

Sebelum membuat tutorial cara membuat database menggunakan Oracle Apex ini, akan dijelaskan beberapa penjelasan dasar mengenai Database. Database atau dalam Bahasa Indonesia yang berartikan Basis Data berasal dari dua suku kata Basis dan Data. Basis berdasarkan KBBI yang memiliki arti asas atau dasar sedangkan Data memiliki arti Keterangan atau bahan nyata yang dapat dijadikan dasar kajian (analisis atau kesimpulan). Jika dijadikan satu pengertian Basis Data mempunyai beragam salah satu pengertiannya Basis Data merupakan urat nadi sistem informasi sehingga peranannya dalam membentuk konsep laporan sangatlah penting yang membuat para pemakai dapat menggunakannya sesuai dengan kebutuhan. Berikut pengertian Basis Data menurut beberapa ahli:

1. Menurut *Fabbri dan Schwab*, basis data adalah suatu sistem terpadu yang dirancang terutama untuk meminimalkan dalam pengulangan data.
2. Menurut *Date*, basis data dapat dianggap sebagai tempat untuk sekumpulan berkas data terkomputerisasi.
3. Menurut *Chou*, Basis Data di definisikan sebagai kumpulan informasi bermanfaat yang diorganisasikan ke dalam tata cara yang khusus.

Sedangkan secara garis besarnya Basis Data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Basis Data dikelola secara langsung oleh perangkat lunak (software) yang disebut DBMS (Database Management System). Basis Data jika digabungkan dengan pengelolanya atau DBMS akan menghasilkan sebuah Sistem. Dalam pembuatan sebuah Sistem Basis Data memiliki Tingkatan atau level bagaimana dalam melihat data di sebuah Sistem Basis Data. Tingkatan atau level sebagai berikut:

4. Level Fisik (*Physical Level*)

Level terendah yang menunjukkan bagaimana data disimpan sebagai teks, angka bahkan bit data.

5. Level Konseptual (*Conceptual Level*)

Level yang menggambarkan data secara fungsional disimpan dalam Basis Data seperti disimpan dalam beberapa tabel atau file.

6. Level Penampakan (*View Level*)

Level tertinggi yang menunjukkan sebagian data dari basis data. Kemunculan data atau tampilan data diatur oleh aplikasi *end user* sehingga data pada level ini sudah data siap saji.

Basis Data memiliki Bahasa khusus yang mengatur interaksi atau komunikasi antara pemakai dengan basis data di dalam mengelola atau mengorganisasikan data. Contoh bahasa khusus Basis data adalah SQL (*Structure Query Language*) sebuah bahasa yang digunakan untuk mengakses data dalam Basis Data Relasional. SQL secara de facto merupakan bahasa standar yang digunakan dalam RDMS (Relational Database Management System). SQL juga merupakan bahasa baku (ANSI/SQL) non prosedural dan berorientasi himpunan (set oriented language) SQL dapat digunakan baik secara interaktif atau ditempelkan pada sebuah program aplikasi. Standarisasi SQL dimulai pada tahun 1986 ditandai dengan dikeluarkannya standar SQL oleh ANSI, kemudian diperbaiki pada tahun 1989 kemudian diperbaiki kembali pada tahun 1992. Selain itu MySQL software merupakan suatu aplikasi yang sifatnya open source serta server basis data MySQL memiliki kinerja sangat cepat, reliable, dan mudah untuk digunakan serta bekerja dengan arsitektur client server atau embedded systems. Dikarenakan faktor open source dan populer tersebut maka cocok untuk mendemonstrasikan proses replikasi basis data. MySQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan

yang sangat besar My SQL berkembang sudah mencapai versi 5. Beberapa sistem manajemen database relasional umum yang menggunakan SQL adalah: Oracle, Sybase, Microsoft SQL server, Acces, Ingres dan lain lain. Meskipun demikian perintah-perintah SQL standar seperti “Select”, ”Insert”, “ Update”, “Delete”, “Create”, “Drop” dapat digunakan pada seluruh DBMS tersebut. MySQL juga dapat menjalankan perintah-perintah Structured Query Language (SQL) untuk mengelola database-database yang ada di dalamnya. Query adalah pertanyaan atau permintaan informasi tertentu dari sebuah basis data yang ditulis dalam format tertentu. Query juga dapat di definisikan sebagai perintah yang digunakn untuk mengakses data pada sistem Basis Data untuk melakukan manipulasi terhadap Basis Data yang dikenal dengan nama SQL (Structured Query Language).

Secara umum SQL terdiri dari dua bahasa, yaitu:

1. Data Defination Language (DDL)

DDL adalah struktur atau skema Basis Data yang menggambarkan atau mewakili desain Basis Data secara keseluruhan dispesifikasikan dengan Bahasa DDL.

Data Defination Language dapat digunakan untuk:

- Membuat Tabel Baru
- Mengubah Tabel
- Menentukan Struktur Penyimpanan Tabel

Output dari kompilasi perintah DDL adalah kumpulan tabel yang disimpan dalam file khusus yang disebut kamus data (Data Dictionary). Kamus data merupakan suatu meta data atau super data yang mendeskripsikan data yang sesungguhnya. Kamus Data selalu diakses dalam suatu operasi Basis Data sebelum suatu file atau tabel data yang sesungguhnya diakses.

2. Data Manipulation Language (DML)

Data Manipulation Language (DML) adalah Bahasa Manipulasi Data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu Basis Data.

Data Manipulation Language dapat digunakan untuk:

- Menyisipkan atau menambahkan data baru ke suatu Basis Data
- Menghapuskan data dari suatu Basis Data
- Mengubah data di suatu Basis Data.

BAB II

Hirarki atau Jenjang Data

Dalam Basis Data terdapat Hirarki atau Jenjang Data, yang memiliki arti kumpulan yang terkecil sampai yang terbesar. Berikut merupakan gambaran dari jenjang data



1. *Character*

Character atau Karakter dalam KBBI memiliki makna huruf, angka, ruang simbol khusus yang dapat di tampilkan pada layar papan ketik. Karakter dalam Basis data adalah bagian data yang terkecil dapat berupa karakter numerik (angka 0 – 9) huruf (A – Z, a – z) ataupun karakter-karakter khusus, seperti *, &, %, # dan lain-lain.

2. *Field*

Field merepresentasikan suatu atribut dari record yang menunjukan suatu item dari data, seperti misalnya nama, alamat dsb.

Setiap field harus mempunyai:

- Field Name :Harus memberi 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

3. Record atau Baris Data

Kumpulan dari field item yang logis membentuk suatu record. Sebuah record menggambarkan suatu unit data individu yang tertentu

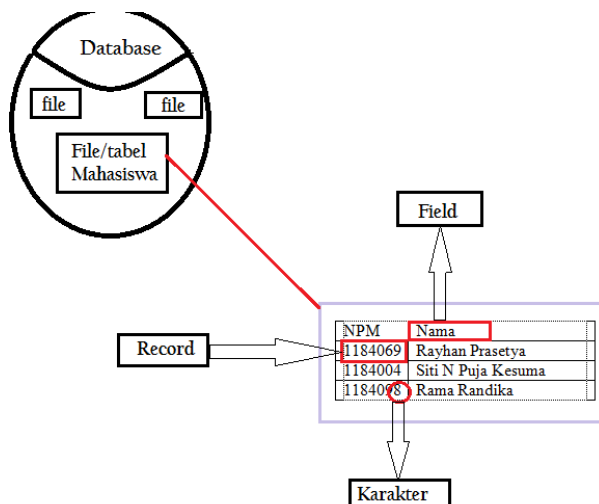
4. File atau Tabel

File terdiri dari kumpulan record yang menggambarkan satu kesatuan data yang sejenis dan logis berhubungan.

5. Database

Database merupakan kumpulan file-file yang berhubungan secara logis dan digunakan secara rutin pada operasi-operasi sistem informasi manajemen.

Contoh gambar dari jenjang data yang sudah dijelaskan diatas:



BAB III

Model Basis Data

Basis Data memiliki beberapa model. Model Basis Data adalah kumpulan dari konsepsi pada suatu Basis Data yang biasanya mewakili struktur dan relasi data yang terdapat pada suatu Basis Data. Sebuah model Basis Data adalah tempat dimana data atau suatu metodologi untuk menyimpan data. Model Basis Data menyatakan hubungan antar rekaman yang tersimpan dalam Basis Data. Jenis-jenis Model Basis Data:

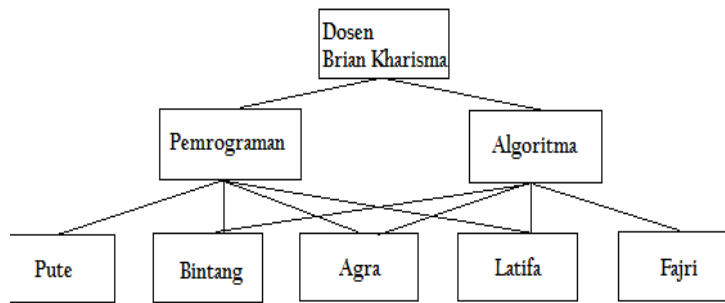
1. Model Basis Data Jaringan

Model jaringan merupakan model Basis Data yang secara fleksibel mewakili objek dan hubungan mereka. Model ini memiliki fitur istimewa yang pada skema diagram diperlihatkan sebagai grafik dengan tipe objek. Model Basis Data Jaringan terdiri atas record yang dihubungkan satu sama lain melalui link yang berupa pointer. Model Basis Data ini dapat menyatakan hubungan:

- Satu ke Satu (One To One) yang artinya satu orang tua memiliki satu anak
- Satu ke Banyak (One To Many) yang artinya satu orang tua punya beberapa anak
- Banyak ke Banyak (Many To Many) yang artinya beberapa anak punya beberapa orang tua.

Model Basis Data ini memiliki kelemahan yaitu lebih kompleks dan lebih susah dalam proses query dan dalam memanipulasi data lebih rumit karena harus menelusuri data pada setiap recordnya.

Contoh Gambar:



2. Model Basis Data Hirarki

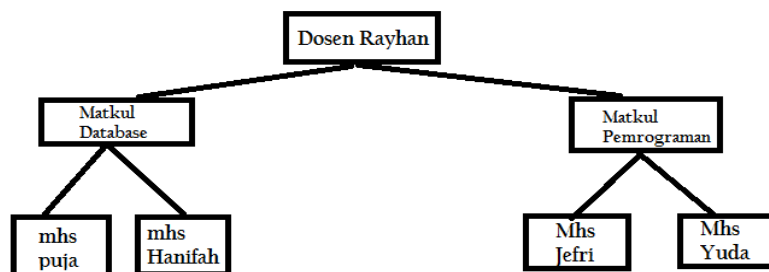
Model Basis Data ini disebut juga model pohon, karena hubungan antar simpul digambarkan seperti skema struktur pohon (*tree-structured*) yang dibalik dengan pola hubungan orang tua-anak (*parent-child*).

Model Basis Data ini menyatakan hubungan:

- Satu ke satu (One to one)
- Satu ke banyak (One to many) tetapi satu anak hanya boleh mempunyai satu orang tua.

Model Basis Data Hirarki ini memiliki kelemahan yaitu ketidakmampuan dalam mengelola hubungan banyak ke banyak (*many to many*) sehingga di dalam Model Basis Data Hirarki ini sering terjadinya redudansi atau duplikasi data.

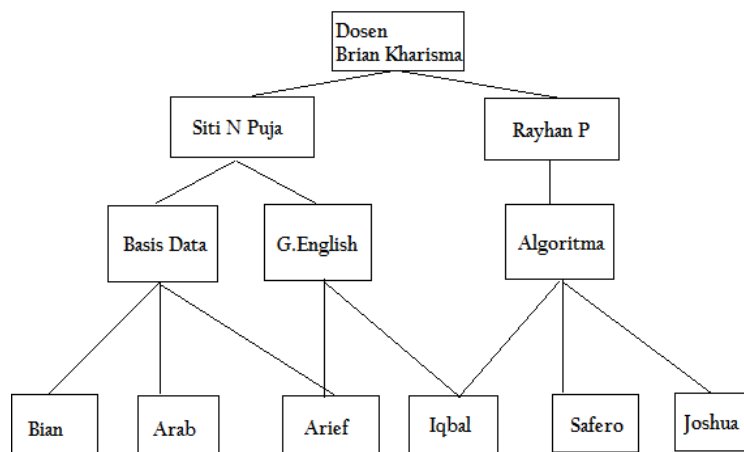
Contoh Gambar:



3. Model Basis Data Relasional

Model Data Relasional adalah suatu model Basis Data yang menggunakan tabel dua dimensi, yang terdiri atas baris dan kolom untuk menggambarkan sebuah berkas data. Model ini menunjukkan cara mengelola atau mengorganisasikan data secara fisik. Keuntungan Model Data ini adalah bentuknya sederhana dan mudah melakukan berbagai operasi data (query, update atau edit, delete), mampu mengurangi adanya redundancy data, fleksibel karena nilai data dalam tabel tidak ada pembatasan dalam berbagai proses pencarian data. Model ini menggunakan Bahasa Query Formal (prosedural dan Non Prosedural) dan Bahasa Query Komersial (QUEL, QBE, dan SQL)

Contoh Gambar:



Didalam Model Basis Data Relasional memiliki istilah-istilah. Berikut merupakan istilah-istilah yang sering dipakai pada Model Basis Data Relasional:

- **Relasi**

Relasi adalah sebuah tabel yang terdiri dari beberapa kolom dan baris.

Contoh:

NPM	Nama Mahasiswa
1184004	Siti N Puja Kesuma
1184069	Rayhan Prasetya
1184022	Echa Dwiifanka

Kode_Kuliah	Matkul
INF01	Basis Data
INF02	Pemrograman
INF03	Algoritma

- **Atribut**

Atribut adalah nama yang tercantum di kolom pada sebuah relasi(field).

- **Tuple**

Tupel adalah baris pada sebuah relasi(record) atau kumpulan elemen yang saling berkaitan.

- **Domain**

Domain adalah kumpulan nilai yang valid untuk satu atau lebih atribut bersifat atomik

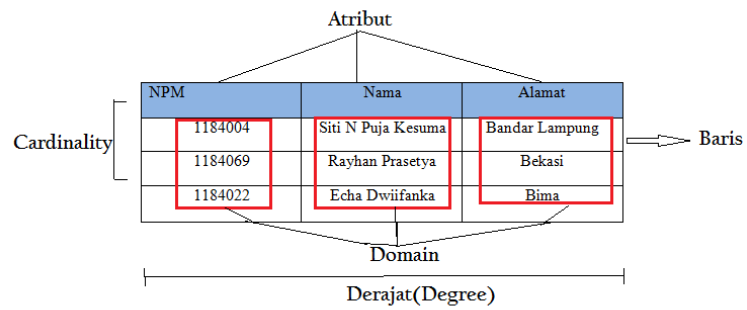
- **Derajat (*degree*)**

Derajat adalah jumlah atribut dalam sebuah relasi (jumlah field)

- **Cardinality**

Cardinality adalah jumlah tupel dalam sebuah relasi (jumlah record).

Contoh Gambar:



BAB IV

Rancangan Basis Data

Perancangan Basis Data diperlukan agar dapat terbentuk Basis Data yang efisien dalam penggunaan ruang penyimpanan, Cepat dalam pengaksesan dan mudah dalam memanipulasi (Tambah, ubah dan hapus) data. Tahapan dalam Perancangan Basis Data ada beberapa yaitu:

1. Normalisasi Data

Normalisasi dalam KBBI adalah tindakan mengembalikan pada keadaan, hubungan, dan sebagainya yang biasa atau yang normal.

Normalisasi dalam Basis Data adalah proses pengelompokan atribut yang membentuk entitas sederhana, non redudansi, fleksibel dan mudah beradaptasi sehingga dapat dipastikan bahwa Basis Data yang dibuat berkualitas baik. Normalisasi merupakan cara pendekatan lain dalam membangun desain logic sebuah Basis Data Relasional dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal atau baik. Dalam pendekatan normalisasi ini perancangan basis data bertolak dari situasi yang nyata dimana telah memiliki item-item data yang siap ditempatkan dalam baris dan kolom pada tabel-tabel relasional.

2. Atribut

Atribut adalah kategori variabel kualitatif atau karakteristik dari entity yang menyediakan penjelasan detail tentang entity atau relationship tersebut. Atribut sendiri identik dengan Kolom data (field) pada sebuah tabel.

Atribut dapat dibedakan menjadi beberapa kelompok:

- **Atribut Kunci**

Atribut kunci adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data dalam tabel secara unik. Dikatakan unik jika pada atribut yang dijadikan kunci tidak boleh ada baris data dengan nilai yang sama.

Ada 4 macam key yang dapat diterapkan pada suatu tabel yaitu:

- *Super Key*

Super Key merupakan satu atau lebih atribut (Kumpulan Atribut) yang dapat membedakan setiap baris data dalam sebuah tabel secara unik

- *Candidate Key*

Candidate key merupakan kumpulan atribut minimal yang dapat membedakan atau mengidentifikasi nilai-nilai kombinasi setiap baris data dalam sebuah tabel secara unik. Candidate Key tidak boleh berisi atribut dari tabel lain.

- *Foreign Key*

Foreign Key atau dalam Bahasa Indonesia Kunci Tamu adalah atribut dengan domain yang sama yang akan menjadi kunci utama pada sebuah tabel tetapi pada tabel lain atribut tersebut tersebut hanya sebagai atribut biasa. Foreign Key ini adalah kunci penyambung antar tabel atau atribut yang digunakan untuk merelasikan atau atribut yang melengkapi satu relationship yang menunjukkan ke induknya, dengan kata lain keduanya saling berkaitan.

- *Primary Key*

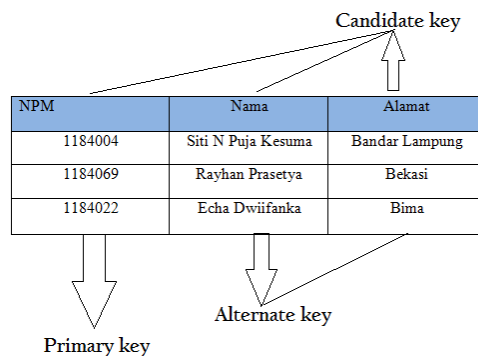
Primary key atau Kunci Utama adalah atribut yang telah dipilih untuk mengidentifikasi setiap record secara unik.

Kunci utama harus merupakan atribut yang benar-benar unik dan tidak boleh ada nilai NULL.

Kriteria menentukan Primary Key:

- Atribut tersebut dijadikan acuan
- Atribut lebih sederhana
- Atribut cukup unik.

Contoh:



- **Atribut Sederhana**

Atribut sederhana adalah atribut atomik yang tidak dapat dipilah lagi atau diuraikan menjadi sub-sub atribut.

Contoh:

Nama_Mahasiswa
Rayhan Prasetya
Siti N Puja Kesuma

- **Atribut Komposit**

Atribut komposit adalah atribut yang masih dapat diuraikan lagi menjadi sub-sub atribut yang masing-masing memiliki makna.

Contoh :

Alamat_Mahasiswa
Jl.Cijarokaso No.18 Sarijadi
Jl.Wala Jaya No.11 Waylaga
Jl.Cilandak NO.50 Sarimanah

- **Atribut Bernilai Tunggal**

Atribut bernilai tunggal adalah atribut-atribut yang memiliki paling banyak satu nilai untuk setiap baris data.

Contoh:

Nama Mahasiswa	NPM
Rayhan Prasetya	1184069
Siti N Puja Kesuma	1184004

- **Atribut Bernilai Banyak**

Atribut bernilai banyak adalah atribut yang dapat diisi dengan lebih dari satu nilai, tetapi jenisnya sama.

Contoh :

Keahlian
Menyanyi
Menjahit

- **Atribut Harus Bernilai**

Atribut harus bernilai adalah atribut pada sebuah tabel yang harus berisi data (nilainya tidak boleh kosong).

Contoh:

Nama Mahasiswa	NPM
Rayhan Prasetya	1184069
Siti N Puja Kesuma	1184004

- **Atribut Nilai Null**

Atribut nilai null adalah atribut yang nilainya boleh dikosongkan. Dapat digunakan untuk menyatakan atau mengisi atribut-atribut yang nilainya memang belum siap atau tidak ada.

Contoh:

Keahlian

- **Atribut Turunan**

Atribut turunan adalah atribut-atribut yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari atribut tabel lain yang berhubungan. Dapat ditiadakan dari sebuah tabel, karena nilainya bergantung pada nilai yang ada di atribut lain.

Contoh:

IP
3,05
3,40

3. Tipe Data

Tipe data digunakan untuk mendefinisikan suatu field atau kolom. Setiap kolom yang dibuat harus didefinisikan terlebih dahulu. Lebih merujuk pada kemampuan penyimpanan data yang mungkin bagi suatu atribut secara fisik. Tipe data bagi setiap atribut relevan untuk diperhitungkan pada saat di implementasikan pada Basis Data. Jenis tipe data ada beberapa macam. Berikut ini macam-macam tipe data yang digunakan di SQL:

- **Tipe Numerik**

Tipe data numerik digunakan untuk menyimpan data berupa angka. Ciri utama nya adalah suatu data yang memungkinkan untuk dikenai operasi aritmatika seperti penambahan, pengurangan, perkalian dan pembagian.

Berikut ini tipe field (kolom) di MySQL yang termasuk kedalam kelompok tipe numerik:

- *Tinyint*
- *Smallint*
- *Mediumint*
- *Int*
- *Bigint*
- *Float*
- *Double atau real*
- *Decimal atau numeric*

- **Tipe Date dan Time**

Tipe date dan time digunakan untuk menyimpan data dan waktu. Berikut ini tipe field (kolom) di MySQL yang termasuk ke dalam kelompok tipe date dan time:

- *Date*

- *Time*
- *DateTime*
- *Year*

- **Tipe String (Text)**

Tipe data string digunakan untuk menyimpan data text. Ciri utama string adalah suatu data yang memungkinkan untuk dikenai operasi aritmatika seperti penambahan, pengurangan, perkalian dan pembagian.

Berikut ini tipe field (kolom) di MySQL yang termasuk kedalam kelompok tipe String.

- *Char*
- *Varchar*
- *Tinytext*
- *Text*
- *Mediumtext*
- *Longtext*

- **Tipe BLOB (Biner)**

Tipe Blob atau Binary Large Object digunakan untuk menyimpan data biner. Tipe ini biasanya digunakan untuk menyimpan kode-kode biner dari suatu file atau object.

Berikut ini tipe field(kolom) yang termasuk kelompok tipe blob

- *Bit*
- *Tinyblob*
- *Blob*
- *Mediumblob*
- *Longblob*

4. Operator

Operator merupakan simbol yang digunakan untuk menginstruksikan program untuk melakukan sesuatu. Operator digunakan dalam query untuk melakukan filter data.

Berikut merupakan jenis-jenis operator di dalam Basis Data:

- **Operator Relasional**

Operator	Arti
=	Sama dengan
!= atau <>	Tidak sama dengan
>	Lebih dari
>=	Lebih dari atau sama dengan
<	Kurang dari
<=	Kurang dari atau sama dengan

- **Operator Logika**

Opernd 1	Operand 2	Hasil Operasi	
		Operator OR	Operator AND
Salah	Salah	Salah	Salah
Salah	Benar	Benar	Salah
Benar	Salah	Benar	Salah
Benar	Benar	Benar	Benar

- **Operator Aritmatika**

Operator	Keterangan	Prioritas
*	Perkalian	1
/	Pembagian	1
%	Sisa Pembagian	2
DIV	Hasil Pembagian bulat	2
+	Penjumlahan	3
-	Pengurangan	3

BAB V

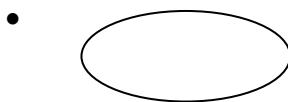
Diagram Entity-Relationship (ERD)

Diagram Entity Relationship adalah suatu model untuk menjelaskan hubungan antar data di dalam Basis Data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD merupakan model data berupa konseptual. Tujuan dibuatnya Diagram E-R ini adalah untuk menunjukkan objek-objek(himpunan entitas) yang terlibat dalam sebuah basis data dan bagaimana hubungan yang terjadi diantara objek-objek tersebut.

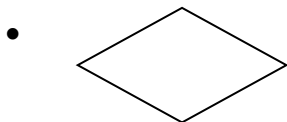
Diagram Entity Relationship ini memiliki notasi simbolik yang terdiri dari :



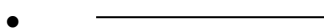
Persegi Panjang yang menyatakan Himpunan Entitas



Elip menyatakan Atribut. Atribut yang berupa kunci(key) diberi tanda unik yaitu digaris bawah.

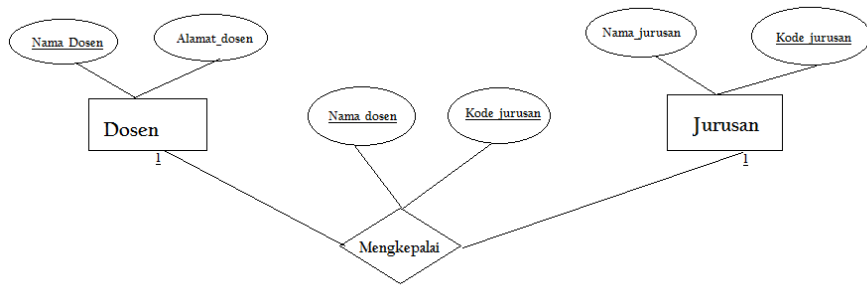


Belah Ketupat yang menyatakan Himpunan relasi.

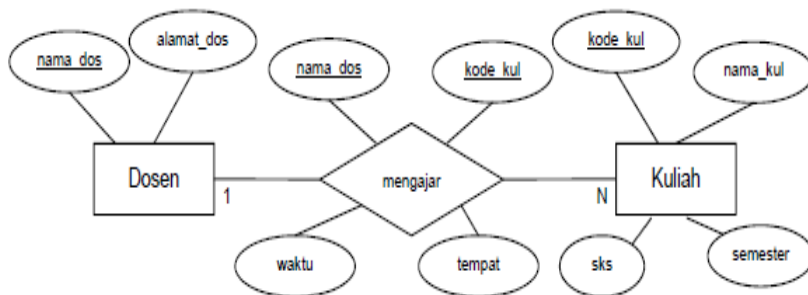


Garis digunakan sebagai penghubung himpunan entitas dengan himpunan relasi.

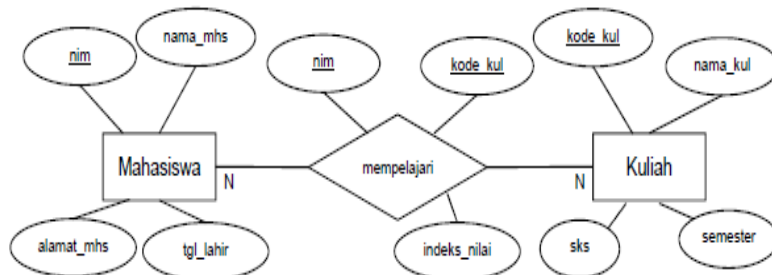
Contoh Diagram E-R **satu ke satu**:



Contoh Diagram E-R **satu ke banyak**:



Contoh Diagram E-R **banyak ke banyak**:



Tahapan pembuatan Diagram E-R

- Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat
- Menentukan atribut-atribut key dari masing-masing himpunan entitas
- Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas beserta foreignkey nya
- Menentukan derajat/kardinalitas relasi untuk setiap himpunan relasi

- Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif.

Entity atau Entitas adalah sebuah objek yang keberadaannya dapat dibedakan terhadap objek lain. Entity sendiri memiliki beberapa varian, biasanya himpunan entitas dalam Diagram E-R adalah himpunan entitas kuat. Berikut merupakan varian entitas dalam Diagram E-R:

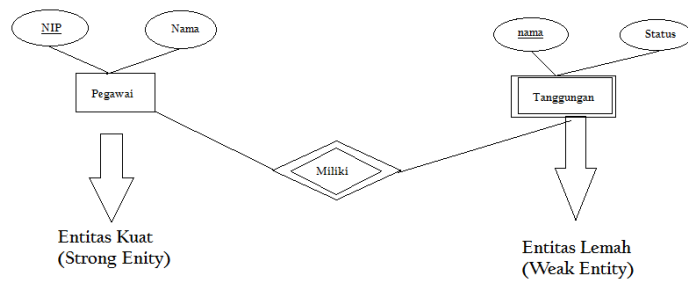
- *Strong Entity Sets*

Strong entity sets tidak memiliki ketergantungan ketergantungan dengan himpunan entitas lainnya, dimana kemunculan entitas-entitas di dalamnya tidak tergantung pada keberadaan entitas di himpunan entitas lainnya. *Strong Entity Sets* ini entitas yang memiliki atribut kunci. Daya entitas kuat mempunyai ciri khas yang istimewa yaitu identifier (suatu atribut tunggal atau perpaduan atribut-atribut yang secara khas bisa dipakai untuk membedakannya dari entitas kuat yang lain).

- *Weak Entity Sets*

Weak Entity Sets biasanya tidak memiliki atribut yang dapat berfungsi sebagai *key* (yang benar-benar menjamin keunikan entitas didalamnya). Entitas lemah di identifikasikan dengan mengaitkan entitas penting dari jenis entitas yang lain ditambah atribut yang berasal dari entitas lemah.

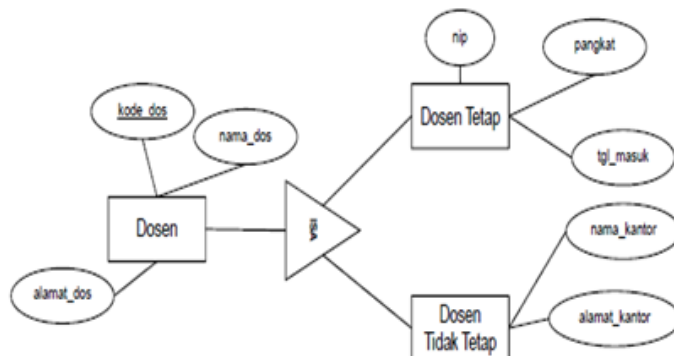
Contoh:



- *Subtype Entity*

Sub entitas adalah himpunan entitas yang beranggotakan entitas-entitas bagian dari himpunan entitas yang lebih superior/utama. Sub Entitas ini merupakan hasil dekomposisi (spesialisasi) himpunan entitas berdasarkan pengelompokan tertentu dari himpunan entitas yang lain.

Contoh:



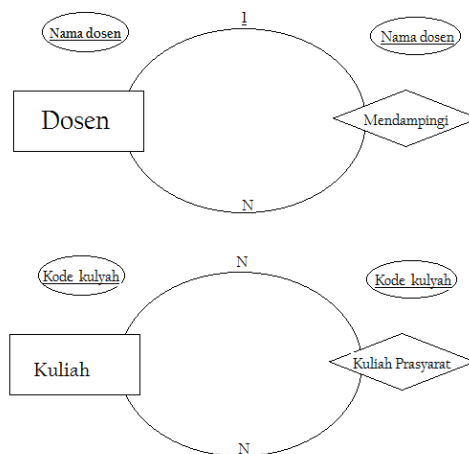
Relasi yang terjadi di antara dua himpunan entitas yang berbeda disebut sebagai relasi Biner (*Binary Relation*) yang merupakan relasi yang paling umum digunakan. Relasi hanya melibatkan sebuah himpunan entitas atau lebih dari dua himpunan entitas.

Relasi memiliki beberapa varian, berikut varian-varian dari relasi:

- **Relasi Tunggal (*unary relation*)**

Relasi Tunggal adalah relasi yang terjadi dari sebuah himpunan entitas ke himpunan entitas yang sama

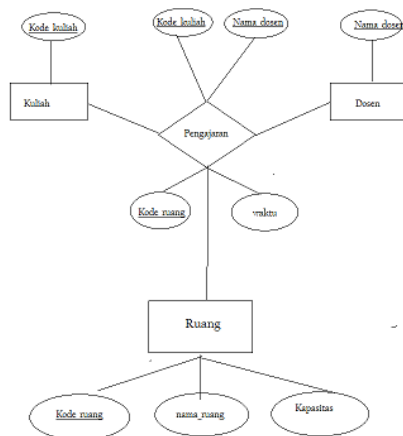
Contoh:



- **Relasi Multi Entitas (*N-ary Relation*)**

Relasi Multi Entitas adalah relasi dari tiga himpunan atau lebih. Relasi ini disarankan untuk dihindari, karena akan mengaburkan derajat relasi (derajat parsial) yang ada dalam relasi tersebut.

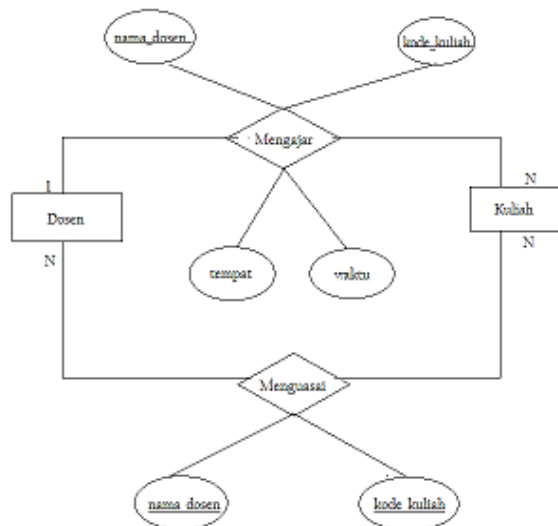
Contoh:



- **Relasi Ganda (*Redundant Relation*)**

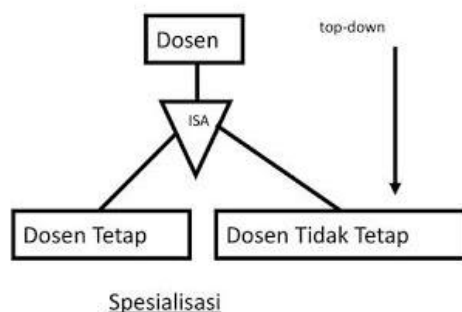
Relasi Ganda adalah relasi yang muncul antara dua himpunan entitas tidak hanya satu relasi, tetapi ada lebih dari satu relasi.

Contoh:



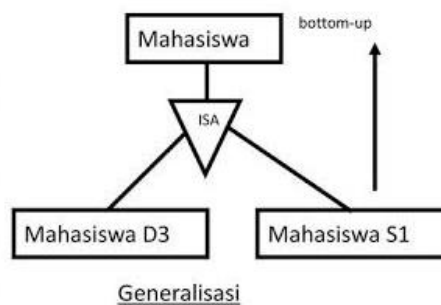
Spesialisasi adalah proses dari sebuah entitas yang dikelompokkan berdasarkan atribut-atribut yang berbeda yang akan melahirkan himpunan entitas baru (proses top-down). Spesialisasi ditekankan pada perbedaan antar kelompok entitas. Notasi relasi yang digunakan adalah relasi ISA berasal dari kata 'Is A'.

Contoh:



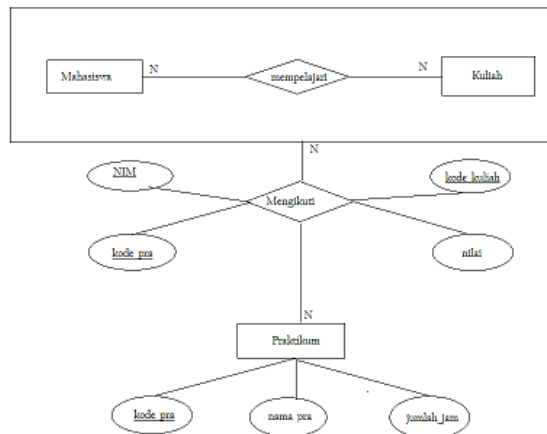
Generalisasi adalah proses dari sebuah entitas, lalu dikelompokkan berdasarkan atribut-atribut yang sama, yang akan melahirkan himpunan entitas baru (*proses bottom-up*). Yang ditekankan pada generalisasi adalah persamaan antar kelompok entitas.

Contoh:



Agreagasi dalam realitas dapat dijumpai adanya relasi yang secara kronologis mensyaratkan telah adanya relasi lain. Sebuah relasi terbentuk tidak hanya dari entitas tetapi juga mengandung unsur dari relasi lain. Fenomena tersebut diakomodasi dengan agregasi.

Contoh:

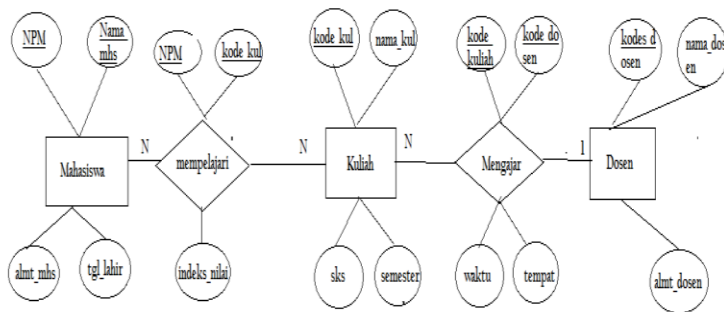


Modifikasi Diagram E-R mengarah pada penyempurnaan dan optimis model data karena pertimbangan-pertimbangan efisiensi ruang atau kecepatan dan kemudahan dalam mengakses data.

- *Key Alternatif (Alternate Key)*

Key Alternatif adalah key yang membedakan secara unik setiap entitas pada himpunan entitasnya, namun tidak dikenal dalam pemakaian sehari-hari sehingga bukan merupakan fakta yang telah ada di dunia nyata. Key dapat dikategorikan baik jika berukuran kecil dan sekuensial.

Contoh:



- Pengkodean Internal

Pengkodean internal adalah data coding yang merupakan cara untuk menyatakan suatu data dalam bentuk lain.

Data coding terdiri dari 3 bentuk, yaitu:

- *Sekuensial*

Sekuensial adalah pengkodean yang dilakukan dengan mengasosiasikan data dengan kode urut (biasanya berdasarkan bilangan atau abjad)

Contoh:

Data Hari ('Senin', 'Selasa', '.....', 'Sabtu')

Data Nilai ('A', 'B', 'C', 'D')

- *Mnemonic*

Mnemonic adalah pengkodean yang dilakukan dengan membentuk suatu singkatan dari data yang ingin dikodekan.

Contoh:

Data Mata Kuliah ('Matematika Diskrit', 'Database') dikodekan dengan ('Matdis', 'DB')

➤ *Blok*

Blok adalah pengkodean yang dinyatakan dalam format.

Contoh:

Data no.induk mahasiswa dengan format XXYYYY yang terbentuk atas XX=dua digit terakhir angka tahun masuk dan YYYY = no.urut mahasiswa.

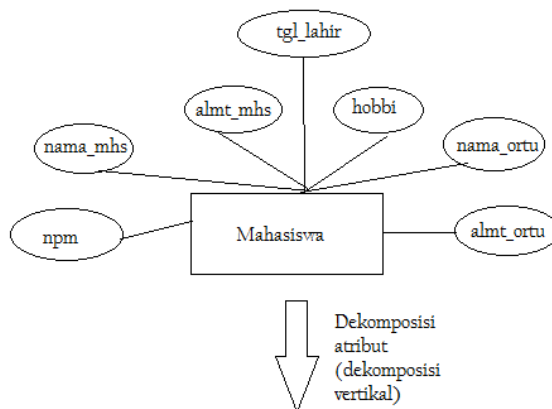
Dekomposisi Himpunan Entitas adalah sebuah himpunan entitas yang ada dalam sebuah diagram E=R dapat didekomposisi menjadi beberapa himpunan entitas baru karena pertimbangan efisiensi ruang penyimpanan dan pertimbangan kemudahan atau kecepatan pengaksesan data. Upaya dekomposisi ini senantiasa akan menghasilkan satu himpunan entitas kuat dan satu atau beberapa himpunan entitas lemah atau sub entitas.

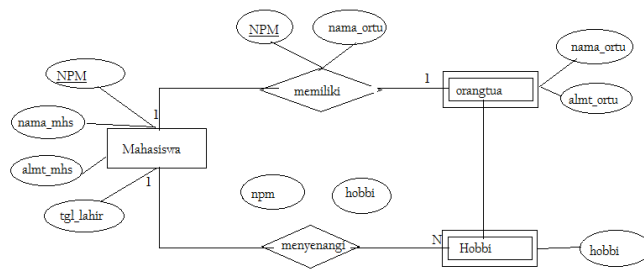
Dekomposisi memiliki dua bentuk yaitu:

- Dekomposisi Atribut

Dekomposisi atribut adalah dekomposisi dengan dengan membagi sebuah himpunan entitas menjadi dua atau lebih dengan pemisahan atribut.

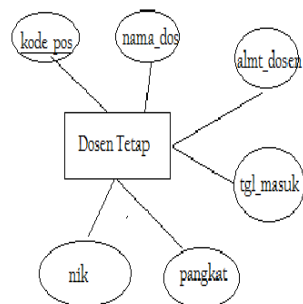
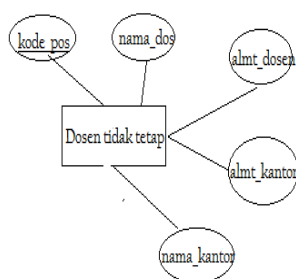
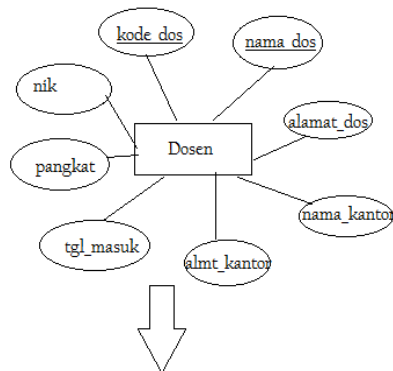
Contoh:





- Dekomposisi Entitas, dekomposisi dengan membagi sebuah himpunan entitas menjadi dua atau lebih dengan pemisahan entitas.

Contoh:



Fleksibilitas didalam basis data dapat direalisasikan dalam bentuk :

- Penambahan atribut
- Pemilihan domain atribut yang lebih luas
- Generalisassi
- Perubhana struktur entitas dari yang berorientasi kolom sampai menjadi yang berorientasi baris

BAB VI

Transformasi atau Implementasi

Model data ke basis data fisik

Transformasi Data adalah upaya yang dilakukan dengan tujuan utama untuk mengubah skala pengukuran **data** asli menjadi bentuk lain sehingga **data** dapat memenuhi asumsi-asumsi yang mendasari analisis ragam. Dalam mengimplementasikan Basis Data adalah upaya untuk membangun Basis Data fisik yang ditempatkan dalam memori sekunder dengan bantuan DBMS tertentu yang dipilih. Tahap ini diawali dengan melakukan **transformasi** dari model data yang telah selesai dibuat kedalam struktur tabel basis data sesuai dengan DBMS tertentu yang dipilih.

Ketentuan transformasi:

- Sebuah Diagram E-R direpresentasikan menjadi sebuah basis data secara fisik
- Himpunan entitas dan himpunan relasi yang ada pada Diagram E-R ditransformasikan melalui aturan tertentu menjadi tabel-tabel
- Atribut-atribut pada setiap himpunan entitas maupun himpunan relasi dinyatakan sebagai *field-field* dari tabel yang sesuai.

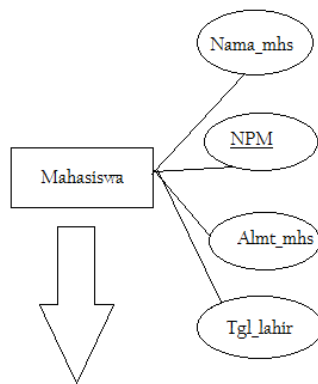
Transformasi memiliki banyak bentuk:

1. Transformasi Umum atau dasar

Transformasi umum atau dasar adalah aturan dalam pemetaan Model Data (Level Konseptual dalam Abstraksi Data) yang digambarkan dengan diagram E-R menjadi Basis Data Fisik. Level Fisik dalam abstraksi data adalah:

- Setiap himpunan entitas akan diimplementasikan sebagai sebuah tabel (file data)

Contoh:

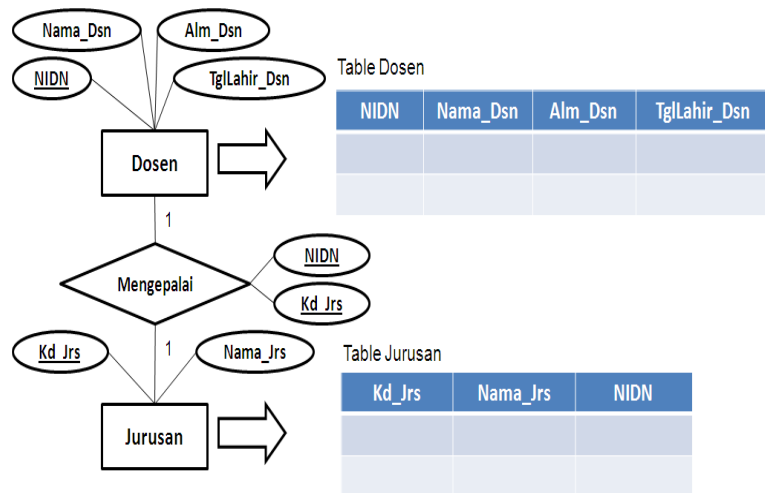


Tabel mahasiswa

NPM	Nama_mhs	Almt_mhs	Tgl_lahir

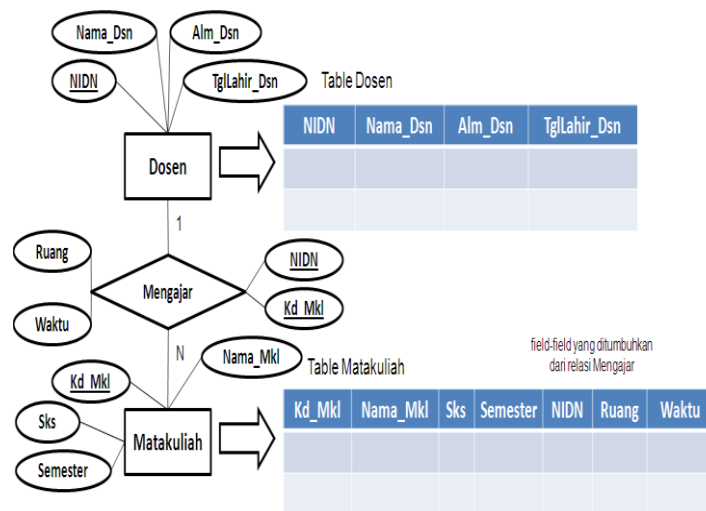
- Relasi dengan derajat relasi 1:1 (satu ke satu) yang akan menghubungkan dua buah himpunan entitas akan direpresentasikan dalam bentuk penambahan atau penyertaan atribut-atribut relasi ke tabel yang mewakili salah satu atribut key dari kedua himpunan entitas yang memiliki derajat minimum lebih besar atau diperkirakan jumlah barisnya lebih sedikit.

Contoh:



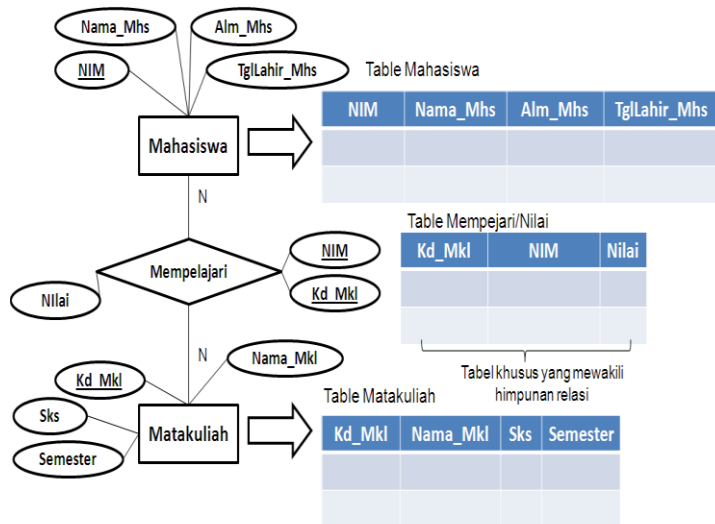
- Relasi dengan Derajat relasi 1-N (satu ke banyak) yang menghubungkan dua himpunan entitas juga akan direpresentasikan dalam bentuk pemberian atau pencantuman atribut key dari himpunan entitas pertama yang berderajat 1 ke tabel yang mewakili himpunan entitas kedua yang berderajat N. Atribut key dari himpunan pertama akan menjadi atribut tambahan bagi himpunan entitas kedua.

Contoh:



- Himpunan relasi dengan derajat N-N yang menghubungkan 2 buah himpunan entitas, akan direpresentasikan dalam bentuk tabel khusus yang memiliki field (*foreign key*) yg berasal dari *key-key* dari Himpunan entitas yang dihubungkannya.

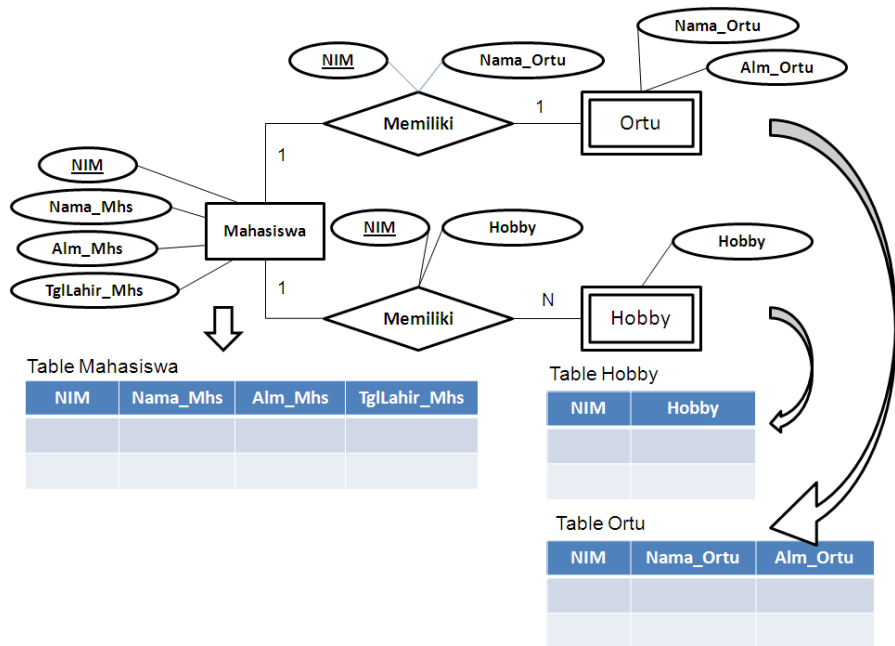
Contoh:



2. Transformasi dari Himpunan Entitas Lemah dan Sub Entitas

Himpunan Entitas Lemah dan Sub Entitas hanya dapat ditransformasi menjadi tabel dengan menyertakan pula atribut *key* yang ada di Himpunan Entitas Kuat yang berelasi dengannya. Bedanya jika himpunan entitas kuat sudah dapat langsung menjadi sebuah tabel utuh atau sempurna walaupun tanpa melihat relasinya dengan himpunan entitas yang lain.

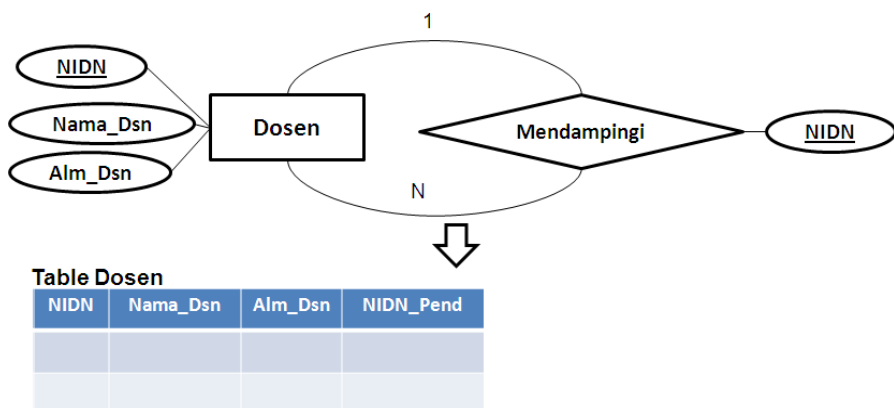
Contoh:



3. Transformasi dari Relasi Tunggal

Relasi Tunggal dari/ke himpunan entitas yang sama dalam Diagram E-R tergantung derajat relasinya. Relasi satu ke banyak di transformasikan melalui penggunaan field key dua kali tapi untuk fungsi yang berbeda.

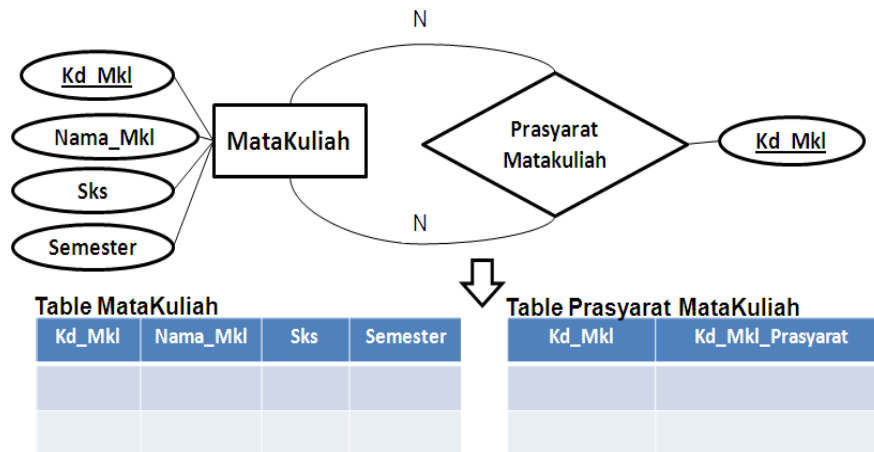
Contoh:



Relasi yang derajatnya banyak ke banyak akan di transformasikan melalui pembentukan tabel baru yang merepresentasikan relasi

tersebut. Tabel baru ini akan mendapatkan field dari semua atribut relasi yang akan ditambah dengan atribut key dari himpunan entitasnya.

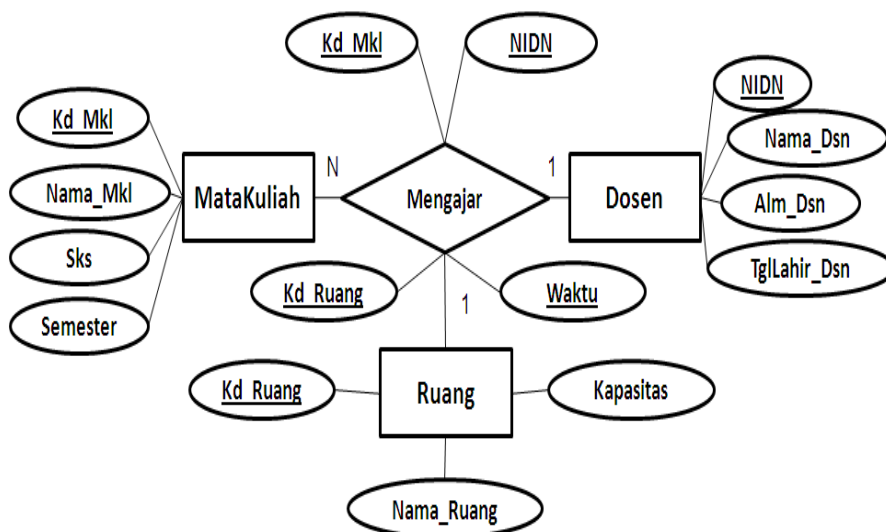
Contoh:



4. Transformasi dari relasi multi entitas

Relasi Multi Entitas yang menghubungkan lebih dari dua himpunan entitas (N himpunan entitas dimana $N > 2$) akan ditransformasikan sebagai sebuah tabel. Jika pada relasi yg menghubungkan N buah himpunan entitas, kita dapat memastikan bahwa derajat relasi parsial diantara $(N-1)$ buah himpunan entitas dengan suatu himpunan entitas (misalnya X) adalah satu-ke-banyak, maka relasi tadi tidak perlu diwujudkan sbagai sebuah tabel khusus dan atribut-atributnya cukup dilekatkan pada himpunan entitas X tersebut.

Contoh:



Hasil Akhir Transformasinya adalah:

Table MataKuliah

Kd_Mkl	Nama_Mkl	Sks	Semester	NIDN	Kd_Ruang	Waktu

Table Pengajaran/Jadwal

Kd_Mkl	NIDN	Kd_Ruang	Waktu

3 buah field yang mewakili relasi Pengajaran

Table Dosen

NIDN	Nama_Dsn	Alm_Dsn	TglLahir_Dsn

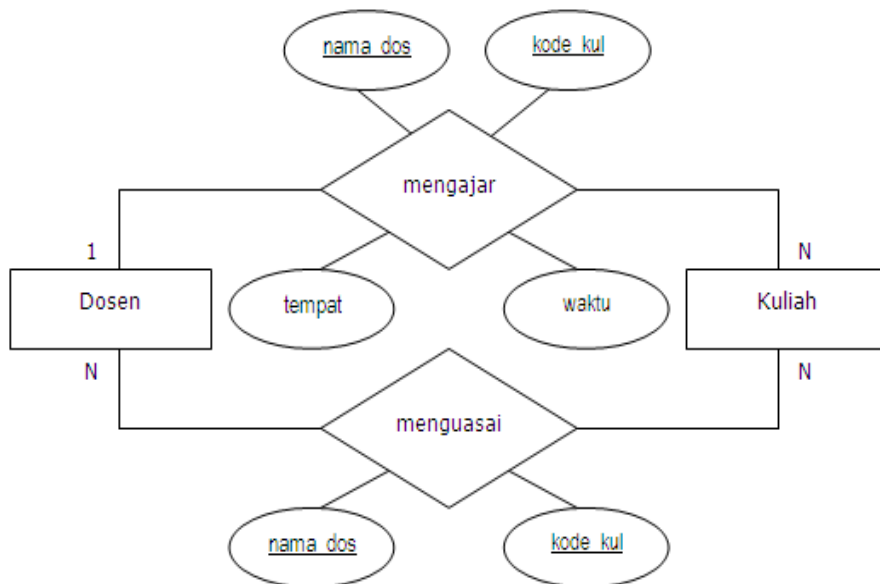
Table Ruang

Kd_Ruang	Nama_Ruang	Kapasitas

5. Transformasi dari Relasi Ganda

Transformasinya kita tinjau pada masing-masing himpunan relasi dan berdasarkan derajat relasi di masing-masing himpunan relasinya.

Contoh:



Hasil Akhir Transformasinya adalah:

Tabel Dosen

Kode_dos	Nama_dos	Alamat_mhs

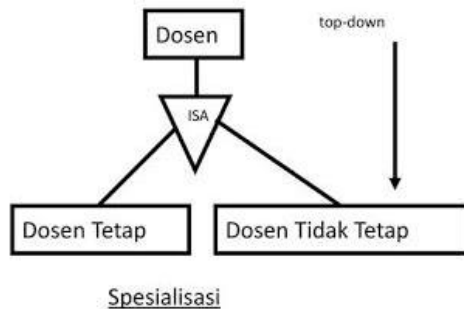
Tabel Kuliah

Kode_kul	Nama_kul	sks	semester	Kode_pos

Tabel Menguasai

Kode_dos	Kode_kul

6. Transformasi Spesialisai



Menjadi:

Tabel Dosen

Kode_dos	Nama_dos	Alamat_dos

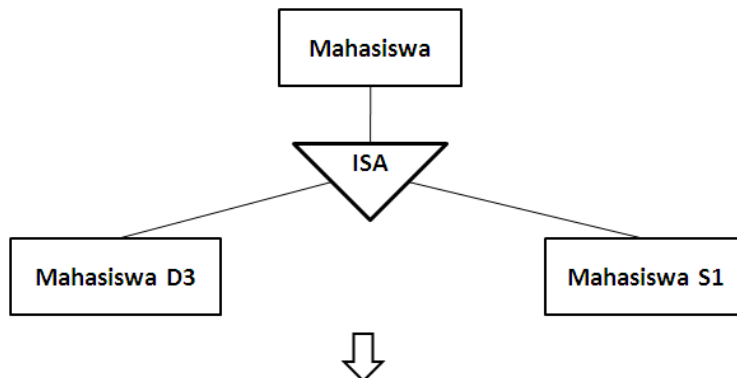
Tabel Dosen Tetap

Kode_dos	NIK	Pangkat	Tgl_masuk

Tabel Dosen Tidak Tetap

Kode_dos	Nama_kantor	Alamat_kantor

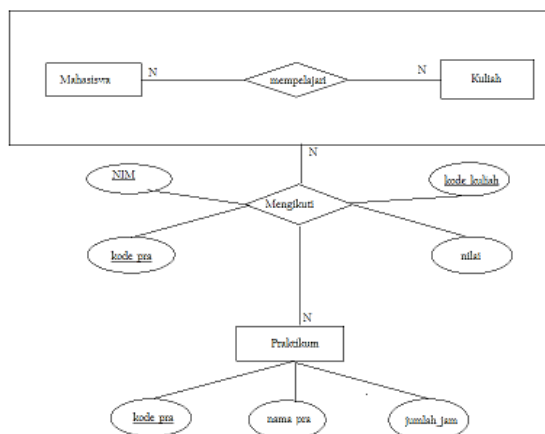
7. Transformasi Generalisasi



Tabel Mahasiswa

<u>Nim</u>	<u>Nama_mhs</u>	<u>Alamat_mhs</u>	<u>Tgl lahir</u>	<u>Prog studi</u>

8. Transformasi Agregasi



Menjadi:

Tabel Nilai

NIM	Kode_kul	Indeks_nilai

Tabel Praktikum

Kode_pra	Nama_pra	Jumlah_jam

Tabel Mengikuti

NIM	Kode_kul	Kode_pra	nilai

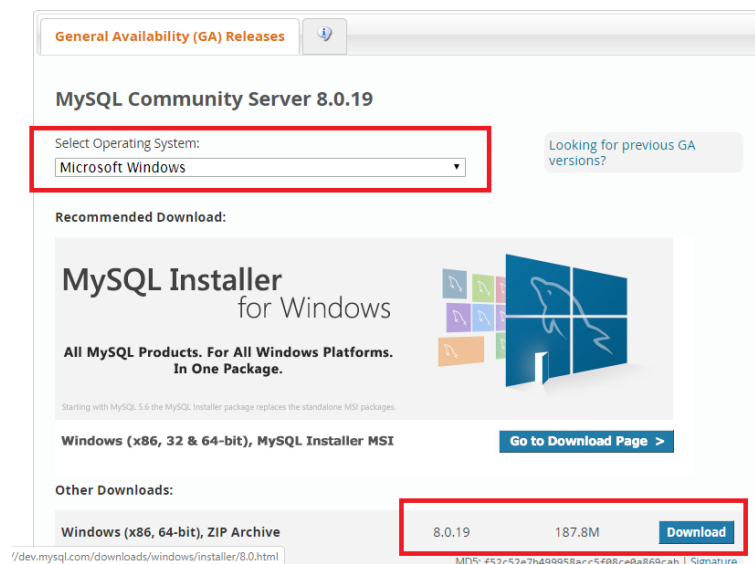
BAB VII

Server Basis Data My SQL

Pada Bab I telah dijelaskan Server Basis Data MySQL secara singkat. Pada Bab ini akan dijelaskan MySQL secara mendetail serta cara menginstall MySQL dan cara mengimplementasikannya.

1. Cara Instalasi Mysql

- Kunjungi website resmi MySQL <http://dev.mysql.com/downloads/mysql/>
- Lalu Scroll down dan pilih Download seperti yang terlihat pada gambar dibawah ini, namun sesuaikan dengan Operating System yang digunakan.



- Lalu pilih “No Thanks” seperti gambar dibawah ini, dan tunggu sampai selesai di download

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »

using my Oracle Web account

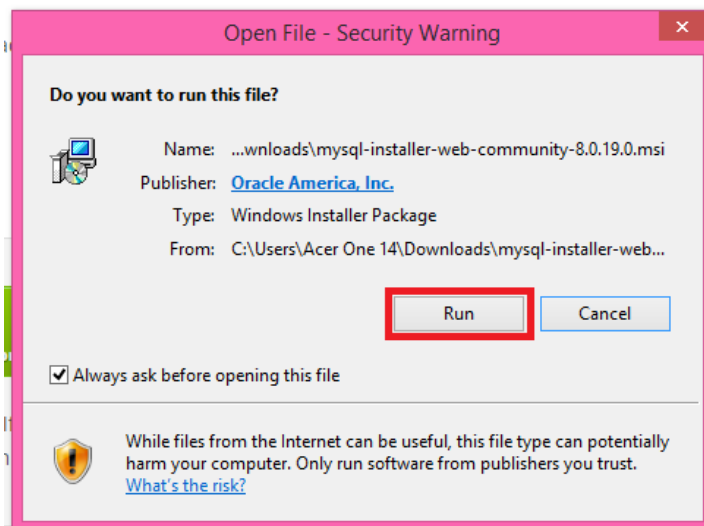
Sign Up »

for an Oracle Web account

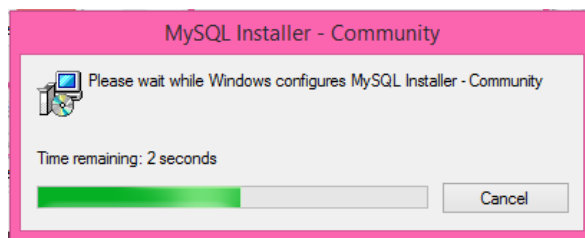
MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

[No thanks, just start my download.](#)

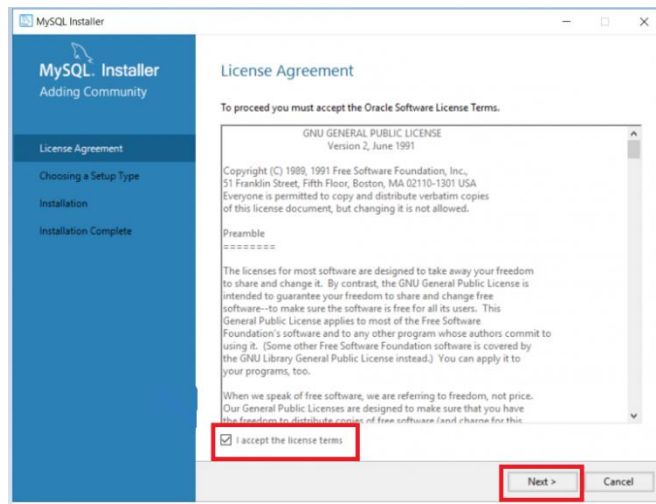
- Setelah ter-download silahkan buka dan akan muncul seperti gambar dibawah lalu klik “Run”



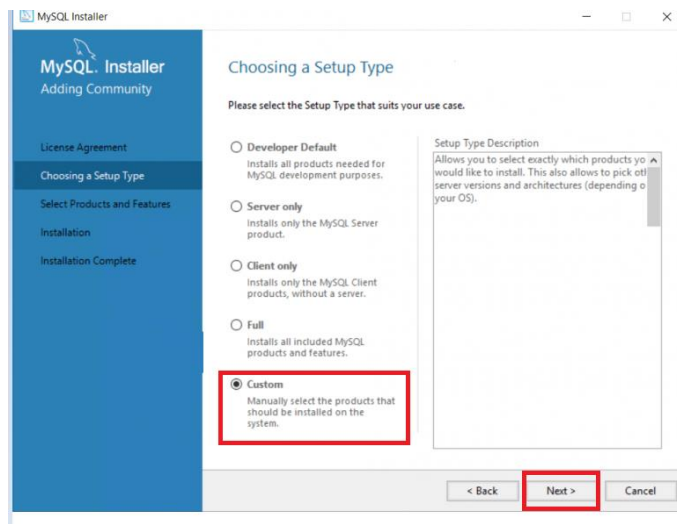
- Lalu tunggu proses seperti dibawah ini



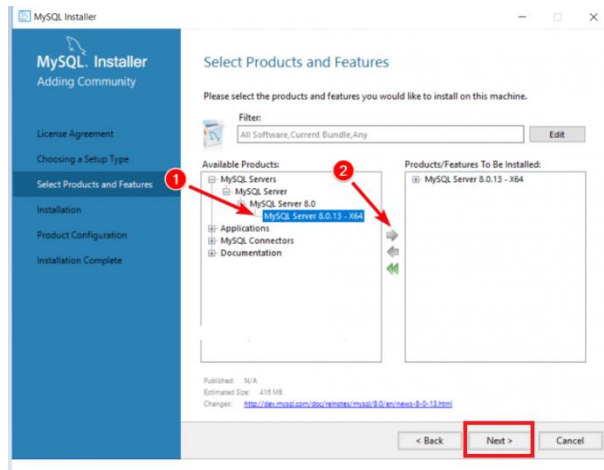
- Setelah itu silahkan ikuti seperti yang terlihat pada gambar



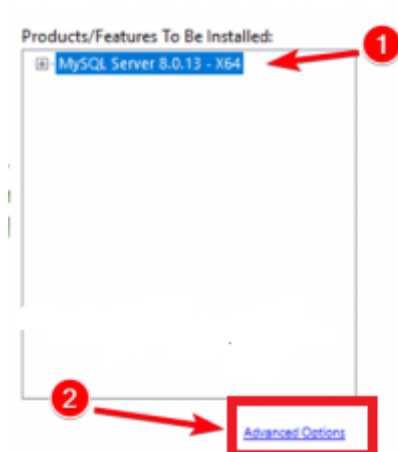
- Lalu pilih custom untuk dapat mengubah folder instalasi MySQL



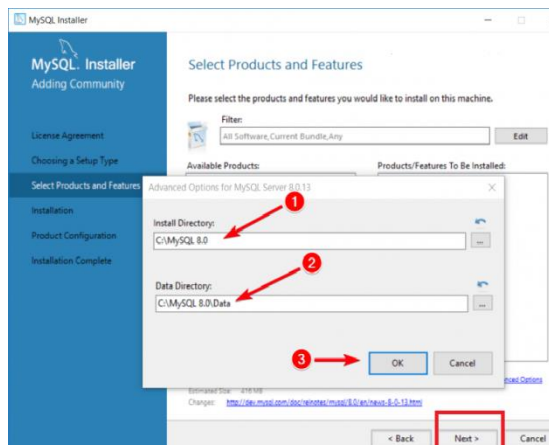
- Lalu ikuti petunjuk yang terdapat pada gambar



- Klik MySQL Server

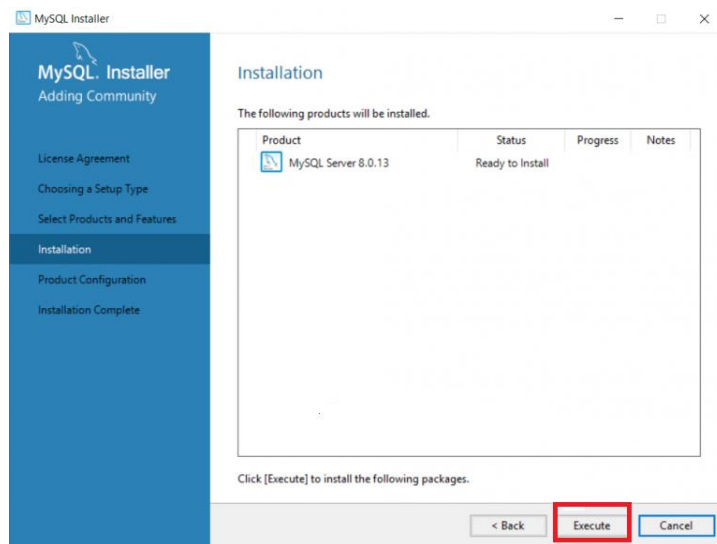


- Akan terdapat pilihan seperti pada gambar

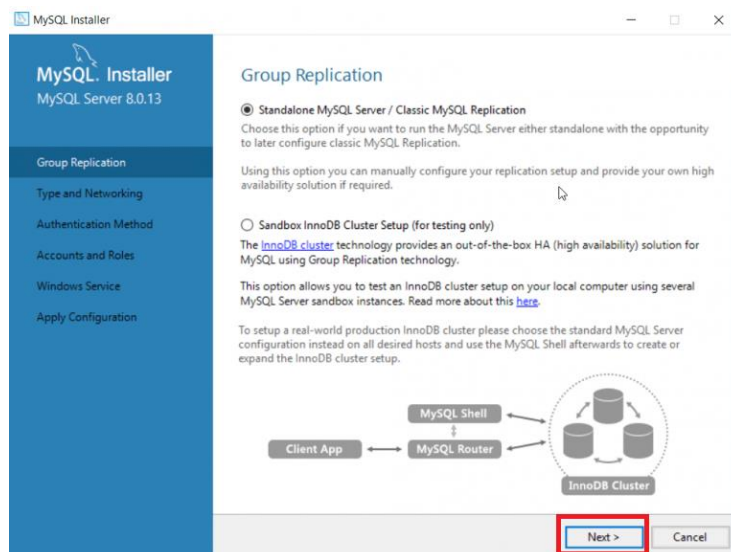


Install Directory adalah folder tempat file program MySQL sedangkan data directory adalah tempat file data seperti Database dan tabel berada.

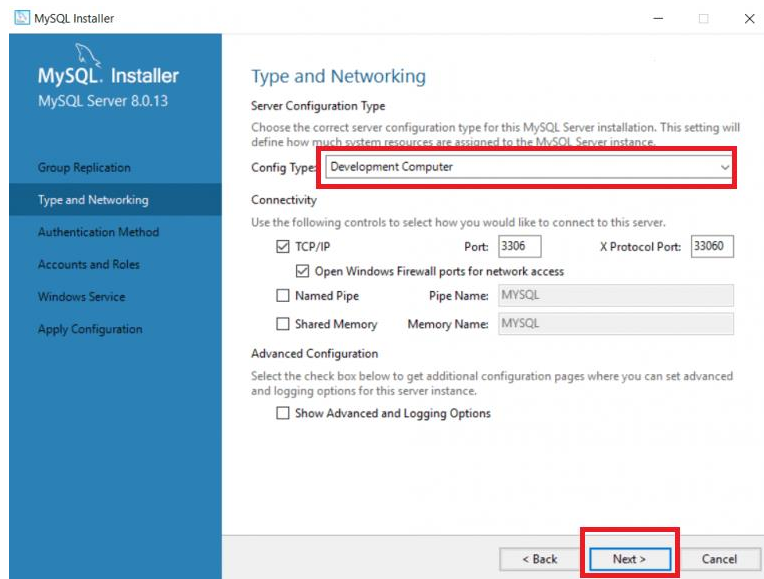
- Lalu pada tab insallation Klik excute untuk memulai proses instalasi MySQL Server



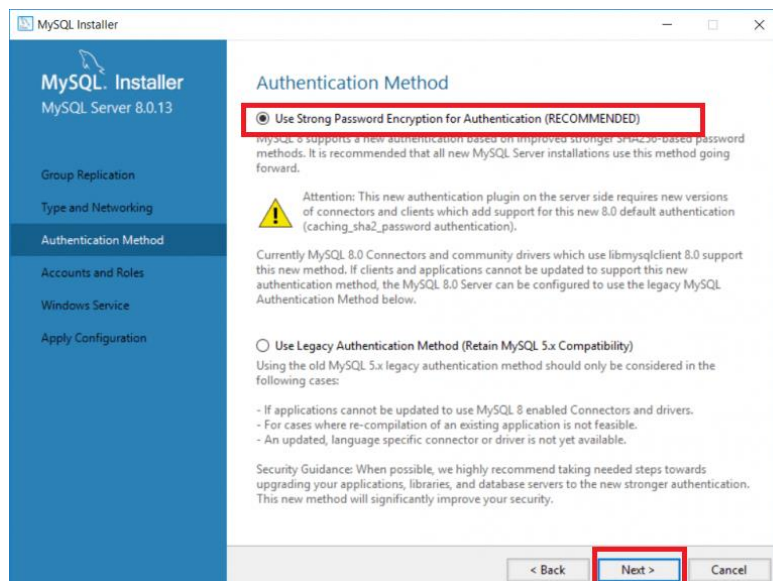
- Lalu akan masuk ke tahap konfigurasi awal seperti terlihat pada gambar



- Lalu pada tab Type and Networking pada config type ikuti saja bawaan pada computer seperti pada gambar

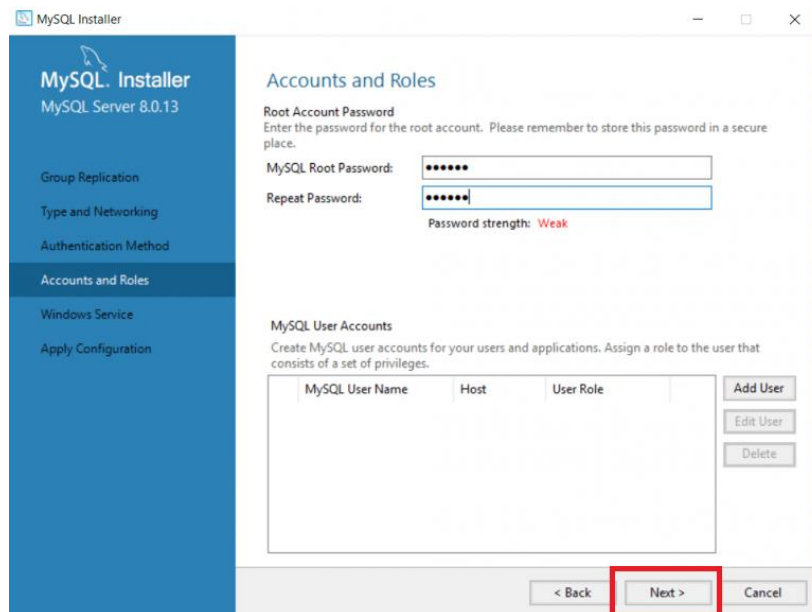


- Lalu ikuti seperti yang terlihat pada gambar

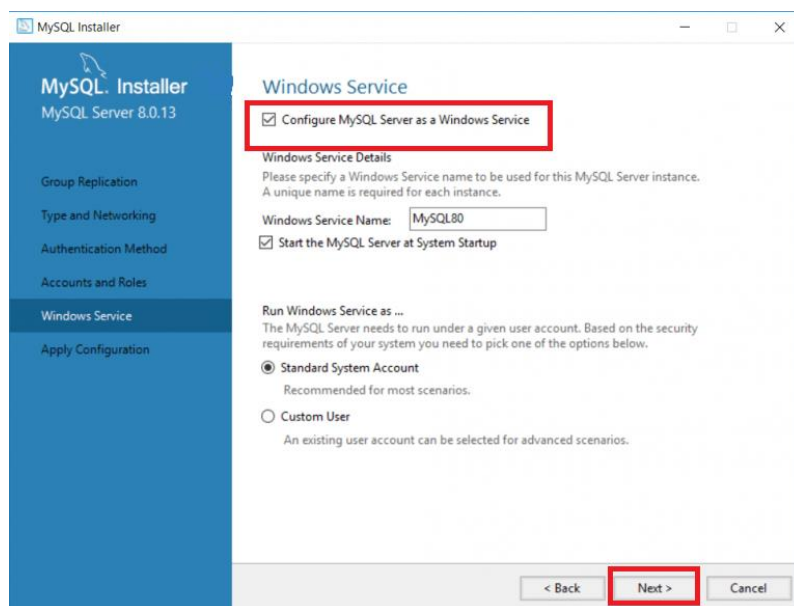


- Lalu pada tab Account and User role masukan pasword untuk user root. Root adalah user tertinggi pada MySQL. User root

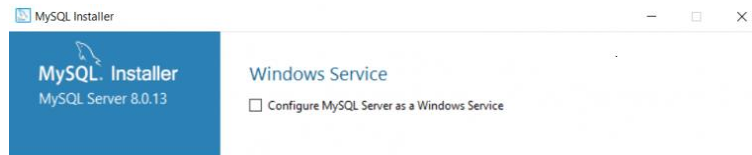
memiliki akses untuk megakses seluruh database yang ada pada computer. Setelah masukan pasword kemudian klik next



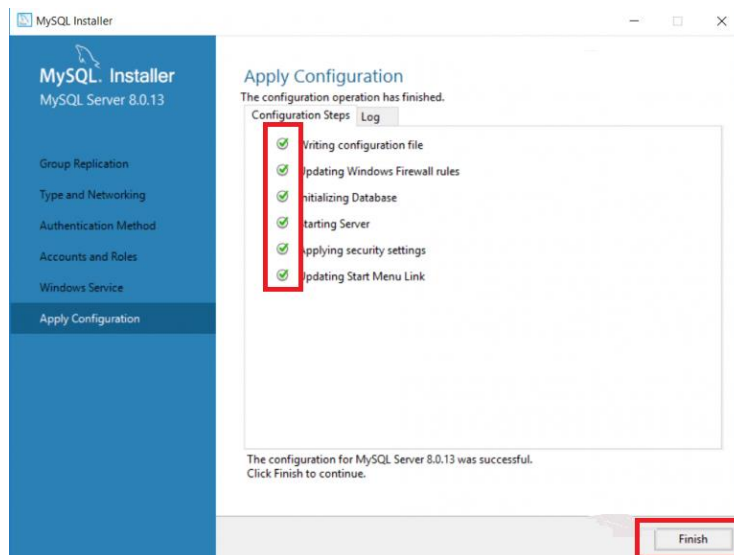
- Lalu pada tab Windows Service akan keluar seperti pada di gambar



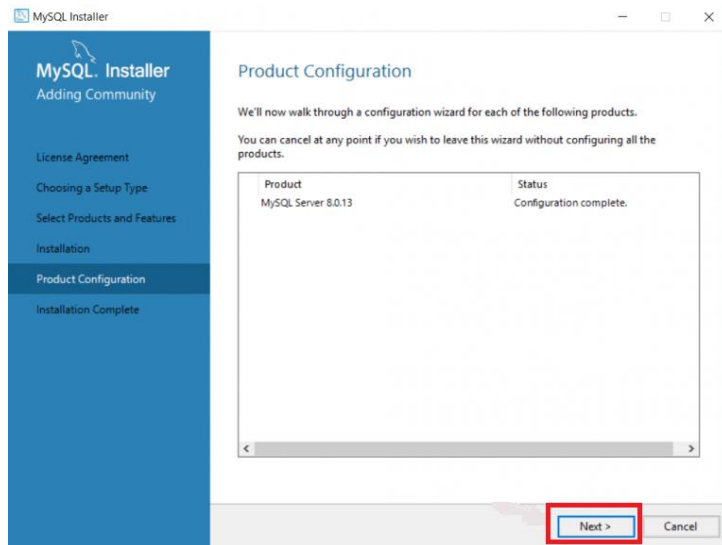
Tetapi pada “Configure MySQL Server as a Windows Service” biasanya di computer sudah otomatis di centang akan tetapi kita hilangkan centangnya karena jika di centang kita tidak dapat mengakses MySQL lebih dari 1 di dalam satu komputer.



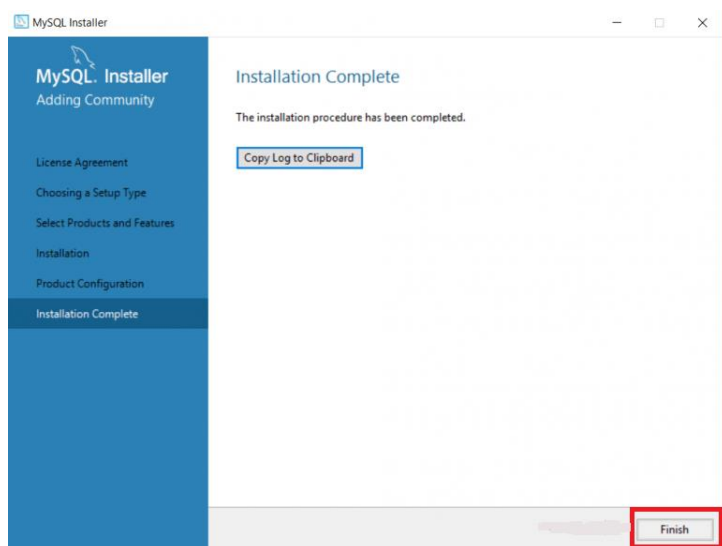
- Lalu pada tab terakhir konfigurasi ini, centang semua step yang ada di configuration steps lalu pilih excute



- Lalu akan kembali ke menu sebelumnya dan pada tab “Product Configuration” pilih next seperti pada gambar



- Lalu pada tab terakhir dari instalasi ini pilih finish



- Lalu selesai proses instalasi Server MySQL ini.