupakan suatu sistem yang terdiri dari sekumpulan file (tabel) yang tentunya saling berhubungan dalam suatu basis data didalam sebuah sistem komputer dan juga dapat dikatakan sebagai sekumpulan program yang dapat memungkinkan beberapa pemakai atau program lain untuk dapat mengakses serta memanipulasi file atau tabel tersebut.

**Sistem Informasi** merupakan suatu sistem yang dapat mengumpulkan, mengubah serta dapat menyebarkan informasi dalam suatu organisasi. Tentunya pada sistem informasi ini data yang diolah tentunya harus terintegrasi serta saling melengkapi untuk dapat menghasilkan suatu informasi yang baik, baik berbentuk gambar ataupun berbentuk tulisan.

1. **Abstraksi Data** merupakan suatu tingkatan atau dapat kita sebut dengan level yang tentunya membahas mengenai bagaimana dalam melihat suatu data dalam sebuah sistem basis data. Tentunya terdapat tiga level dari Abstraksi Data ini yaitu:

* **Level fisik (Physical Level)**

Merupakan level terendah yang dapat menunjukan bagaimana suatu data dapat disimpan, pemakai dapat melihat data sebagai suatu gabungan dari struktur datanya.

* **Level Konseptual**

Merupakan level yang dapat menggambarkan suatu data secara fungsional

disimpan dalam suatu basis data.

* **Level penampakan (view level)**

Level penampaka merupakan suatu level tertinggi yang dapat menunjukan sebagian data dari suatu sistem basis data.

1. **Hirarki/ Jenjang Data**

* Characters -> Merupakan suatu bagian data yang terkecil, serta dapat berupa karakter numerik (0- 9), huruf (A-Z, a-z) serta karakter khusus seperti $,%,& dan lain sebagainya.
* Field - > Field ini dapat merepresentasikan suatu atribut dari record yang tentunya dapat menunjukan suatu item dari data, contohnya seperti nama, alamat dan lain sebagainya.

Setiap field tentunya harus mempunyai:

* Field name: harus mempunyai nama yang unik
* Field representation: pada field ini terdapat tipe field seperti teks, karakter, tanggal, angka serta mempunya lebar field yaitu raang maksimum yang tentunya dapat diisi dengan data.
* Field value : merupakan isi dari field
* Record/ Basis Data - > merupakan suatu kumpulan dari field yang berbentuk suatu record atau data.
* File/Table -> merupakan suatu file yang terdiri dari record atau data-data yang menggambarkan suatu kesatuan data yang sejenis.

1. **Pendekatan normalisasi:** merupakan perancangan basis data yang bertitik tolak dari suatu situasi yang nyata dimana telah memiliki item data yang tentunya telah siap untuk ditempatkan dalam suatu baris dan kolom pada tabel- tabel relasional serta mengetahui sejumlah aturan mengenai keterhubungan antara item data tersebut.

**Pendekatan Model ER:** merupakan suatu perancangan basis data dengan langsung membuat model data yang telah diketahui prinsip sistem secara keseluruhan karena adanya suatu kelangkaan data yang dimiliki.

1. **Tahapan sederhana dalam merancang database:**

* Mencari contoh data yang sudah ada (dalam bentuk cetakan/ hardcopy) pada dunia nyata (real world) dari sistem yang ditinjau.
* Menganalisis contoh data tersebut untuk mengetahui karakteristik masing- masing tabel
* Membuat struktur tabel. Untuk dapat diimplemtasikan ketika kita sudah melakukan analisis

1. **Sebutkan dan jelaskan 5 macam attribut:**
2. **Key dan Attibut deskriptif** = **Key** merupakan suatu gabungan dari beberapa atribut yang dapat membedakan semua baris data (row) dalam tabel secara unik, sedangkan **Attribut deskriptif** merupakan atribut- atribut anggota dari primary key.
3. **Attibut sederhana dan Attribut Komposit = Attibut sederhana** merupakan attribute atomic yang tidak dapat diuraikan lagi menjadi sub- sub attribute. **Attribut komposit** merupakan attribut yang masih dapat diuraikan menjadi sub- sub atribut yang masing- masing memiliki makna.
4. **Attribut bernilai tunggal dan Attribut bernilai banyak = Attribut bernilai tunggal** ditujukan pada atribut yang memiliki paling banyak satu nilai untuk setiap baris data, **Attribut bernilai banyak** ditujukan pada atribut yang dapat diisi dengan lebih dari satu nilai tetapi dengan jenis yang sama.
5. **Attribute harus bernilai dan Nilai Null = harus bernilai** merupakan attribute pada sebuah tabel yang harus berisikan data, sedangkan **nilai null** merupakan attibut yang nilainya boleh kosong.
6. **Attribut turunan =** merupakan atribut yang nilainya diperoleh dari pengolahan ataupun dapat diturunkan dari attribute atau tabel lain yang tentunya berhubungan.
7. **Ketergantungan Fungsional (KF),** merupakan suatu konsep normalisasi yang menjelaskan mengenai hubungan atara suatu attribute atau dapat menjelaskan suatu nilai attribute yang dapat menuntukan nilai attibut lainnya.

Contohnya: NPM dengan Nama Mahasiswa -> NPM menentukan nama yang mahasiswa kerena suatu NPM yang sama dapat menunjukan nama Mahasiswa yang sama.

1. **- Model Lojik Data Bedasarkan Objek (Object-Based Logical Models)** merupakan suatu model lojik data yang dtentukan berdasarkan objek, yang terdiri dari:

* Model Ketergantungan Entitas (Entity-Relationship Model)
* Model Berorientasi Objek (Object-Oriented Model)
* Model Data Semantik (Semantic-Data Model)
* Model Data Fungsional (Functional Data Model)

**- Model Lojik Data Bedasarkan Record (Record-Based Logical Models)** merupakan suatu model lojik data yang dtentukan berdasarkan record, yang terdiri dari:

* Model Relasional (Relational Model)
* Model Hirarkis (Hierarchical Model)
* Model Jaringan (Network Model)

1. **– Satu ke satu (One to one) =** Merupakan setiap entitas pada suatu entitas misal himpunan entitas A berhubungan dengan satu entitas dari himpunan entitas B, dan begitu juga sebaliknya entitas pada suatu himpunan entitas B berhubungan dengan satu entitas pada himpunan entitas A **.**

**– Satu ke banyak (One to Many) =** Merupakan setiap entitas pada suatu entitas misal himpunan entitas A berhubungan dengan banyak entitas dari himpunan entitas B, dan tidak sebaliknya dimana entitas pada suatu himpunan entitas B berhubungan dengan satu entitas pada himpunan entitas A **.**

**– Satu ke banyak (Many to one) =** Merupakan setiap entitas pada suatu entitas misal himpunan entitas A berhubungan dengan satu entitas dari himpunan entitas B, dan tidak sebaliknya dimana entitas pada suatu himpunan entitas B berhubungan dengan banyak entitas pada himpunan entitas A**.**

**– Satu ke banyak (One to Many) =** Merupakan setiap entitas pada suatu entitas misal himpunan entitas A berhubungan dengan banyak entitas dari himpunan entitas B, dan begitu juga sebaliknya dimana entitas pada suatu himpunan entitas B berhubungan dengan banyak entitas pada himpunan entitas A **.**

1. **Tiga syarat membuat tabel yang baik:**
2. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (Lossless-Join Decomposition).
3. Terpeliharanya ketergantungan fungsional pada saat perubahan data (Dependency Preservation)
4. Tidak melanggar Boyce-Code Normal Form (BCNF)

**Soal B**

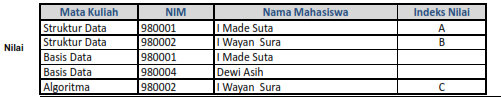
**Tabel – tabel Yang Harus Dinormalisasi**

1. **Tabel Nilai**

Untuk dapat menormalisasi tabel dibawah ini, maka diperlukan untuk menghapus salah satu field atau atribut yaitu Nama Mahasiswa, dikarenakan nama Mahasiswa sudah diwakilkan oleh attribut bernama **NIM** yang difungsukan sebagai kunci tamu atau foreign key pada tabel **Nilai** dan menjadi primary key pada tabel **Mahasiswa**.

Kemudian ganti nama attribut **Mata Kuliah** mejadi **Kode\_MataKuliah** karena **Kode\_Matakuliah** tersebut digunakan sebegai foreign key pada tabel **Nilai** dan menjadi primary key pada tabel **Kuliah**

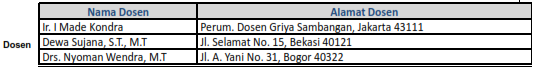
Dikarenakan dalam tabel **Nilai** tersebut belum terdapat primary key maka saya akan membuat satu attribut dengan nama **Kode\_Nilai** untuk dapat mewakili tabel Nilai tersebut yang difungsikan menjadi primary key, karena setiap tabel diperlukan adanya primary key.



Kesimpulan dari tabel diatas :

* Tabel nilai tersebut merepresentasikan antara data Kuliah dengan data Mahasiswa yang bertujuan untuk matakuliah yang diambil oleh Mahasiswa dan nilai yang didapatkan oleh masing – masing mahasiswa.
* Setiap orang mahasiswa dapat untuk tidak mengambil matakuliah apapun atau dapar memilih lebih dari satu matakuliah

1. **Tabel Dosen**

Pada tabel Dosen kita akan menormalisasinya yaitu menambahkan satu attribute dengan nama **nip** yang akan difungsikan sebagai primary key pada tabel **Dosen** 

Kesimpulan tabel diatas: Tabel diatas difungsikan untuk menampung data dosen baik nama maupun alamat dosen yang besangkutan dan diwakili dengan **nip** sebagai primary key yang akan dibuat.

1. **Tabel Jadual**

Untuk dapat menormalisasi tabel dibawah ini, maka kita diperlukan untuk mengubah **Nama Dosen** menjadi **NIP,** karena attribute NIP ini digunakan sebagai foreign key pada tabel **Jadual** dan menjadi primary key pada tabel **Dosen.**

Kemudian ganti nama attribut **Mata Kuliah** mejadi **Kode\_MataKuliah** karena **Kode\_Matakuliah** tersebut digunakan sebegai foreign key pada tabel **Nilai** dan menjadi primary key pada tabel **Kuliah**

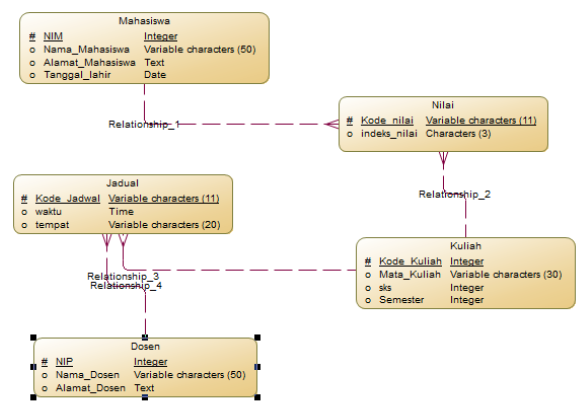
Dikarenakan dalam tabel **jadual** tersebut belum terdapat primary key maka saya akan membuat satu attribut dengan nama **Kode\_Jadwal** untuk dapat mewakili tabel Jadwal tersebut yang difungsikan menjadi primary key, karena setiap tabel diperlukan adanya primary key.



Kesimpulan :

* Tabel jadwal tersebut difungsikan untuk menampung data jadwal
* Untuk memanggil data dosen yang telah direlasikan
* Untuk memanggil data Kuliah yang telah direlasikan

**CDM Dari Hasil Normalisasi Diatas**



**PDM Dari Hasil Normalisasi Diatas**

