BASISDATA

(Teori dan Praktek dengan Query SQL)

Penerbit UNIPMA Press
Universitas PGRI Madiun
Jl. Setiabudi No. 85 Madiun Jawa Timur 63118
E-Mail: upress@unipma.ac.id
Website: kwu.unipma.ac.id

ISBN 978-623-6318-48-5

BASIS DATA (Teori dan Praktek dengan Query SQL)

HANI ATUN MUMTAHANA

UNIPMA Press
WE GOT IT

BASISDATA

(Teori dan Praktek dengan Query SQL)

HANI ATUN MUMTAHANA

Basis Data:
Teori dan Praktek dengan Query SQL

Basis Data:

Teori dan Praktek dengan Query SQL

Hani Atun Mumtahana, S.Kom.,M.Kom

UNIPMA Press
WE GOT IT

Basis Data: Teori dan Praktek dengan Query SQL

Penulis:
Hani Atun Mumtahana, S.Kom.,M.Kom

Perancang Sampul:
Dimas Setiawan, S.Kom.,M.Kom

Penata Letak:
Tim Kreatif UNIPMA Press

Cetakan Pertama, Desember 2021

Diterbitkan Oleh:
UNIPMA Press Universitas PGRI Madiun
JI. Setiabudi No. 85 Madiun Jawa Timur 63118
E-Mail: upress@unipma.ac.id
Website: kwu.unipma.ac.id
Anggota IKAPI: No. 207/Anggota Luar Biasa/JTI/2018

ISBN: 978-623-6318-48-5

Prakata

Alhamdulillah segala puji atas nikmat syukur, rahmat dan karunia Allah SWT dan dukungan dari pihak terkait sehingga buku "**Basisdata dengan Query SQL**" ini dapat terselesaikan. Buku ini merupakan buku pertama saya yang membahas tentang Basisdata dengan menggunakan Query SQL.

Buku ini diharapkan dapat menjadi pegangan bagi pembaca dari berbagai kalangan (pelajar, mahasiswa, pengajar maupun praktisi) dalam mengolah data sehingga dapat menyajikan informasi sesuai dengan kebutuhan. Dalam buku ini akan dibahas mengenai pengenalan Basisdata, bagaimana melakukan analisa kebutuhan data pada sebuah kasus, bagaimana melakukan pendefinisian struktur data dengan menggunakan *Database Management System* (DBMS) MySQL, dan bagaimana melakukan manipulasi data dengan menggunakan Query SQL.

Dalam buku ini telah dijelaskan secara detail berbagai fungsi yang dapat digunakan dalam melakukan pengolahan data, sehingga pembaca mampu melakukan pengolahan data dengan menggunakan fungsi. Selain melakukan pengolahan data dengan menggunakan berbagai fungsi, dalam buku ini telah dijelaskan penggunaan operator, seperti operator aritmatika, operator pembanging, operator pengolah kata dan operator logika.

Pada prinsipnya suatu data dapat diolah menjadi informasi dengan melakukan pengambilan data dari beberapa tabel yang ada dalam database. Konsep pengambilan data dari beberapa tabel dijelaskan pada teori Relasi Database pada buku ini. Dengan memahami konsep relasi yang dijelaskan dalam buku ini, diharapkan pembaca mampu melakukan pengambilan data dan menampilkan data yang saling berelasi antar tabel.

Terimakasih disampaikan kepada seluruh pihak terkait. Pihak UNIPMA yang telah memberikan dukungan penuh dalam penyelesaian buku ini. Kesempurnaan dalam buku ini merupakan tujuan dari penulis. Namun dalam buku ini masih terdapat banyak kekurangan yang perlu diperbaiki. Oleh karena itu penulis sangat mengharapkan adanya masukan dari berbagai pihak sehingga dapat dijadikan masukan bagi penulis untuk melakukan perbaikan dalam penyusunan buku Basisdata dengan Query SQL bagian-2. Semoga buku ini mampu memberikan manfaat bagi pihak pembaca dan menjadikan pegangan dalam melakukan pengolahan data sehingga mampu menghasilkan informasi yang lebih baik.

Penulis

Daftar Isi

Prakata v
Daftar Isi vii

BAB 1 - Konsep Basisdata 1
1. Data, Basisdata, Database Manajemen System dan Sistem Basisdata 1
2. Tujuan Basisdata 3

BAB 2 - Basisdata Relational 4
1. Entitas 5
2. Atribut 5
3. Relasi 9
4. Spesialisasi, Generalisasi dan Agregasi 13
5. Contoh Relasi Basisdata 14

BAB 3 - DBMS MySQL 19
1. Instalasi MySQL 19
2. Menjalankan MySQL dengan Command Prompt 21
3. Tipe data pada MySQL 23

BAB 4 - Structure Query Language 26

BAB 5 - Query SQL pada DATABASE dan TABLE 28
1. Membuat Database 28
2. Menampilkan Database 29
3. Memilih Database 29
5. Membuat Table 30
6. Membuat Auto Increment 33
7. Membuat table dari table lain 33
9. Menghapus table 40

BAB 6 - Manupulasi Data (Insert, Update dan Delete) **41**
1. INSERT (menambah data) 41
2. UPDATE (mengubah data) 50
3. DELETE (menghapus data) 55

BAB 7 - Mengolah data dengan SELECT **60**
1. SELECT sederhana 60

BAB 8 - Operator pada Database **75**
1. Operator Aritmatika 75
2. Operator Pembanding 79
3. Operator Logika 80
4. Operator BETWEEN 83
5. Operator LIKE 84

BAB 9 - Join Table pada Database **88**
1. Prinsip JOIN 88
2. IMPLISIT JOIN 88
3. INNER JOIN 90
4. OUTER JOIN 91
5. JOIN untuk banyak Table 96
6. JOIN dengan berbagai Kondisi 97

BAB 10 - Ekspresi Logika **99**
1. Ekspresi Logika IF 99
2. Logika CASE 104

BAB 11 - Mengenal Database Relational dan Database NoSQL **108**
1. MongoDB 109
2. Redis 109
3. Cassandra 109
4. Oracle NoSQL 109

Daftar Pustaka 111
Biografi Penulis 112

BAB 1
Konsep Basisdata

1. Data, Basisdata, Database Manajemen System dan Sistem Basisdata

Pengembangan aplikasi berbasis Teknologi Informasi dan Sistem Informasi tidak lepas dari peran pengolahan data. Pengolahan data yang baik merupakan salah satu penunjang pemberian informasi yang tepat kepada pengguna. Hal ini tidak lepas dari analisa kebutuhan yang tepat, desain data yang sesuai, implementasi yang baik dan pemilihan aplikasi penunjang yang sesuai dengan kapasitas dan kebutuhan.

Pengolahan data dalam penunjang implementasi aplikasi sehingga mampu menjadi sebuah Sistem Informasi merupakan penerapan dari basisdata. Selanjutnya hasil penerapan pengolahan data akan ditransformasikan dalam sebuah aplikasi pengolah data yang sering disebut dengan *database manajemen system* (DBMS). Suatu konsep desain basisdata yang telah ditransformasikan dalam DBMS dan telah digunakan oleh pengguna baru bisa dikatakan dengan Sistem Basisdata. Lebih jelasnya berikut adalah pengertiannya :

a. Data

Pengertian adalah suatu keterangan yang benar dan nyata. Data adalah keterangan atau bahan nyata yang dapat dijadikan dasar kajian (analisis atau kesimpulan). Pada dasarnya data merupakan suatu fakta yang dapat berupa objek atau kejadian yang ada.

b. Basisdata

Secara konsep basisdata merupakan suatu teknik pengumpulan data yang dapat disimpan dan diolah sehingga menghasilkan informasi. Data tersebut merupakan sebuah fakta yang dapat berupa objek atau suatu kejadian yang ada.

Adapun pengertian basisdata menurut para ahli :

(Oracle, n.d.) *A database is an organized collection of structured information, or data, typically stored electronically in a computer system*

(Date, 2003). *A databse is a collection of persistent data that is used by the aplication system of some given enterprise.*

(Raharjo, 2013), *"Database atau basisdata adalah kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat".*

(Kusrini, 2007), *Basisdata adalah kumpulan data yang saling berelasi*

(Jogiyanto, 2005), *Basisdata merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya tersimpan dan di simpanan luar komputer dan digunakan oleh perangkat lunak tertentu untuk memanipulasinya*

Dapat disimpulkan bahwa pengertian basisdata merupakan sekumpulan dalam suatu kelompok data yang saling berhubungan satu dengan lainnya dan dapat disimpan, diolah dengan menggunakan aplikasi sehingga mampu menghasilkan informasi.

Sebagai contoh dalam sebuah kegiatan AKADEMIK memiliki objek/kejadian berupa fakta antara lain :

a) Adanya MAHASISWA sebagai objek
b) Adanya kegiatan KRS sebagai suatu kejadian
c) Adanya DOSEN sebagai objek
d) Adanya JADWAL KULIAH sebagai suatu kejadian
e) Adanya MATAKULIAH sebagai objek
f) Adanya NILAI sebagai suatu kejadian yang muncul karena adanya proses PERKULIAHAN

Dari contoh diatas, dapat digambarkan, bahwa basisdata meruapakan kumpulan dari objek/kejadian dalam suatu kelompok kegiatan. Objek/kejadian yang ada akan saling terhubung/relasi untuk menghasilkan suatu informasi. Contohnya :

a) Adanya hubungan antara MAHASISWA yang mengambil MATAKULIAH sehingga akan menghasilkan NILAI.
b) Adanya DOSEN yang mengajar MATAKULIAH sehingga akan menjadi suatu JADWAL KULIAH

Hubungan/relasi yang ada dapat digambarkan anatar objek dengan objek, objek dengan kejadian atau kejadian dengan kejadian. Penjelasan tentang hubungan/relasi antar data dalam sebuah basisdata akan dijelaskan pada bab 4

c. *Database Manajemen System (DBMS)*

Dalam mengolah data dan menyimpan data diperlukan suatu aplikasi penunjang. Aplikasi yang dirancang untuk melakukan definisi, membangun, melakukan manipulasi dan membagikan suatu data sering disebut dengan *Database Manajemen System* (DBMS). Beberapa contoh DBMS yang paling banyak digunakan antara *lain MySQL, MS SQL Server, Oracle, Firebird, Database Desktop Paradox dan MS Access.*

Salah satu DBMS yang paling banyak diguanakan secara standart adalah MySQL. Secara singkat MySQL diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia yang diberi nama UNIREG. Dalam perkembangannya, pada tahun 1994 UNIREG dianggap sudah relevan dengan kebutuhan aplikasi berbasis WEB,

sehingga pada tahun 1995 sebuah RDBMS bernama MySQL dirilis. Sampai dengan saat ini, pengguna DBMS MySQL terdiri dari berbagai kalangan. Hal ini karena MySQL memiliki banyak kelebihan yang mampu dimanfaatkan oleh kalangan pelajar, pengembang pemula, dan para developer PHP.

d. Sistem Basisdata

Sebuah basisdata yang telah di transformasikan dalam DBMS selanjutnya ada pengguna yang telah memanfaatkannya disebut dengan Sistem Basisdata/*Database System*. Terdapat beberapa komponen dasi Sistem Basisdata antara lain :

a) Perangkat Keras
b) Sistem Operasi
c) Basisdata
d) Aplikasi DBMS
e) Jaringan

Berdasarkan penjelasan Basisdata, DBMS dan Sistem Basisdata tersebut di atas, diharapkan mampu memberikan gambaran perbedaan dari istilah tersebut. Seringkali terdapat kesalahan pemahaman dari istilah tersebut.

2. Tujuan Basisdata

Tujuan dari Basisdata antara lain :

a. Kecepatan dan kemudahan dalam penyimpanan, pengolahan dan mendapatkan informasi yang dibutuhkan
b. Efisiensi *space,* dengan basisdata dapat mengurangi adanya redudansi data sehingga dapat menghemat penyimpanan
c. Keakuratan mendapatkan data, karena dalam basisdata setiap data yang tersimpan diberikan suatu kode unik untuk dapat membedakan isian data.
d. Ketersediaan data dapat disesuaikan dengan kebutuhan penyajian informasi, sehingga data yang sudah tidak digunakan dalam periode yang telah ditentukan dapat dipindahkan pada penyimpanan berkas (*backup*). Hal ini bertujuan untuk mengurangi beban penyimpanan dan kecepatan ketika akses data.
e. Kelengkapan, secara fleksibel dapat dilakukan perubahan struktur basisdata, penambahan isian data, penambahan kolom guna memenuhi kebutuhan informasi bagi pengguna
f. Keamanan basisdata dapat dilakukan dengan pemberian akses yang berbeda pada pengguna. Langkah ini merupakan langkah dasar yang sering digunakan pengembang untuk meningkatkan keamanan data.
g. Kebersamaan pengolahan data dilakukan untuk memberikan kemudahan akses data yang dilakukan secara *multiuser.*

BAB 2
Basisdata Relational

Dalam perkembangannya implementasi basisdata tidak lepas dari hasil desain relasi yang telah dibuat. Terdapat beberapa model untuk membuat desain basisdata, antara lain model data flat, hirarki, jaringan dan relasional. Model data Relasional atau sering disebut dengan RDBMS (*Relational Database Manajemen System*) merupakan salah satu model desain basisdata yang paling banyak digunakan oleh para pengembang. Pada perlajanannya dan menyesuaikan kebutuhan kapabilitas dalam pengolahan data dan informasi munculah model database NoSQL pada pertengahan tahun 2000. Namun pada buku ini akan dibahas secara rinci pemanfaatan basisdata untuk pengembangan aplikasi dengan model data relasional.

Model data relasional adalah kumpulan dari data yang saling terhubung antara data satu dengan yang lainnya sehingga memberikan kemudahan untuk akses bagi penggunanya. Model data relasional merupakan model data yang paling banyak digunakan oleh para pengembang aplikasi mulai dari pemula, untuk pembelajaran sampai dengan pengembang profesional. Model data relasional pertama kali dirumuskan pada tahun 1969 oleh **Edgar F. Codd** yang bertujuan untuk menyediakan metode deklaratif dalam merepresentasikan data dan query sehingga pengguna basisdata dapat memperoleh informasi apa yang dibutuhkan (https://id.wikipedia.org/wiki/Model_relasional). Beberapa keunggunal model data relasional antara lain (https://searchdatamanagement.techtarget.com/definition/relational-database) :

1. Pengelompokan data yang mudah : data yang diolah dapat dikelompokkan sesuai dengan kebutuhannya dan dapat diolah dengan bahasa sql sehingga mampu menyajikan kebutuhan informasi sebagai salah satu penunjang pengambilan keputusan bagi manajemen. Selain itu proses integrasi data dengan data lainnya pada setiap aplikasi dapat diimplementasikan dengan mudah.
2. Model basisdata relasional meminimalkan danya duplikat data
3. Model basisdata relational ini mudah digunakan dan difahami oleh semua kalangan.
4. Mampu digunakan oleh *multiuser* dengan menekankan nilai relasi antar data sehingga banyak pengguna mampu melakukan akses pada data yang sama
5. Tingkat keamanan tinggi dengan pembatasan akses yang akan diberikan pada pengguna.

Penggambaran model relasional biasa disebut dengan *Entity Relationship Diagram* (ERD). Diagram ER merupakan model konseptual desain basisdata yang akan menggambarkan struktur logis dari suatu basisdata dalam bentuk grafis. Dalam membuat ERD terdapat beberapa komponen antara lain :

1. Entitas

Entitas merupakan gambaran dari data yang dapat berupa objek atau kejadian sesuai dengan fakta yang ada yang dapat dibedakan dengan data yang lain. Setiap entitas dapat direlasikan dengan entitas lain. Saat diterapkan dalam DBMS, entitas tersebut akan ditransformasikan ke dalam bentuk TABEL.

Sebagai contoh dalam pengembangan Aplikasi PENILAIAN HASIL BELAJAR MAHASISWA mendeskripsikan entitas dalam desain basisdata sebagai berikut :

- MAHASISWA merupakan objek yang akan melakukan evaluasi pembelajaran
- MATAKULIAH merupakan objek kegiatan yang ditempuh oleh mahasiswa
- NILAI merupakan suatu kejadian proses evaluasi yang dilakukan oleh mahasiswa dari kegiatan yang diikuti

Pada model basisdata relasional entitas dapat dibedakan menjadi entitas kuat (*strong entity*) dan entitas lemah (*weak entity*).

a. *Strong Entity* : entitas yang dapat berdiri sendiri dan tidak bergantung pada entitas lain.

   Contoh : Entitas MAHASISWA, Entitas DOSEN, Entitas MATAKULIAH, Entitas PROGRAM_STUDI

   Simbol *Strong Entity* adalah 

b. *Weak Entity* : entitas yang tidak dapat berdiri sendiri dan keberadaannya bergantung pada entitas lain.

   Contoh :

   - Entitas ORANG_TUAS_MHS yang bergantung pada adanya Entitas MAHASISWA
   - Entitas ASISTEN_DOSEN yang bergantung pada adanya Entitas DOSEN
   - Entitas NILAI yang bergantung pada adanya Entitas MAHASISWA, Entitas MATAKULIAH

   Simbol *Weak Entity* adalah 

2. Atribut

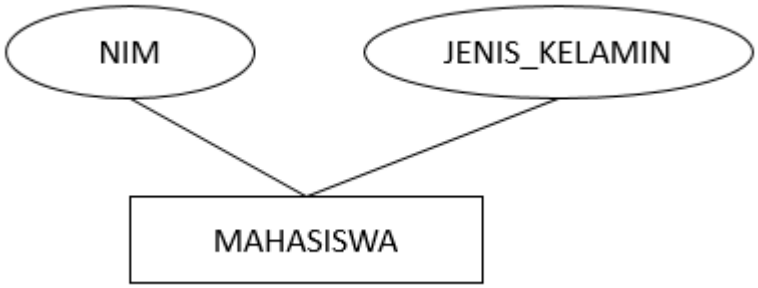
Karakteristik dari suatu entitas sehingga informasi dari entitas tersebut dapat dijelaskan dalam bentuk atribut. Atribut pada desain basisdata akan ditrasnformasikan dalam bentuk *FIELD*/KOLOM pada sebuah TABEL. Setiap atribut harus dapat didefinisikan tipe data yang akan digunakan, berapa panjang karakter yang harus disediakan, bagaimana sifat yang akan

dideskripsikan sehingga ketika ditrasnformasikan dalam bentuk *FIELD*/KOLOM dapat menyimpan informasi yang di-*input* sesuai dengan fakta.

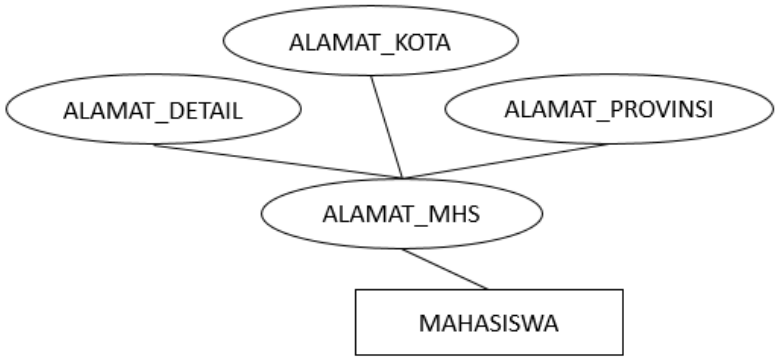
Sebagai contoh pada entitas MAHASISWA akan memiliki atribut NIM, NAMA_MHS, TEMPAT_LAHIR_MHS, TANGGAL_LAHIR_MHS, JENIS_KELAMIN, ALAMAT_MHS, NO_HP/TLP. Setiap atribut pada entitas MAHASISWA tersebut akan dapat menyimpan informasi tentang data mahasiswa.

Beberapa jenis atribut sebagai berikut :

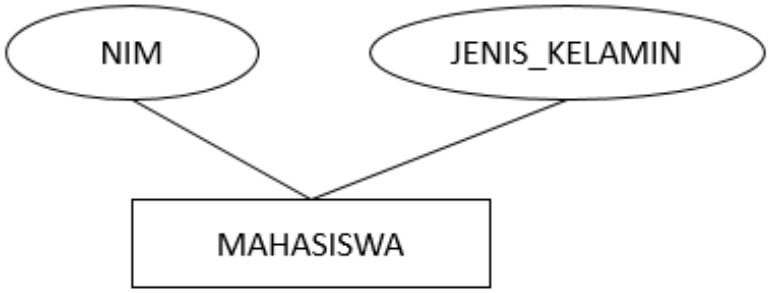
a. Atribut sederhana : atribut yang tidak dapat dipilah kembali menjadi sub atribut. Sebagai contoh atribut NIM, JENIS_KELAMIN, NAMA_MHS pada entitas MAHASISWA sudah tidak dapat dibagi lagi menjadi atribut lain.



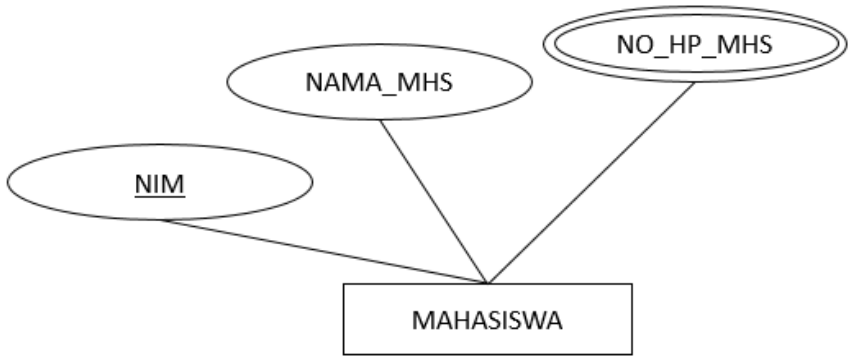
b. Atribut komposit : atribut yang dapat dipilah menjadi beberapa atribut. Sebagai contoh atribut ALAMAT_MHS dapat diuraikan menjadi beberapa atribut lagi yaitu ALAMAT_DETAIL, ALAMAT_KOTA, ALAMAT_PROVINSI



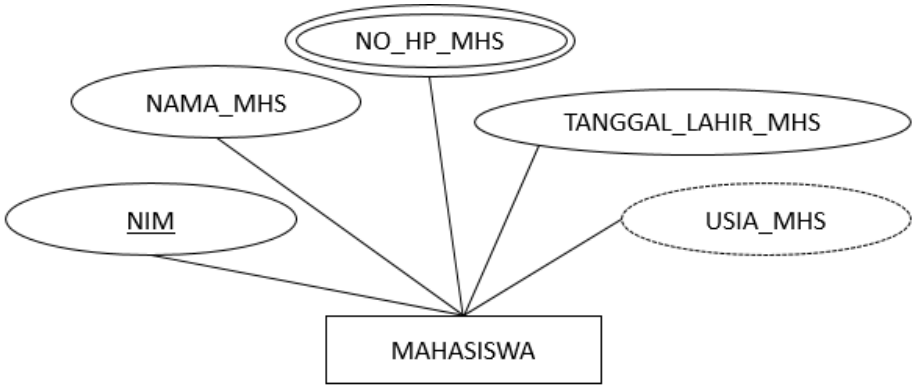
c. Atribut *Single Value* : atribut yang hanya memiliki satu nilai informasi yang akan disimpan. Contoh pada entitas MAHASISWA dengan atribut JENIS_KELAMIN hanya akan memiliki satu nilai Laki-laki atau Perempuan, tidak mungkin ada mahasiswa yang akan memiliki dua jenis kelamin.



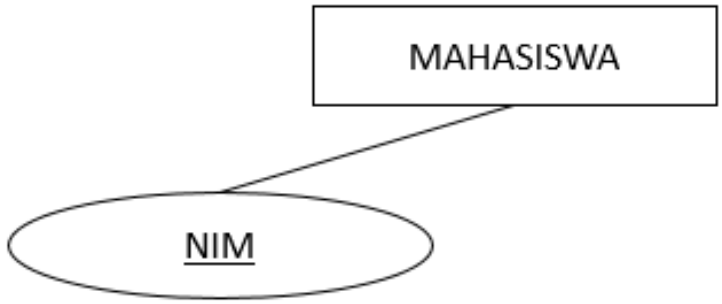
d. Atribut *Multi Value* : atribut yang dapat memiliki nilai lebih dari satu untuk informasi yang akan disimpan. Contoh pada atribut NO_HP adakalanya seorang mahasiswa akan memiliki lebih dari satu nomor handphone. Keberdaan atribut *multi value* ini sangat tidak disarankan. Jika dalam membuat desain basisdata dan ditemukan adanya atribut *multi value* maka perlu dilakukan proses NORMALISASI pada entitas/tabel tersebut. Biasanya, keberadaan suatu atribut yang memiliki *nilai multi value* akan terlihat saat ditransformasikan dalam bentu tabel yang sudah diisi data.



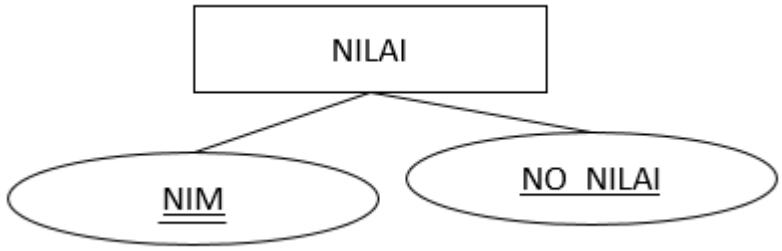
e. Atribut Derivatif : atribut yang dihasilkan atau diturunkan dari atribut lain. Contoh pada entitas MAHASISWA terdapat atribut TANGGAL_LAHIR_MHS, dari tanggal lahir yang sudah dituliskan dapat diketahui usia mahasiswa dengan melakukan pengolahan data. Keberadaan atribut derivatif pada suatu entitas tidak wajib untuk dituliskan dalam pembuatan desain basisdata. Informasi pada atribut derivatif akan muncul ketika dilakukan manipulasi data dengan menggunakan *Query SQL*.



f. *Primary Key* : atribut yang dipilih menjadi kunci pada suatu entitas. Sifat dari *primary key* akan menjadi pembeda informasi dari satu baris dengan baris yang lainnya. Suatu atribut *primary key* harus memiliki nilai UNIK sehingga tidak dapat diisi dengan nilai yang sama. Selain itu data pada atribut *primary key* tidak boleh kosong/NULL. Contoh pada entitas MAHASISWA dapat mendefinisikan NIM sebagai *primary key*. NIM seorang mahasiswa tidak mungkin sama dengan mahasiswa yang lain, selain itu seorang yang terdaftar sebagai mahasiswa sudah tentu atau wajib memiliki NIM. Simbol penanda bahwa suatu atribut merupakan *primary key* adalah dengan memberi garis bawah pada nama atribut.



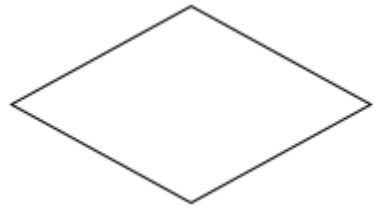
g. *Foreign Key* : atribut *primary key* pada suatu entitas yang menduduki kunci asing atau kunci tamu pada entitas lainnya. Sifat dari *foreign key* adalah menjadi atribut penghubung dari relasi entitas. Contoh pada hubungan antara entitas MAHASISWA dengan entitas NILAI, dimana pada entitas NILAI membutuhkan satu atribut kunci dari entitas MAHASISWA yang akan merujuk informasi NILAI YANG DIPEROLEH MAHASISWA. Simbol penanda jika suatu atribut adalah *foreign key* dengan garis bawah dua



3. Relasi

Relasi merupakan gambara hubungan dari satu entitas, dua entitas atau lebih. Relasi dilakukan untuk menghubungkan antara ENTITAS dengan ENTITAS sehingga mampu menghasilkan adanya aliran data yang sesuai dengan kebutuhan informasi. Hasil dari relasi dapt digunakan untuk menampilkan data lebih dari satu tabel pada suatu basisdata.

Suatu relasi dalam sebuah desain basisdata dengan model ERD biasanya ditandai dengan simbol jajaran genjang yang diberi suatu kata penghubung berupa kata kerja untuk mengatahui adanya hubungan antar ENTITAS. Adapun simbol relasi sebagai berikut :



Dalam mambuat desain relasi antar ENTITAS terdapat beberapa aturan yang perlu digunakan. Aturan tersebut memberikan batasan tentang penentuan DERAJAT RELASI dan CARDINALITAS dalam membuat desain relasi.

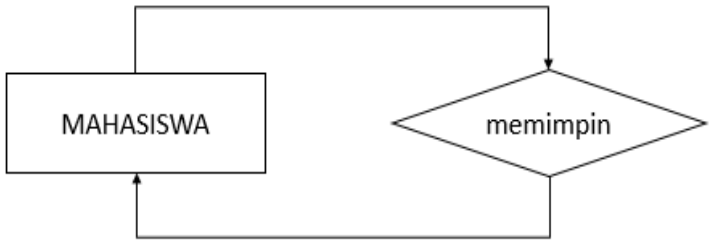
a. DERAJAT RELASI

Terdapat tiga jenis derajar relasi yang dapat digunakan dalam membuat desain relasi antar ERNTITAS, antara lain :

1) UNARY

Unary merupakan suatu bentuk relasi yang menghubungkan satu ENTITAS saja, dimana suatu ENTITAS akan berelasi dengan ENTITAS itu sendiri. Contoh :

Pada suatu kelas mahasiswa, akan ditunjuk seorang mahasiswa untuk menjadi ketua kelas.



2) BINARY

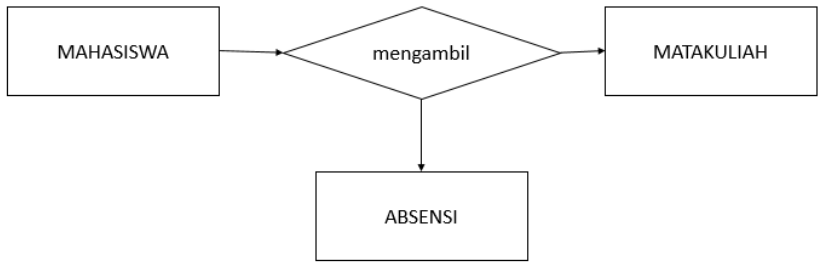
Binary merupak suatu bentuk relasi yang menghubungkan antara satu ENTITAS dengan ENTITAS lain, atau hubungan antara dua ENTITAS. Contoh :

Seorang mahasiswa akan mendapatkan nilai.

3) TERNARY

Ternary merupakan suatu bentuk relasi yang menghubungkan antara tiga ENTITAS sekaligus. Contoh :

Seorang mahasiswa yang mengambil matakualiah pasti akan menemukan datanya pada saat absensi di matakuliah tersebut.



b. **KARDINALITAS**

Dalam membuat relasi antar ENTITAS perlu dilakukan analisa aliran data yang terjadi dari hubungan tersebut. Analisa tersebut dapat digambarkan dengan mendeskripsikan kardinalitas pada suatu hubungan antar ENTITAS. Kardinalitas merupakan suatu gambaran banyaknya jumlah maksimum entitas yang dapat berelasi dengan entitas lain pada suatu himpunan entitas. Sebagai contoh :

- Seorang mahasiswa dapat mengambil/memprogram matakuliah hanya satu matakuliah atau lebih dari satu matakuliah.
- Seorang dosen dapat mengampu matakuliah hanya satu atau lebih dari satu
- Seorang mahasiswa dapat menemukan namanya pada absensi hanya pada satu matakuliah atau lebih dari satu matakuliah.

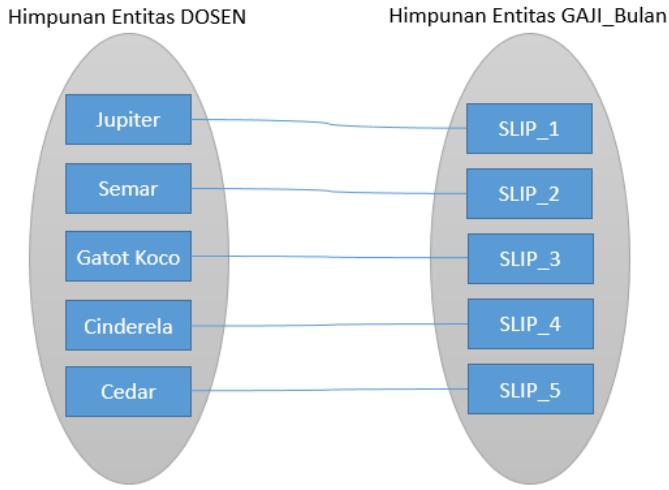
Beberapa jenis kardinalitas antaralain :

1) One To One

Setiap entitas pada suatu himpunan entitas dapat berelasi paling banyak dengan satu entitas pada himpunan entitas lainnya dan sebaliknya.

Contoh :

- Jupiter adalah seorang dosen, dimana setiap bulannya Jupiter dan dosen lainnya akan menerima satu slip gaji, begitupun sebaliknya satu slip gaji akan diterima oleh setiap dosen pada setiap bulannya.



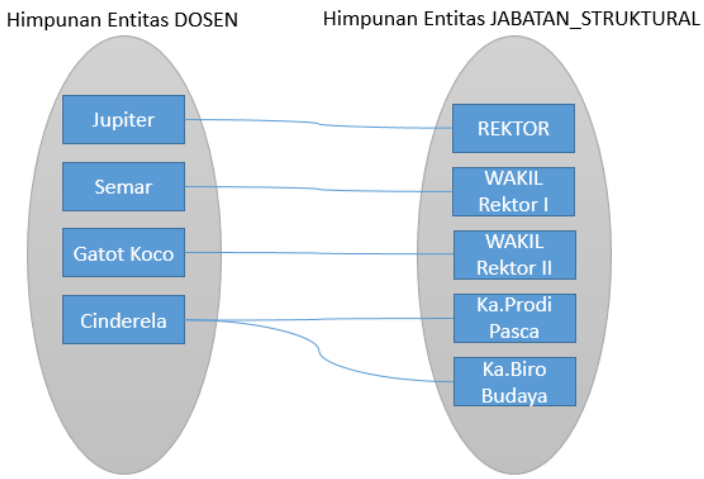
Contoh relasi dengan Kardinalitas One To One

2) One To Many

Setiap entitas pada suatu himpunan entitas dapat berelasi dengan banyak entitas pada himpunan entitas lainnya dan tidak sebaliknya.

Contoh :

- Jupiter adalah seorang dosen yang memiliki Jabatan Struktural sebagai REKTOR, setiap dosen dapat memiliki lebih dari satu jabatan struktural seperti Cinderela yang menjabat sebagai Ka.Prodi Pasca dan Ka.Biro Budaya. Namun tidak sebaliknya setiap satu jabatan struktural hanya dapat dimiliki oleh maksimal satu orang dosen.



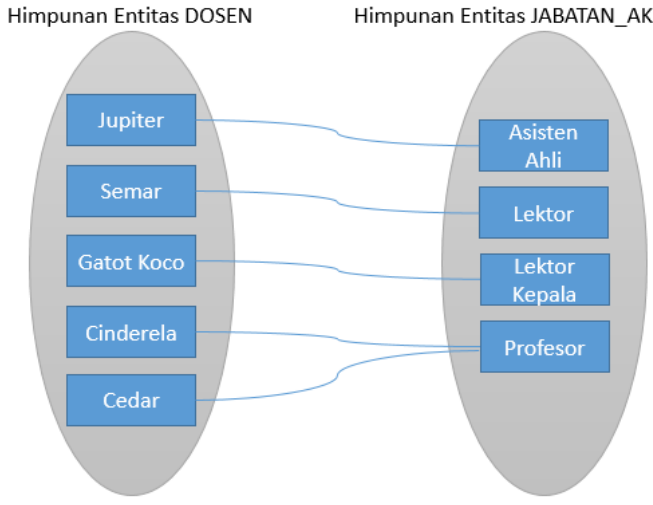
Contoh relasi dengan Kardinalitas One To Many

3) Many To One

Setiap entitas pada suatu himpunan entitas dapat berelasi dengan paling banyak satu entitas pada himpunan entitas dan tidak sebaliknya.

Contoh :

- Jupiter adalah seorang dosen, dimana setiap dosen hanya boleh memiliki satu jabatan akademik



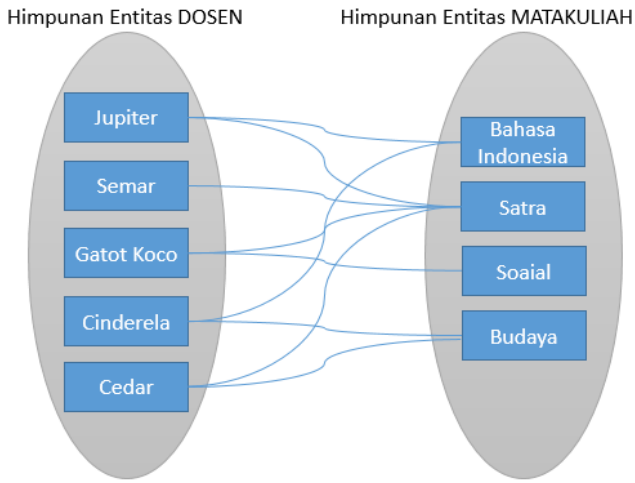
Contoh relasi dengan Kardinalitas Many To One

4) Many To Many

Setiap entitas pada suatu himpunan entitas dapat berelasi dengan banyak entitas pada himpunan entitas lainnya dan sebaliknya.

Contoh :

- Jupiter adalah seorang dosen, Jupiter dapat mengampu matakuliah Bahasa Indonesia dan Sastra, begitupun sebaliknya matakuliah Sastra dapat diampu oleh Semar dan Gatot Koco.



Contoh relasi dengan Kardinalitas Many To Many

c. NILAI KARDINALITAS MINIMUM

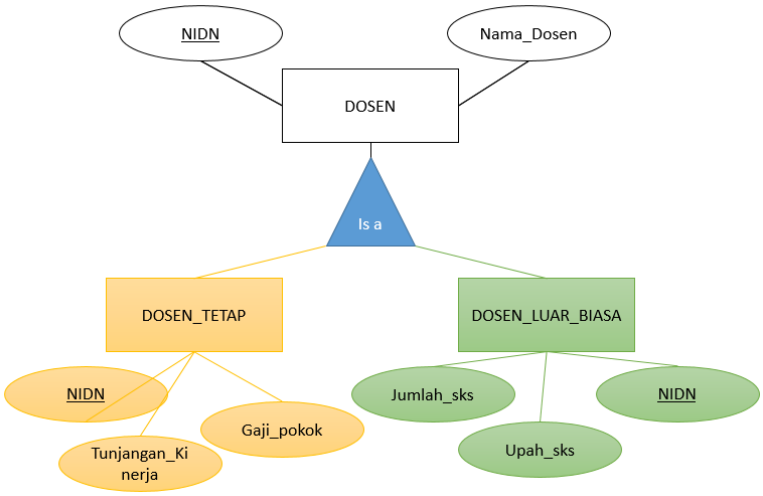
Telah dibahas bahwa kardinalitas merupakan suatu gambaran banyaknya jumlah maksimum entitas yang dapat berelasi dengan entitas lain pada suatu himpunan entitas. Pada contoh : Seorang mahasiswa dapat mengambil/memprogram matakuliah hanya satu matakuliah atau maksimal lebih dari satu matakuliah dan begitupun sebaliknya setiap matakuliah dapat dipilih oleh seorang mahasiswa atau maksimal lebih dari satu mahasiswa. Namun pada konteksnya akan terjadi bahwa seorang mahasiswa tidak akan mengambil/memprogram satu matakuliah pun (dikarenakan sedang mengambil cuti perkuliahan) atau sebaliknya sebuah matakuliah tidak akan dipilih oleh seorang mahasiswa (dikarenakan tergolong matakuliah pilihan). Berdasarkan hal tersebut terdapat nilai kardinalitas minimun untuh hubungan yang terjadi.

4. Spesialisasi, Generalisasi dan Agregasi

Spesialisasi merupakan suatu teknik desain relasi basisdata secara ***top-down*** dengan melakukan desain *supgrouping* dari himpunan entitas menjadi himpunan entitas yang berbeda. Tujuan dari spesialisai adalah memberikan gambaran konseptual tentang perbedaan karakteristik dari himpunan entitas tersebut. Konspe desain spesialisasi lebih mengarah pada penurunan

(*inheritance*) himpunan entitas menjadi himpunan entitas yang memiliki level lebih rendah dan memiliki atribut tersendiri. Simbol spesialisasi dinotasikan dengan simbol segitiga yang berlabe "Is a".

Sebagai contoh seorang dosen yang terdaftar sebagai pegawai pada suatu institusi pasti memiliki status Dosen Tetap atau Dosen Luar Biasa, dan dari pembeda tersebut terdapat beberapa atribut yang akan menjadi pembeda dari himpunan entitas yang diturunkan.



Generalisasi merupakan suatu teknik desain relasi basisdata secara ***bottom-up*** dengan melakukan desain *supgrouping* dari himpunan entitas yang lebih rendah menjadi himpunan entitas pada level yang lebih tinggi. Generalisasi merupakan kebalikan dari Spesialisasi. Dalam melakukan manipulasi data, seringkali manipulasi pada entitas level tinggi tidak disimpan, namun digunakan **View** untuk menampilkan data.

Agregasi merupakan suatu teknik desain relasi basisdata yang menggambarkan hubungan antar himpunan entitas dengan himpunan relasi. Agregasi akan menghasilkan suatu entitas baru yang dapat direlasikan dengan entitas lainnya. Sebagai contoh setiap dosen dapat mengampu matakuliah satu atau lebih dari satu demikian sebaliknya, satu matakuliah dapat diampu satu atau lebih dari satu dosen. Dari desain tersebut akan menghasilkan suatu himpunan entitas baru yang dapat direlasikan dengan entitas lain.

5. Contoh Relasi Basisdata

Pada pembahasan ini akan dijelaskan pembuatan desain relasi basisdata dengan menggunakan model *Entity Relationship Diagram* (ER-D). Contoh desain yang akan dibuat berdasarkan pertanyaan untuk penyajian kebutuhan informasi dari hasil pengolahan data.

Contoh kali ini berdasarkan dari kebutuhan penyajian informasi pada Sistem Informasi Manajemen suatu Intitusi Perguruan Tinggi. Pada aplikasi dibutuhkan informasi "***Siapa saja nama Mahasiswa yang mengambil Matakuliah Basisdata pada kelas 3A***".

Berdasarkan kebutuhan informasi tersebut akan dilakukan beberapa langkah antara lain :

a. Analisa kebutuhan data/entitas
b. Deskripsi atribut pada setiap entitas
c. Membuat hubungan antar entitas

Adapun hasil dari analisa yang dilakukan adalah sebagai berikut :

a. Analisa kebutuhan data/entitas

   Dari hasil analisa, adapun entitas yang dibutuhkan untuk menjawab pertanyaan tersebut adalah :

   Entitas Mahasiswa

   Entitas Matakuliah

   Entitas KRS

   Entitas Jadwal

b. Deskripsi atribut pada setiap entitas

   Hasil deskripsi kebutuhan atribut pada setiap entitas adalah sebagai berikut :

Entitas Mahasiswa	=	{nim, nama_mahasiswa, alamat_mahasiswa, tempat_lahir, tgl_lahir, no_tlp }
Entitas Matakuliah	=	{kode_matkul, nama_matkul, semester, sks}
Entitas MHS_KRS	=	{id_krs}
Entitas Jadwal	=	{id_jadwal, kelas, hari, jam_ke, ruang}

c. Hubungan antar entitas

   Sebelum membuat ER-D, lakukanlah analisa untuk mendeteksi sebab dan akibat harus adanya relasi yang terjadi antar entitas.

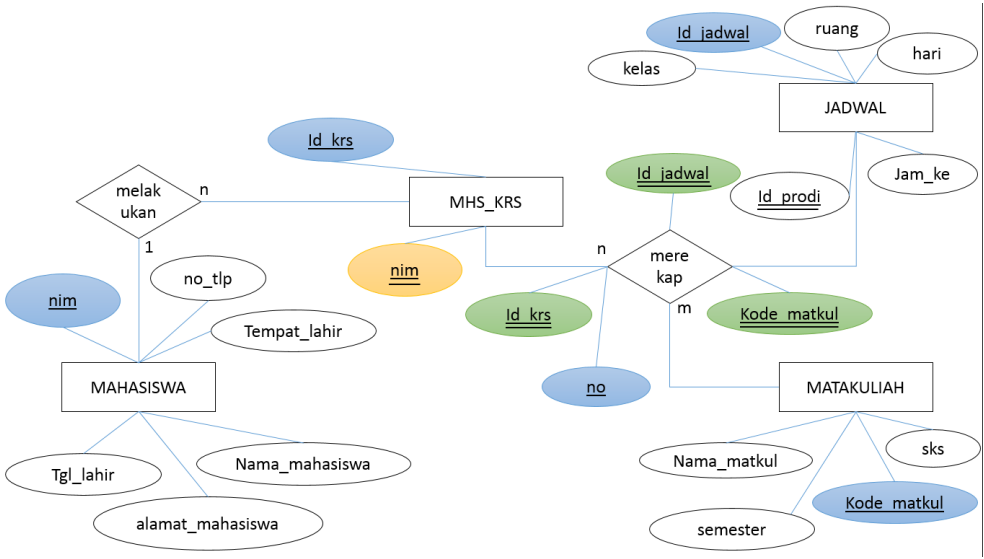
   a) Entitas MAHASISWA berelasi dengan MHS_KRS dengan kardinalitas 1 : n (one to many), dimana setiap MAHASISWA akan melakukan dan atau memiliki beberapa kali proses MHS_KRS selama studi, namun tidak sebaliknya.

   b) Entitas MHS_KRS berelasi dengan MATAKULIAH dengan kardinalitas n : m (many to many), dimana setiap proses KRS dapat memuat satu atau banyak MATAKULIAH dan sebaliknya, satu atau banyak MATAKULIAH dapat direkap dalam satu atau banyak proses KRS.

   c) Relasi antara MHS_KRS, MATAKULIAH dan JADWAL memiliki derajat relasi ternary, dimana dari hubungan ketiga entitas tersebut menghasilkan atribut yang

menjadi *primary foreign key* yaitu id_krs, id_jadwal dan kode_matkul. Relasi **merekap** dapat dijadikan entitas baru dan menambahkan satu atribut yang akan menjadi atribut kunci.

Selanjutnya lakukan tahapan analisa, apakah desain yang sudah dibuat dapat menjawab pertanyaan "***Siapa saja nama Mahasiswa yang mengambil Matakuliah Basisdata pada kelas 3A ?"***. Analisa dapat dilakukan dengan mensimulasikan hasil desain dalam bentuk tabel yang diisi data/record.



T.Mahasiswa

Nim	Nama_mahasiswa	Alamat_mahasiswa	Tempat_lahir	Tgl_lahir	No_tlp
210210001	Cinderella	Jl. Cinderawasih JKT	Madiun	2021-11-11	0879765xxx
210210002	Rapunzel	Jl. Apus Mojokerto	Jayapura	1999-12-10	081511876xxx
210210003	Zepot	Kelerang rt.07 Mejayan	Madiun	1998-10-21	08564665xxx

T. Matakuliah

Kode_matkul	Nama_matkul	Semester	sks
MKP1001	Basisdata	3	3
MKP1002	Algoritma Pemrograman	1	3
MKW1001	Sistem Enterprise	3	4

T. Jadwal

Id_jadwal	Ruang	Hari	Jam_ke	Kelas	Kode_prodi
J0001	G4.102	Senin	1	3A	SITA
J0002	G4.102	Senin	4	3B	IKOM
J0003	G4.102	Senin	7	3A	SITA

T.MHS_KRS

Id_krs	Nim
Krs210001	210210001
Krs210002	210210002
Krs210003	210210003

T.Detail_KRS

NO	Id_KRS	Id_jadwal	Kode_matkul
1	Krs210001	J0001	MKP1001
2	Krs21000	J0002	MKW1001

Dari Hasil simulasi tersebut dapat dilihat isian data pada tabel KRS dapat dihasilkan informasi ***Siapa mahasiswa yang melakukan KRS,*** dilihat dari NIM sebagai primary key pada tabel MAHASISWA yang muncul pada tabel KRS menjadi foreign key.

Prinsip dari adanya foreign key adalah untuk menampilkan data yang ada pada table mahasiswa pada saat dilakukan pemanggilan dengan table KRS. Contoh melakukan pemanggilan nama_mahasiswa dan alamat_mahasiswa yang telah melakukan KRS.

View T.KRS + T.Mahasiswa			
Id_krs	**Nim**	**Nama_mahasiswa**	**Alamat_mahasiswa**
Krs210001	210210001	Cinderella	Jl. Cinderawasih JKT
Krs210002	210210002	Rapunzel	Jl. Apus Mojokerto

Menampilkan informasi yang diambil dari beberapa tabel seperti pada contoh di atas, dapat dilakukan dengan manipulasi data menggunakan Query SQL yang akan dijelaskan pada bab selanjutnya. Dalam menampilkan informasi dari beberapa tabel perlu memperhatikan syarat wajib yaitu adanya satu atribut Primary key (PK) yang menjadi atribut Foreign Key (FK) pada tabel lain. Contoh di atas menjelaskan adanya NIM yang menjadi PK pada table MAHASISWA juga muncul sebagai FK pada tabel KRS atau sering dituliskan MAHASISWA.NIM = KRS.NIM dalam query pada bahasa SQL.

Catatan :
Ketika melakukan analisa untuk hubungan antar entitas, harus dipastikan terdapat adanya aliran data yang muncul. Hal ini dapat ditandai dengan munculnya atribut primary key pada suatu entitas yang menjadi atribut foreign key pada entitas lain

Lakukan validasi hasil desain relasi dengan mensimulasikan dalam bentuk tabel yang diisi dengan data/record.

Jika masih terjadi kesalahan tampilan informasi dari hasil simulasi, maka lakukan perbaikan desain relasi
Kesalahan yang sering terjadi yaitu :

1. *Adanya kolom yang memiliki nilai multivalue sehingga perlu dilakukan proses normalisasi*
2. *Adanya penempatan kunci tamu (foreign key) yang kurang tepat*

BAB 3
DBMS MySQL

Pada bab sebelumnya telah dijelaskan banyak sekali DBMS yang digunakan oleh para pengembang perangkat lunak. Pada buku ini akan dibahas implementasi hasil desain pada DBMS MySQL. MySQL merupakan salah satu DBMS yang sangat terkenal di dunia. MySQL banyak digunakan oleh berbagai kalangan khususnya para pengembang aplikasi berbasis web.

Setiap pembahasan pada buku ini dapat langsung dipraktekkan dengan MySQL. Kita dapat mendownload *MySQL Community Edition* secara gratis di website www.mysql.com dan melakukan instalasi pada perangkat.

1. **Instalasi MySQL**

Instalagi MySQL dapat dilakukan dengan langsung melakukan instalasi MySQL Installer atau melakukan instalasi aplikasi web server side yang mendukung MySQL.

a) MySQL Installer

Untuk melakukan instalasi MySQL Installer, terlebih dahulu dapat dilakukan download driver melalui link https://dev.mysql.com/downloads/installer/ .