



LABORATORIUM  
TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS AHMAD DAHLAN



# PETUNJUK PRAKTIKUM

---

## BASIS DATA

**Penyusun:**

**Dewi Soyusiawaty, S.T., M.T.**

**Anna Hendri Soleliza Jones, S.Si., M.Cs.**

**Miftahurrahmah Rosyda, S.Kom., M.Eng.**

**Sri Winiarti, S.T., M.Cs.**

**Jefree Fahana, S.T., M.Kom.**

**2020**

## KATA PENGANTAR

Puji dan syukur kita panjatkan kepada Allah SWT yang telah memberi rahmat dan hidayahNya sehingga penyusunan revisi petunjuk praktikum Basis Data ini akhirnya bisa diselesaikan. Modul ini disusun sebagai panduan untuk pelaksanaan praktikum mata kuliah Basis Data di lingkungan Program Studi Teknik Informatika Universitas Ahmad Dahlan.

Materi yang disajikan sudah diurutkan disesuaikan dengan RPS berbasis Outcome Based Education (OBE) matakuliah basidata, sehingga mahasiswa dapat dengan mudah memahami. Pada setiap pertemuan diberikan penjelasan tentang teori terkait materi yang diberikan dan langkah praktikum berisi tahapan kegiatan yang harus dilakukan mahasiswa/praktikan pada saat praktikum.

Penulis menyadari masih banyak ketidaksempurnaan pada penulisan ini, baik isi maupun redaksinya, oleh karenanya kritik dan saran yang membangun diharapkan dapat memperbaiki untuk tahun-tahun berikutnya.

Terima kasih kepada semua pihak yang telah membantu baik secara langsung ataupun tidak terhadap terselesaikannya petunjuk praktikum ini. Akhir kata, insyaAllah dapat bermanfaat bagi siapa saja yang membutuhkannya.

Yogyakarta, 02 September 2020

Penyusun

## DAFTAR PENYUSUN

Dewi Soyusiawaty, S.T., M.T.

Anna Hendri Soleliza Jones, S.Si., M.Cs

Miftahurrahma Rosyda, S.Kom., M.Eng.

Sri Winiarti, S.T., M.Cs.

Jefree Fahana, S.T., M.Kom.

## HALAMAN REVISI

Yang bertanda tangan di bawah ini:

Nama : Jefree Fahana, S.T., M.Kom.

NIK/NIY : 60160979

Jabatan : Dosen

Dengan ini menyatakan pelaksanaan Revisi Petunjuk Praktikum Basis Data untuk Program Studi Teknik Informatika telah dilaksanakan dengan penjelasan sebagai berikut:

No	Keterangan Detail Revisi (Per Pertemuan)	Tanggal Revisi	Nomor Modul
1.	Menambahkan materi Normalisasi	15 Agustus 2015	PP/018/III/R2
2	Menambahkan materi Pengenalan Software Pemodelan data dan Pengolahan Basisdata	2 September 2020	PP/018/III/R3
	Menambahkan materi Trigger		

Yogyakarta, 2 September 2020

Koordinator Penyusun



**Jefree Fahana, S.T., M.Kom.**

NIY. 60160979

## HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Lisna Zahrotun, S.T., M.Cs.

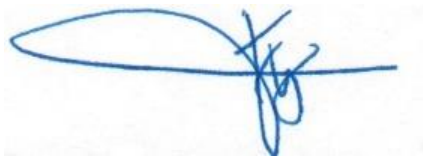
NIK/NIY : 60150773

Jabatan : Kepala Laboratorium Praktikum Teknik Informatika

Menerangkan dengan sesungguhnya bahwa Petunjuk Praktikum ini telah direview dan akan digunakan untuk pelaksanaan praktikum di Semester Gasal Tahun Akademik 2019/2020 di Laboratorium Praktikum Teknik Informatika, Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan.

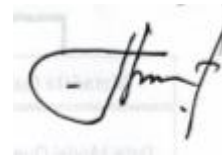
Yogyakarta, ... September 2020

Mengetahui,  
Ketua Kelompok Keilmuan Rekayasa Perangkat  
Lunak dan Data (RELATA)



**Drs., Tedy Setiadi, M.T.**  
NIY. 60030475

Kepala Laboratorium Praktikum Teknik  
Informatika



**Lisna Zahrotun, S.T., M.Cs.**  
NIY. 60150773

## VISI DAN MISI PRODI TEKNIK INFORMATIKA

### VISI

Menjadi Program Studi Informatika yang diakui secara internasional dan unggul dalam bidang Informatika serta berbasis nilai-nilai Islam.

### MISI

1. Menjalankan pendidikan sesuai dengan kompetensi bidang Informatika yang diakui nasional dan internasional
2. Meningkatkan penelitian dosen dan mahasiswa dalam bidang Informatika yang kreatif, inovatif dan tepat guna.
3. Meningkatkan kuantitas dan kualitas publikasi ilmiah tingkat nasional dan internasional
4. Melaksanakan dan meningkatkan kegiatan pengabdian masyarakat oleh dosen dan mahasiswa dalam bidang Informatika.
5. Menyelenggarakan aktivitas yang mendukung pengembangan program studi dengan melibatkan dosen dan mahasiswa.
6. Menyelenggarakan kerja sama dengan lembaga tingkat nasional dan internasional.
7. Menciptakan kehidupan Islami di lingkungan program studi.

## TATA TERTIB LABORATORIUM TEKNIK INFORMATIKA

### DOSEN/KOORDINATOR PRAKTIKUM

1. Dosen harus hadir saat praktikum minimal 15 menit di awal kegiatan praktikum dan menandatangani presensi kehadiran praktikum.
2. Dosen membuat modul praktikum, soal seleksi asisten, pre-test, post-test, dan responsi dengan berkoordinasi dengan asisten dan pengampu mata praktikum.
3. Dosen berkoordinasi dengan koordinator asisten praktikum untuk evaluasi praktikum setiap minggu.
4. Dosen menandatangani surat kontrak asisten praktikum dan koordinator asisten praktikum.
5. Dosen yang tidak hadir pada slot praktikum tertentu tanpa pemberitahuan selama 2 minggu berturut-turut mendapat teguran dari Kepala Laboratorium, apabila masih berlanjut 2 minggu berikutnya maka Kepala Laboratorium berhak mengganti koordinator praktikum pada slot tersebut.

### PRAKTIKAN

1. Praktikan harus hadir 15 menit sebelum kegiatan praktikum dimulai, dan dispensasi terlambat 15 menit dengan alasan yang jelas (kecuali asisten menentukan lain dan patokan jam adalah jam yang ada di Laboratorium, terlambat lebih dari 15 menit tidak boleh masuk praktikum & dianggap INHAL).
2. Praktikan yang tidak mengikuti praktikum dengan alasan apapun, wajib mengikuti INHAL, maksimal 4 kali praktikum dan jika lebih dari 4 kali maka praktikum dianggap GAGAL.
3. Praktikan harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
  - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
  - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
  - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
  - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Praktikan tidak boleh makan dan minum selama kegiatan praktikum berlangsung, harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di dalam laboratorium (tidak boleh membuang sampah sembarangan baik kertas, potongan kertas, bungkus permen baik di lantai karpet maupun di dalam ruang CPU).
5. Praktikan dilarang meninggalkan kegiatan praktikum tanpa seizin Asisten atau Laboran.
6. Praktikan harus meletakkan sepatu dan tas pada rak/loker yang telah disediakan.
7. Selama praktikum dilarang NGENET/NGE-GAME, kecuali mata praktikum yang membutuhkan atau menggunakan fasilitas Internet.
8. Praktikan dilarang melepas kabel jaringan atau kabel power praktikum tanpa sepengetahuan laboran
9. Praktikan harus memiliki FILE Petunjuk praktikum dan digunakan pada saat praktikum dan harus siap sebelum praktikum berlangsung.
10. Praktikan dilarang melakukan kecurangan seperti mencontek atau menyalin pekerjaan praktikan yang lain saat praktikum berlangsung atau post-test yang menjadi tugas praktikum.
11. Praktikan dilarang mengubah setting software/hardware komputer baik menambah atau mengurangi tanpa permintaan asisten atau laboran dan melakukan sesuatu yang dapat merugikan laboratorium atau praktikum lain.

12. Asisten, Koordinator Praktikum, Kepala laboratorium dan Laboran mempunyai hak untuk menegur, memperingatkan bahkan meminta praktikan keluar ruang praktikum apabila dirasa anda mengganggu praktikan lain atau tidak melaksanakan kegiatan praktikum sebagaimana mestinya dan atau tidak mematuhi aturan lab yang berlaku.
13. Pelanggaran terhadap salah satu atau lebih dari aturan diatas maka Nilai praktikum pada pertemuan tersebut dianggap 0 (NOL) dengan status INHAL.

### ASISTEN PRAKTIKUM

1. Asisten harus hadir 15 Menit sebelum praktikum dimulai (konfirmasi ke koordinator bila mengalami keterlambatan atau berhalangan hadir).
2. Asisten yang tidak bisa hadir WAJIB mencari pengganti, dan melaporkan kepada Koordinator Asisten.
3. Asisten harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
  - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
  - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
  - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
  - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Asisten harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di laboratorium, menegur atau mengingatkan jika ada praktikan yang tidak dapat menjaga kebersihan, ketertiban atau kesopanan.
5. Asisten harus dapat merapikan dan mengamankan presensi praktikum, Kartu Nilai serta tertib dalam memasukan/Input nilai secara Online/Offline.
6. Asisten harus dapat bertindak secara profesional sebagai seorang asisten praktikum dan dapat menjadi teladan bagi praktikan.
7. Asisten harus dapat memberikan penjelasan/pemahaman yang dibutuhkan oleh praktikan berkenaan dengan materi praktikum yang diasistensi sehingga praktikan dapat melaksanakan dan mengerjakan tugas praktikum dengan baik dan jelas.
8. Asisten tidak diperkenankan mengobrol sendiri apalagi sampai membuat gaduh.
9. Asisten dimohon mengkoordinasikan untuk meminta praktikan agar mematikan komputer untuk jadwal terakhir dan sudah dilakukan penilaian terhadap hasil kerja praktikan.
10. Asisten wajib untuk mematikan LCD Projector dan komputer asisten/praktikan apabila tidak digunakan.
11. Asisten tidak diperkenankan menggunakan akses internet selain untuk kegiatan praktikum, seperti Youtube/Game/Medsos/Streaming Film di komputer praktikan.

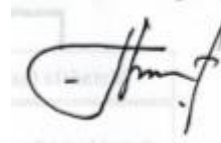
### LAIN-LAIN

1. Pada Saat Responsi Harus menggunakan Baju Kemeja untuk Laki-laki dan Perempuan untuk Praktikan dan Asisten.
2. Ketidakhadiran praktikum dengan alasan apapun dianggap INHAL.
3. Izin praktikum mengikuti aturan izin SIMERU/KULIAH.
4. Yang tidak berkepentingan dengan praktikum dilarang mengganggu praktikan atau membuat keributan/kegaduhan.
5. Penggunaan lab diluar jam praktikum maksimal sampai pukul 21.00 dengan menunjukkan surat ijin dari Kepala Laboratorium Prodi Teknik Informatika.



Yogyakarta, 23 Agustus 2019

Kepala Laboratorium Praktikum Teknik  
Informatika

A handwritten signature in black ink, appearing to read 'Lisna Zahrotun', is written over a faint, rectangular grid pattern.

**Lisna Zahrotun, S.T., M.Cs.**

NIY. 60150773

## DAFTAR ISI

KATA PENGANTAR.....	1
DAFTAR PENYUSUN.....	2
HALAMAN REVISI.....	3
HALAMAN PERNYATAAN.....	4
VISI DAN MISI PRODI TEKNIK INFORMATIKA .....	5
TATA TERTIB LABORATORIUM TEKNIK INFORMATIKA.....	6
DAFTAR ISI .....	9
PRAKTIKUM 1: PENGENALAN TOOLS PEMODELAN DATA.....	10
PRAKTIKUM 2: PEMAHAMAN KASUS DENGAN IDENTIFIKASI ENTITAS .....	18
PRAKTIKUM 3: PENGOLAHAN DATA.....	25
PRAKTIKUM 4: KONVERSI LOGICAL MODEL KE RELATIONAL MODEL .....	32
PRAKTIKUM 5: RELASI TABEL.....	41
PRAKTIKUM 6: NORMALISASI & BUSSINESS RULES .....	46
PRAKTIKUM 7: DDL dan DML.....	54
PRAKTIKUM 8: FUNGSI AGREGASI .....	63
PRAKTIKUM 9: QUERY DARI RELASI TABEL.....	67
PRAKTIKUM 10: RELASI TABEL DENGAN JOIN .....	72
PRAKTIKUM 11: SUBQUERY .....	77
PRAKTIKUM 12: TRIGGER.....	82
PRAKTIKUM 13: PHP dan MySQL.....	87
DAFTAR PUSTAKA.....	97

## PRAKTIKUM 1: PENGENALAN TOOLS PEMODELAN DATA

---

Pertemuan ke	: 1
Total Alokasi Waktu	: 180 menit
• Pre-Test	: 30 menit
• Praktikum	: 120 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 50 %
• Post-Test	: 30 %

---

### 1.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu menerapkan:

1. Mengetahui Tools yang digunakan dalam Basis Data
2. Dapat Menginstall XAMPP

Indikator ketercapaian diukur dengan:

1. Praktikan mampu mengenal beberapa *tools* yang digunakan untuk pemodelan data
2. Praktikan mampu menginstall XAMPP pada laptop masing-masing

### 1.2. TEORI PENDUKUNG

#### 1. XAMPP

Xampp merupakan aplikasi cross platform: Apache, MySQL, PHP dan Perl. XAMPP juga memberikan solusi sederhana dan cukup ringan untuk dijalankan, memungkinkan Anda membuat web server local untuk melakukan pengetesan website. Xampp dapat dijalankan pada Mac, Windows dan Linux.

#### 2. TOOLS Pemodelan Data

Alat pemodelan data membantu kita membuat struktur database dari diagram ini. Karenanya menjadi lebih mudah untuk menghubungkan data dan membentuk struktur data yang sempurna sesuai kebutuhan kami.

Alat yang berbeda mendukung sistem operasi yang berbeda dan sebagian besar pemodel data mendukung sistem operasi Windows. Sementara beberapa mendukung Mac dan Linux. Selain itu, alat yang berbeda mendukung database yang berbeda.

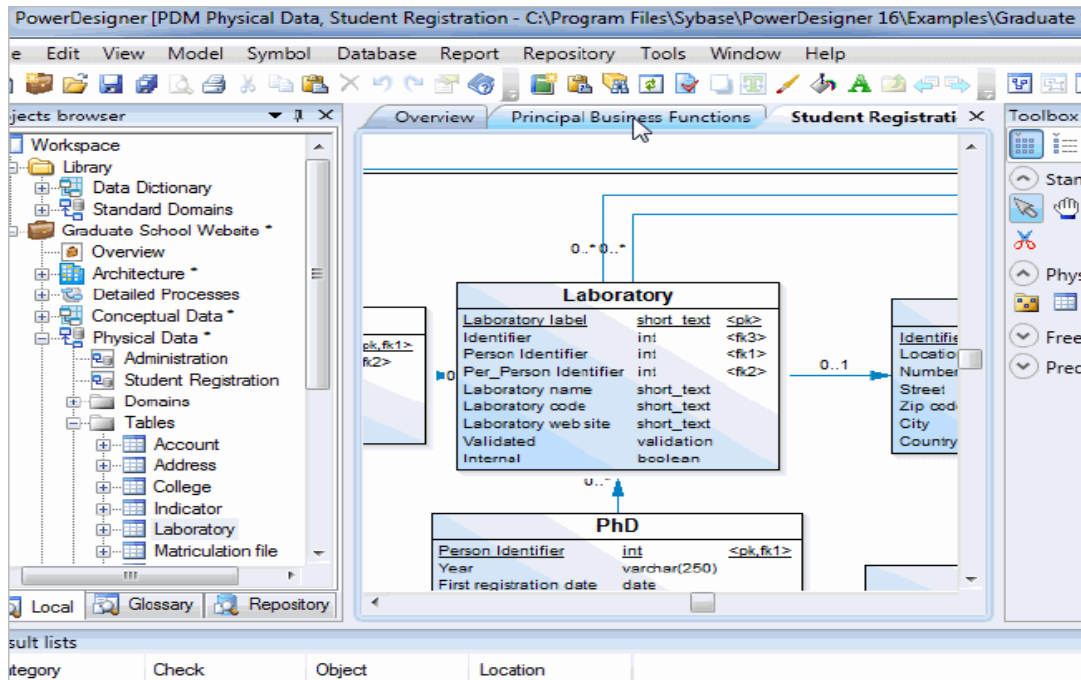
Alat-alat ini menawarkan fitur-fitur seperti pembuatan struktur data dari diagram, rekayasa maju & mundur, fasilitas impor & ekspor, dokumentasi, dukungan untuk banyak database, pelaporan, dll. Beberapa alat dapat digunakan secara online juga.

Beberapa alat model data dapat diintegrasikan dengan platform data besar seperti MongoDB atau Hadoop Hive. Alat-alat ini juga bisa disebut sebagai alat pemodelan data besar. Contoh alat semacam itu adalah ER / Studio.

Pemodelan data di gudang tidak lain adalah menggunakan model data untuk mendesain database secara konseptual, logis, dan fisik. Demikian pula, mereka digunakan untuk membuat tabel relasional, kunci utama & asing, dan prosedur tersimpan.

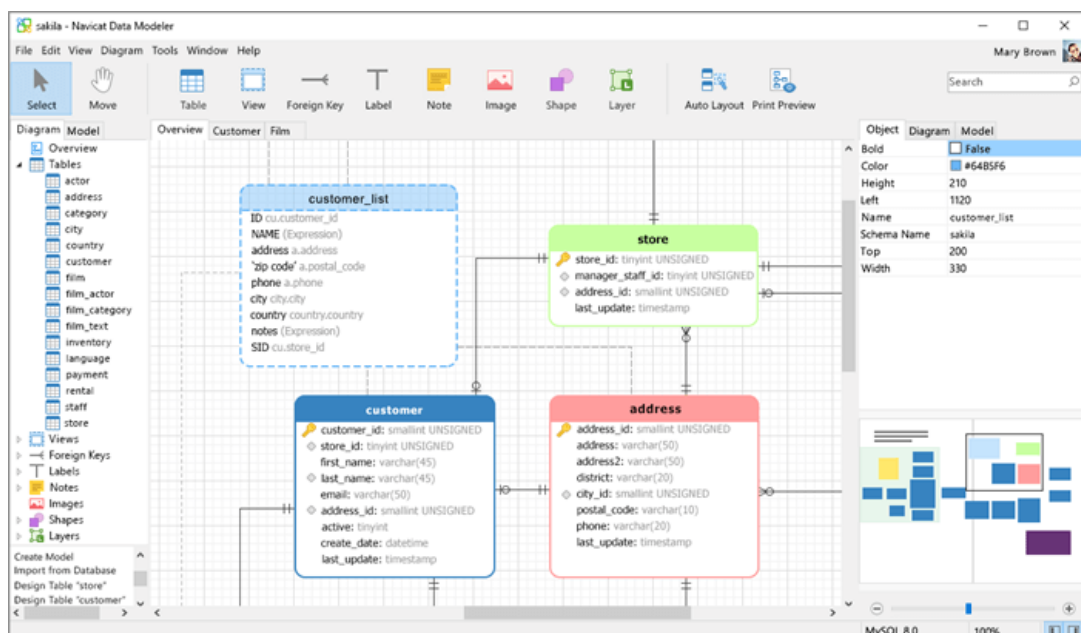
Dalam melakukan praktikum basis data ini, mahasiswa perlu mengenal beberapa tool pemodelan data baik yang berbayar maupun free. Berikut Beberapa Tools Pemodelan Data

### 1. Power Designer (Berjalan di windows)



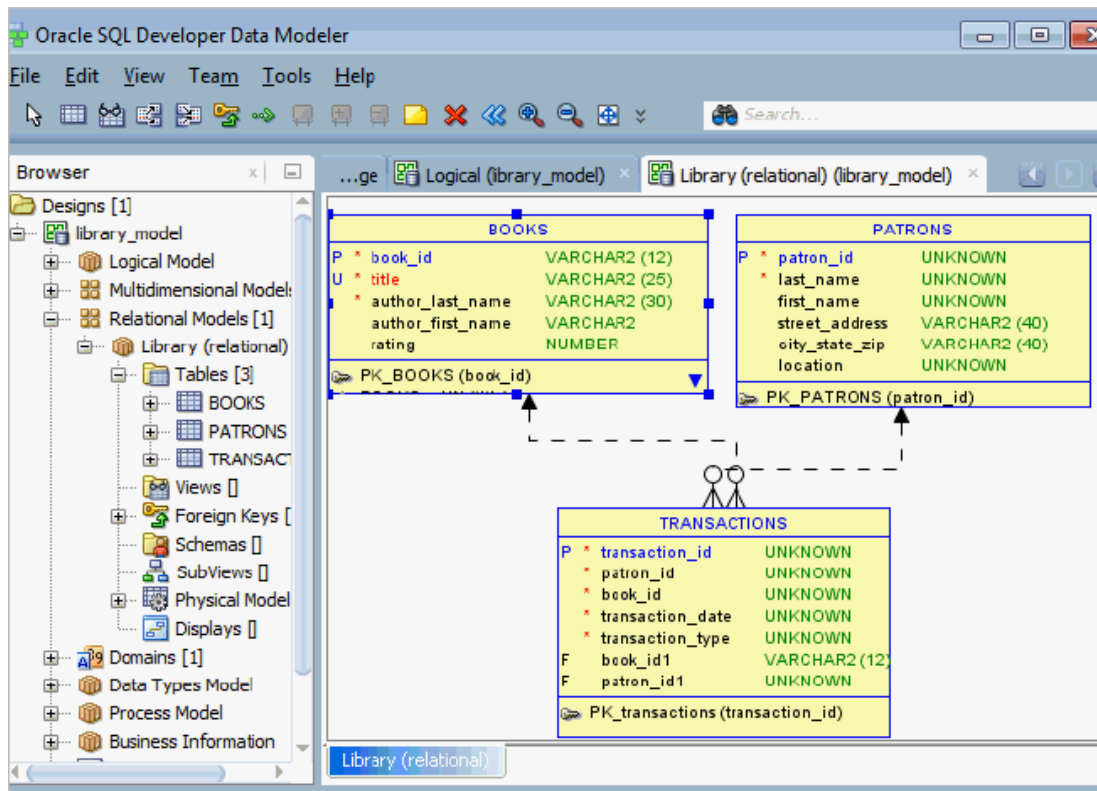
Gambar 1. 1. Antarmuka PowerDesigner

### 2. Navicat Data Modeller (Berjalan pada cross-platform)



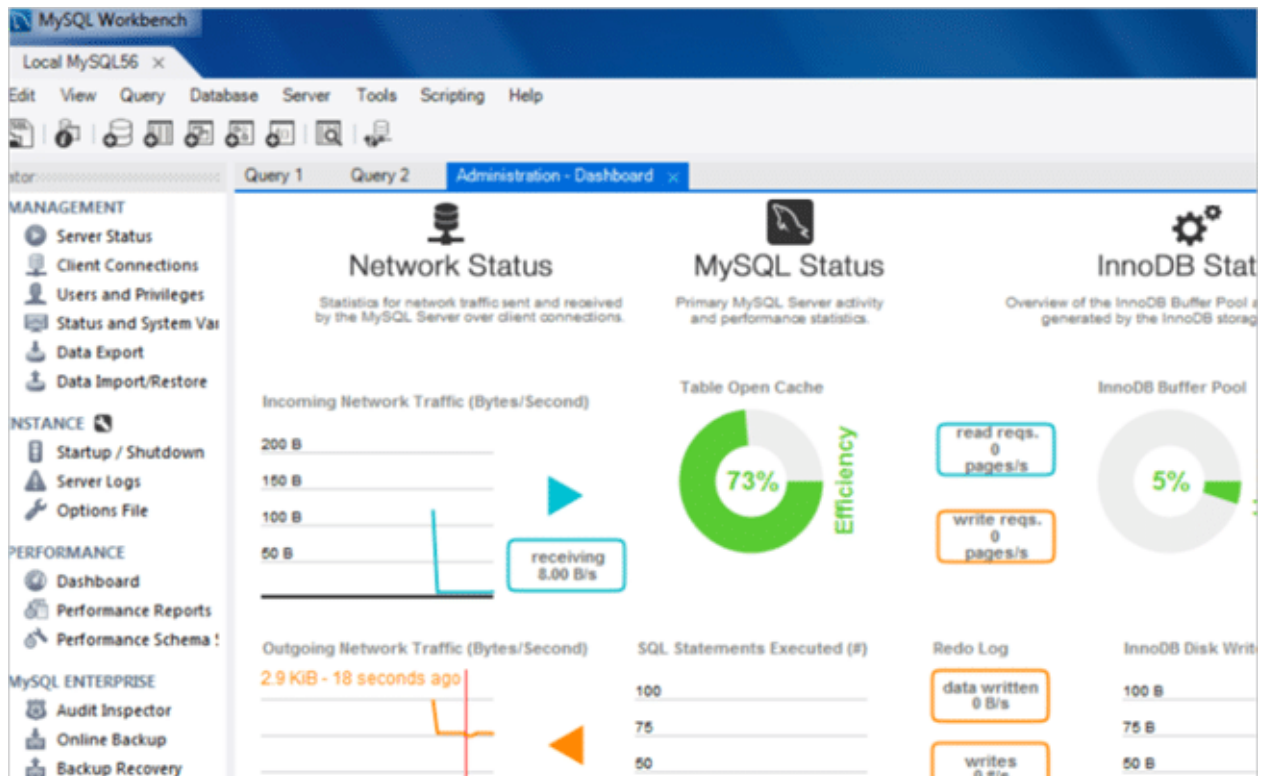
Gambar 1. 2. Antarmuka Navicat

### 3. Oracle SQL Data Modeller (Berjalan pada cross-platform)



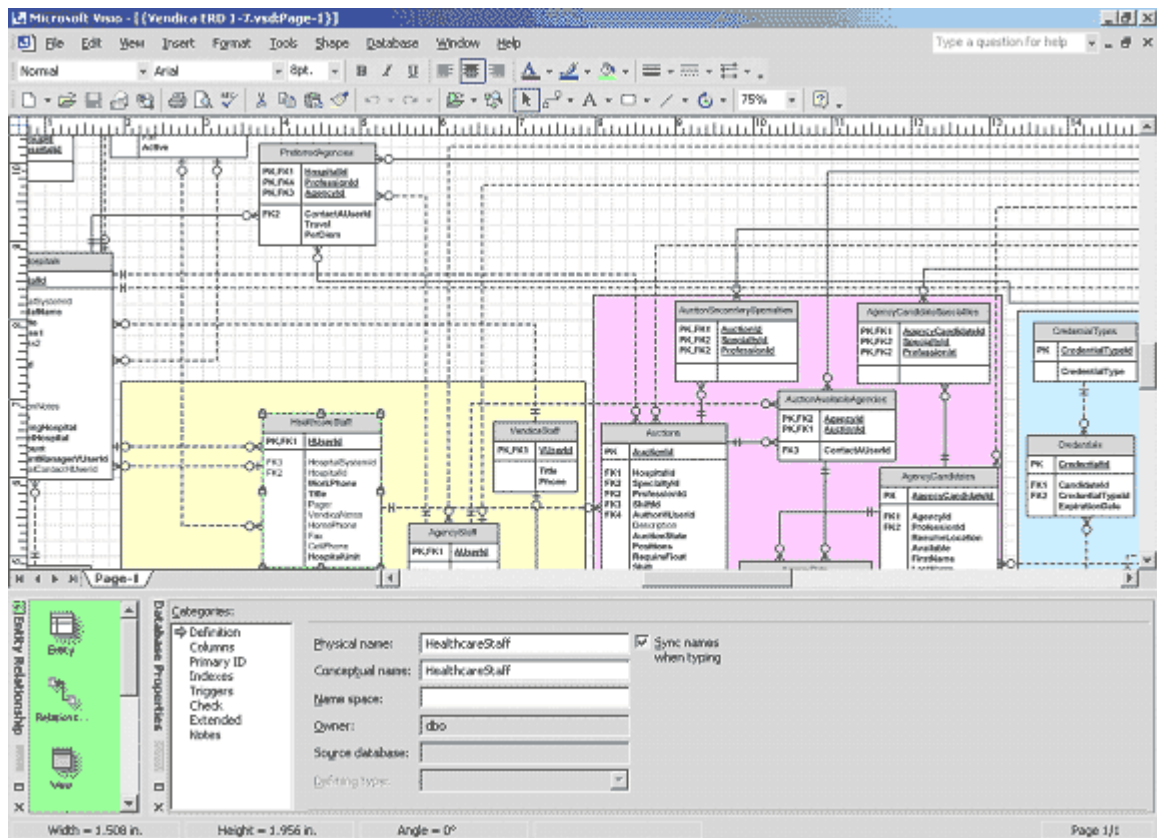
Gambar 1. 3. Antarmuka Oracle

### 4. MySQL WorkBench



Gambar 1. 4. Antarmuka MySQL

## 5. Microsoft Visio (Berjalan pada windows)



Gambar 1. 5. Antarmuka Visio

### 1.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. XAMPP.

### 1.4. LANGKAH PRAKTIKUM

#### Menginstall XAMPP

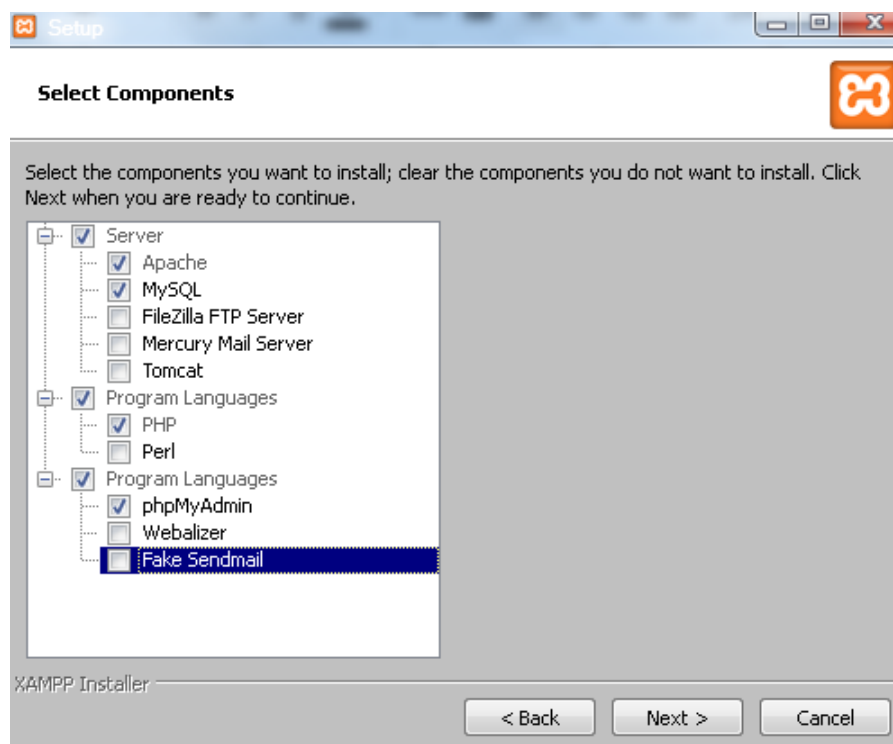
1. Download XAMPP dari link yang telah diberikan
2. Double click file XAMPP yang telah didownload



Gambar 1. 6 Setup XAMPP

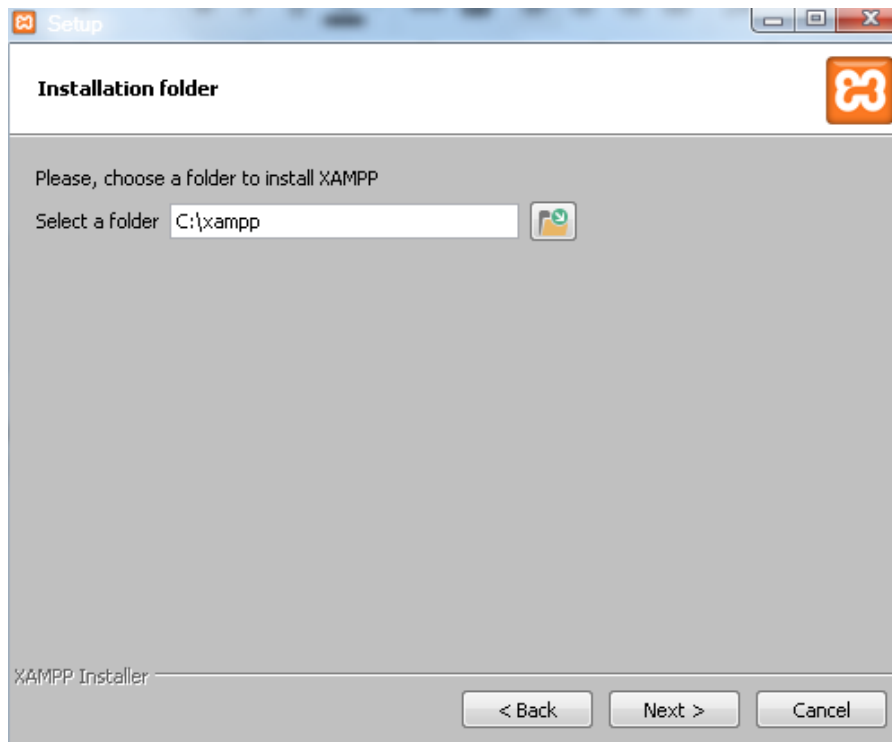
klik next

2. Centang komponen yang diperlukan lalu klik next



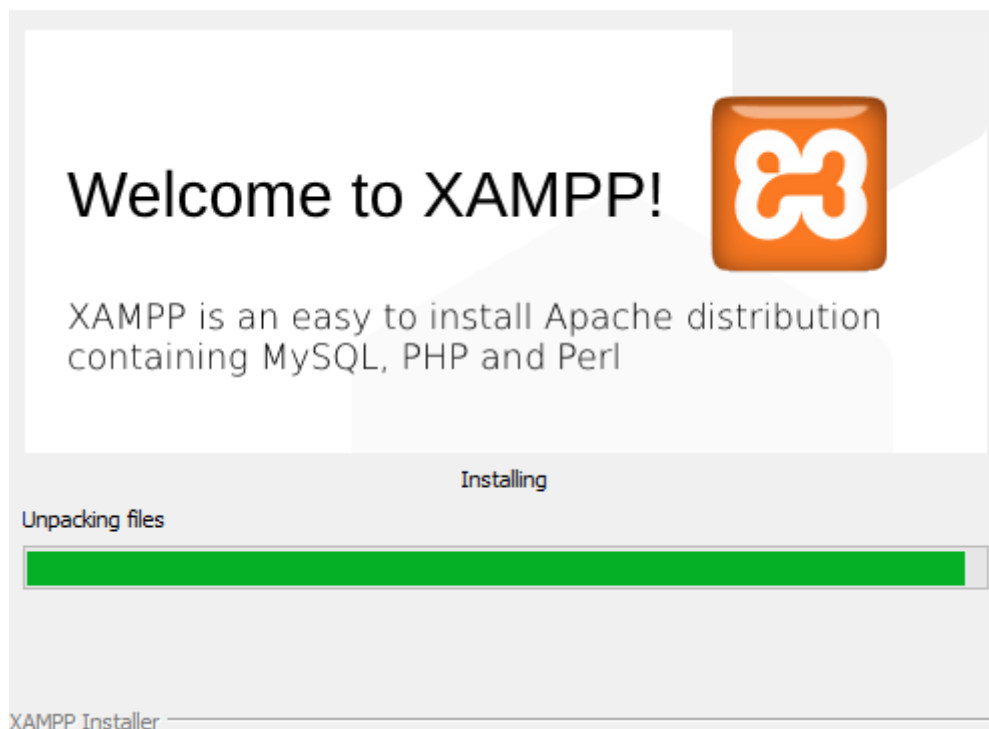
Gambar 1. 7. Pilih Komponen

3. Letakkan folder instalasi Xampp pada Drive : C (diutamakan)



Gambar 1. 8. Meletakkan Folder Instalasi

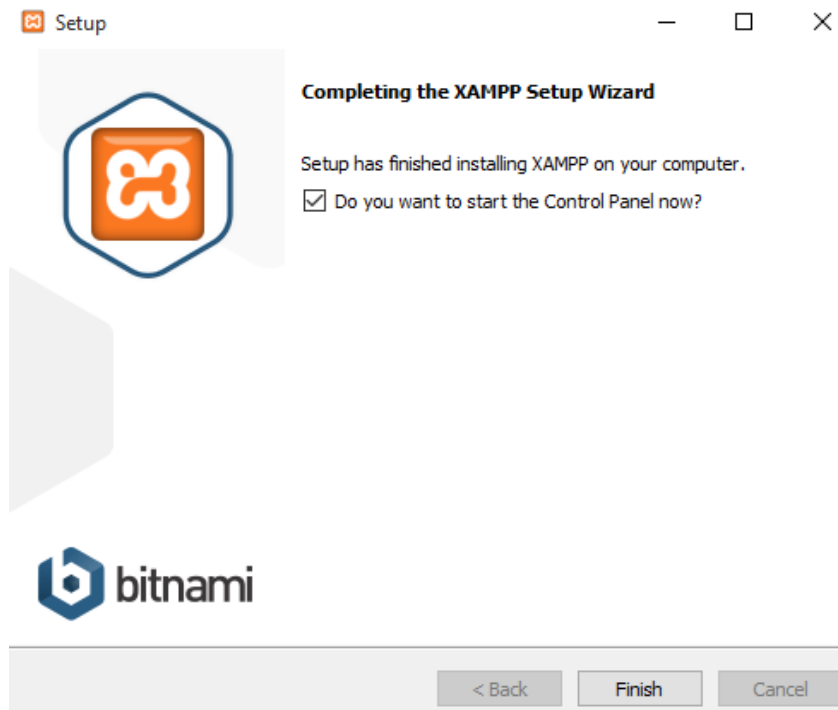
#### 4. Proses Instalasi



Gambar 1. 9. Proses Instalasi

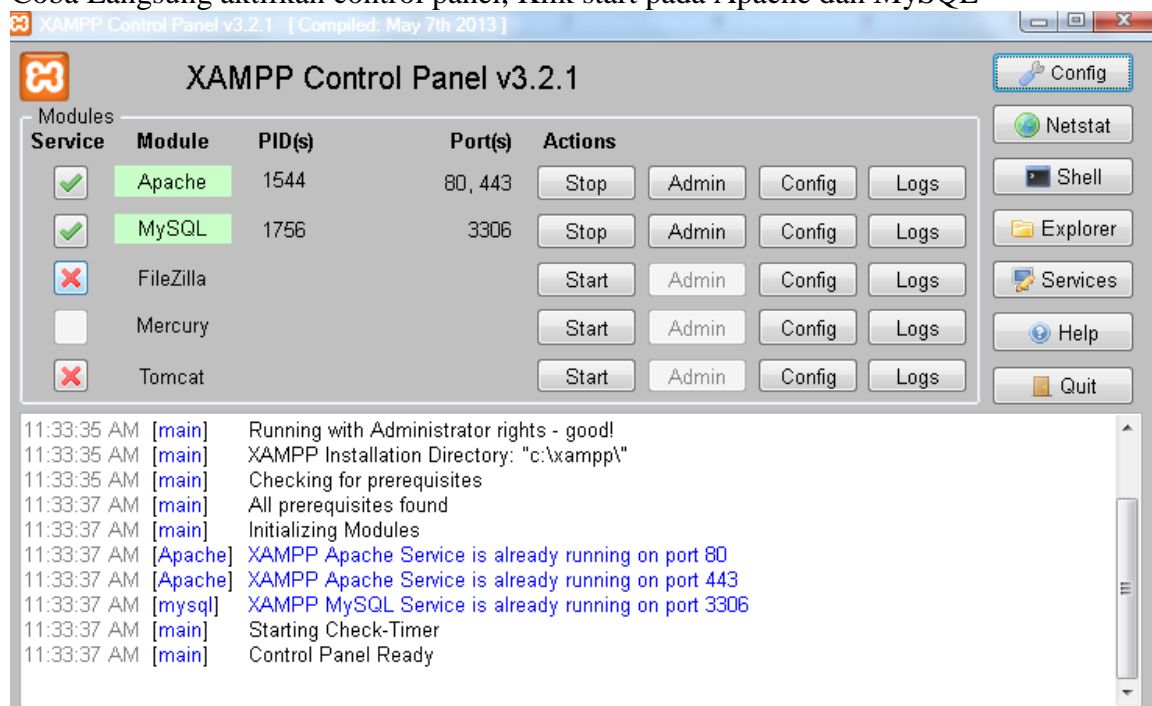
#### 5. Selesai Instalasi





Gambar 1. 10. Selesai Instalasi

6. Coba Langsung aktifkan control panel, Klik start pada Apache dan MySQL



Gambar 1. 11. Masuk Control Panel XAMPP

## 1.5. TUGAS

Tugas diberikan oleh asisten praktikum.

Pastikan XAMPP terinstall dengan baik pada laptop Anda

**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 1: PEMAHAMAN KASUS  
DENGAN IDENTIFIKASI ENTITAS**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--

## PRAKTIKUM 2: PEMAHAMAN KASUS DENGAN IDENTIFIKASI ENTITAS

---

Pertemuan ke	: 2
Total Alokasi Waktu	: 180 menit
• Pre-Test	: 30 menit
• Praktikum	: 120 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 50 %
• Post-Test	: 30 %

---

### 2.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu menerapkan:

1. *Create database*
2. *Create table*

Indikator ketercapaian diukur dengan:

1. Praktikan mampu mengenal *MySQL* dan *Database Management System*.
2. Praktikan mampu memahami tabel, tipe data dan primary key dalam *Database Management System*.
3. Praktikan mampu mengidentifikasi entitas pada suatu kasus yang disediakan.

### 2.2. TEORI PENDUKUNG

#### 1. Tabel dan Atribut

Entitas atau tabel merupakan individu atau objek yang memiliki sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Atribut adalah karakteristik atau ciri yang mendefinisikan entitas sehingga dapat membedakan antara entitas satu dengan entitas lainnya. Dalam sebuah entitas pasti memiliki satu atau lebih atribut untuk mendefinisikan karakteristik dari entitas tersebut. Contoh atribut dari Entitas Mahasiswa yaitu nama, nim, kelas.

#### 2. Mengenal MySQL

MySQL adalah Sebuah program database server yang mampu menerima dan mengirimkan datanya sangat cepat, multi user serta menggunakan perintah dasar SQL (Structured Query Language). MySQL merupakan dua bentuk lisensi, yaitu Free Software dan Shareware. MySQL yang biasa kita gunakan adalah MySQL Free Software yang berada dibawah Lisensi GNU/GPL (General Public License). MySQL merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.

MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius. Selain database server, MySQL juga merupakan program yang dapat mengakses suatu database MySQL yang berposisi sebagai Server, yang berarti program kita berposisi sebagai Client. Jadi MySQL adalah sebuah

database yang dapat digunakan sebagai Client maupun server. Database MySQL merupakan suatu perangkat lunak database yang berbentuk database relasional atau disebut Relational Database Management System (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama SQL (Structured Query Language).

Database MySQL memiliki beberapa kelebihan dibanding database lain, antara lain :

- a. MySQL merupakan Database Management System ( DBMS ).
- b. MySQL sebagai Relation Database Management System ( RDBMS ) atau disebut dengan database Relational.
- c. MySQL Merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.
- d. MySQL merupakan sebuah database client.
- e. MySQL mampu menerima query yang bertupuk dalam satu permintaan atau Multi Threading.
- f. MySQL merupakan Database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran GigaByte sekalipun.
- g. MySQL didukung oleh driver ODBC, artinya database MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Visual Basic, PHP dan Delphi.
- h. MySQL merupakan Database Server yang multi user, artinya database ini tidak hanya digunakan oleh satu pihak orang akan tetapi dapat digunakan oleh banyak pengguna.
- i. MySQL mendukung field yang dijadikan sebagai kunci primer dan kunci unik (Unique).
- j. MySQL memiliki kecepatan dalam pembuatan table maupun update table.

### 3. Pemahaman Kasus pada Sistem Informasi Perpustakaan

Saat meminjam buku di perustakaan, seseorang akan melakukan registrasi terlebih dahulu. Kemudian, data peminjam akan disimpan oleh sistem dalam basis data. Setelah itu baru dapat dilakukan berbagai transaksi yaitu peminjaman buku, perpanjangan peminjaman buku, perpanjangan peminjaman buku, pengembalian buku, dan pembayaran denda. Seluruh transaksi yang dilakukan disimpan dalam basis data dan dapat diakses kembali oleh sistem sesuai dengan kebutuhan.

Peminjaman dapat mencari buku yang akan dipinjam dengan menekan tombol find atau search. Setelah itu sistem informasi akan berinteraksi dengan basis data perpustakaan untuk mencari buku tersebut. Lalu hasil pencarian akan ditampilkan pada layar. Melalui interaksi yang terjadi antara sistem informasi dengan basis data, petugas perpustakaan dapat mengetahui secara otomatis jika peminjaman buku lebih dari ketentuan yang berlaku, peminjam belum mengembalikan buku tetapi akan meminjam lagi buku yang lain, dan pemberitahuan ada yang kena denda pada hari H. Data yang disimpan dalam basis data meliputi :

Data buku, misalnya kode buku, judul buku, pengarang, penerbit, dan tahun terbit.	Data transaksi perpanjangan peminjaman, misal tgl perpanjangan, kode buku, jml buku diperpanjang, no peminjam, perpanjangan ke berapa, dan alasan
Data peminjam, misalnya nomor peminjam, nama peminjam, alamat, nomor telepon, dan email	
Data petugas perpustakaan, misalnya nomor induk karyawan, jabatan dan level.	Data transaksi pengembalian buku, misal tgl pengembalian, kode buku, jumlah buku yang dikembalikan, nomor peminjam.

Data transaksi peminjaman, misalnya tanggal pinjam, kode buku, jumlah buku, nomor peminjam, dan lama pinjaman.	Data transaksi pemb denda, misal tgl denda, kode buku, jml buku yang kena denda, nomor peminjam, status lunas.
--	--

### 2.3. ALAT DAN BAHAN

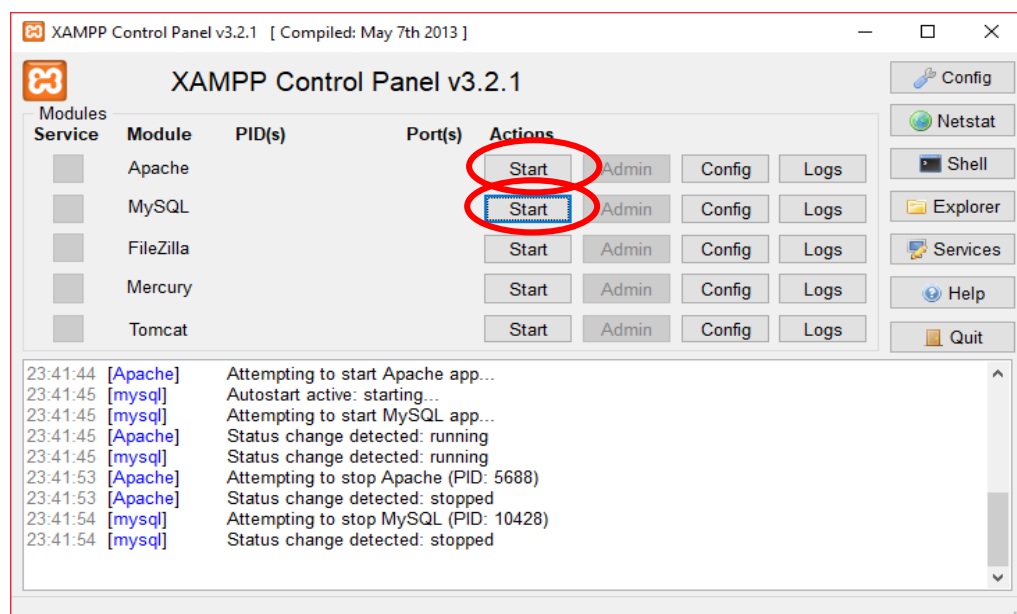
Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. XAMPP.
3. Browser (*mozilla firefox, chrome*).

### 2.4. LANGKAH PRAKTIKUM

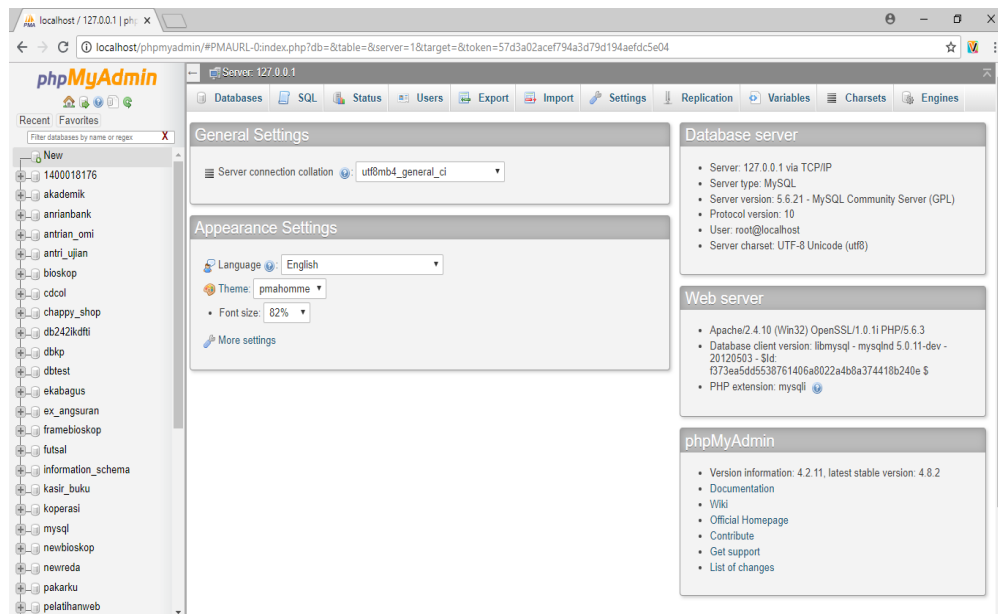
#### 1. Mengakses PHPMyAdmin dengan XAMPP

Jalankan XAMPP Control Panel hingga muncul jendela aplikasi XAMPP Control Panel. Kemudian klik tombol Start pada modul Apache dan MySQL. Modul Apache digunakan untuk mengakses PHPMyAdmin pada browser sehingga lebih mudah dalam mengakses MySQL karena menggunakan GUI (Graphical User Interface). Modul MySQL digunakan untuk melayani request atau query yang diterima dari PHPMyAdmin.



Gambar 2.1. XAMPP

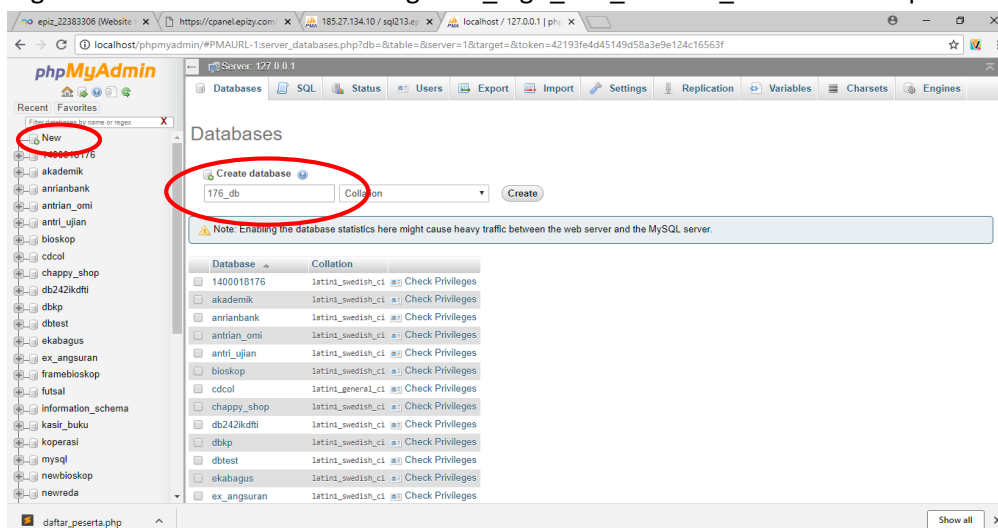
Kemudian membuka browser (Chrome, Mozilla, Opera, dll) dan mengaksesnya dengan mengetikkan "localhost/phpmyadmin" pada kolom isian URL, sehingga muncul seperti pada gambar di bawah ini.



Gambar 2.2. PHP MyAdmin

## 2. Membuat Database

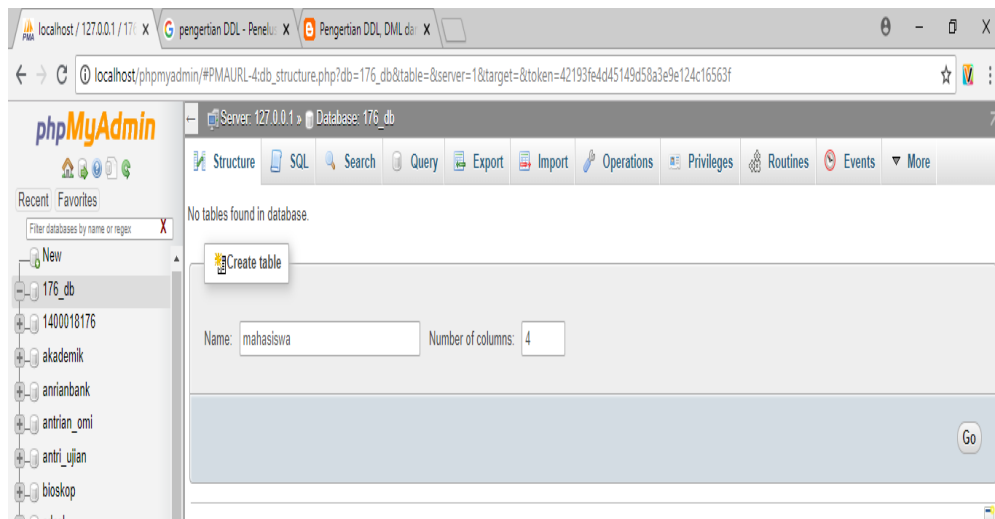
Pilih menu “new” kemudian isi nama database pada kolom yang sudah disediakan, untuk keseragaman nama database diisi dengan “3\_digit\_nim\_terakhir\_db” kemudian pilih “create”.



Gambar 2.3. Create Database

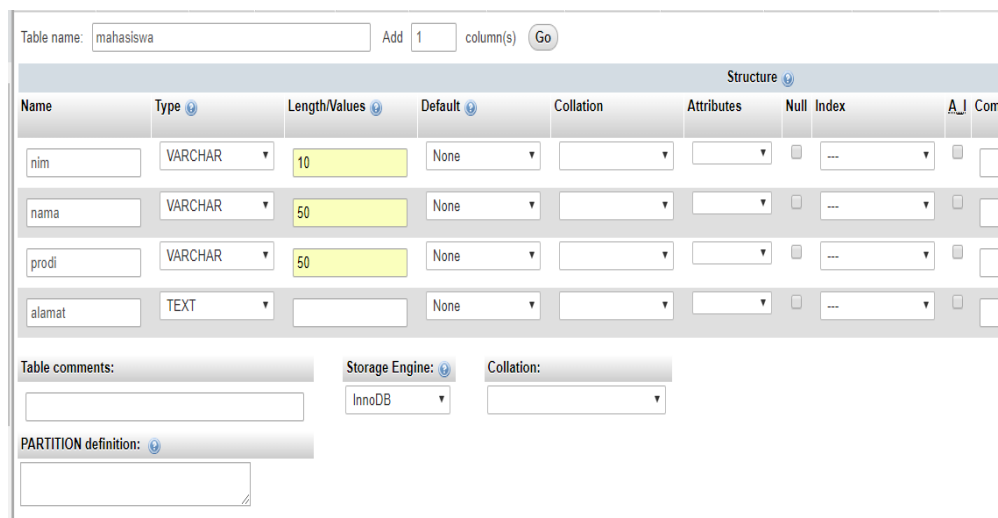
## 3. Membuat Tabel

Membuat tabel dengan nama tabel “mahasiswa” yang memiliki atribut nim, nama, program studi dan alamat. Langkahnya yaitu pilih menu database yang hingga muncul gambar seperti di bawah ini.



Gambar 2.4. Create Table

Kemudian mengisi form untuk memberi atribut pada tabel “mahasiswa” yang sudah dibuat. Pada entitas ini, atribut nim akan digunakan sebagai pembeda pada data yang akan dimasukkan ke dalam tabel atau sering disebut **primary key**. Atribut nim, nama dan program studi diidentifikasi dengan tipe VARCHAR dengan panjang karakter yang disesuaikan dengan keinginan programmer. Sedangkan atribut alamat diidentifikasi dengan tipe TEXT karena memiliki jumlah karakter yang lebih panjang.



Gambar 2.5. Isi Data Table

Kemudian pada atribut nim, karena sebagai primary key maka pada index diisi dengan PRIMARY kemudian pilih “Save”.

Table name: mahasiswa Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments
nim	VARCHAR	10	None			<input type="checkbox"/>	PRIMARY	<input type="checkbox"/>	
nama	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
prodi	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
alamat	TEXT		None			<input type="checkbox"/>	---	<input type="checkbox"/>	

Table comments:

Storage Engine: InnoDB Collation:

PARTITION definition:

Gambar 2.6. Primary Key

## 2.5. TUGAS

Tugas diberikan oleh asisten praktikum.



**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 2: PEMAHAMAN KASUS  
DENGAN IDENTIFIKASI ENTITAS**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--

## PRAKTIKUM 3: PENGOLAHAN DATA

---

Pertemuan ke	: 3
Total Alokasi Waktu	: 180 menit
• Pre-Test	: 30 menit
• Praktikum	: 120 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 50 %
• Post-Test	: 30 %

---

### 3.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu menerapkan:

1. *Insert* data
2. *Update* data
3. *Delete* data

Indikator ketercapaian diukur dengan:

1. Praktikan mampu mengenal *MySQL* dan *Database Management System*.
2. Praktikan mampu memahami tabel, tipe data dan primary key dalam *Database Management System*.
3. Praktikan mampu mengidentifikasi entitas pada suatu kasus yang disediakan.
4. Praktikan mampu mengolah data pada tabel yang meliputi menambah data, menghapus dan mengedit data.

### 3.2. TEORI PENDUKUNG

Basis Data adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu. Operasi dasar basis data :

- a. *Create database*
- b. *Drop database*
- c. *Create table*
- d. *Drop table*
- e. *Insert*
- f. *Retrieve / Search*
- g. *Update*
- h. *Delete*

### 3.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.

2. XAMPP.
3. Browser (mozilla firefox, chrome).

### 3.4. LANGKAH PRAKTIKUM

#### 1. Insert Data

Pada langkah ini digunakan untuk menginputkan data ke dalam tabel “mahasiswa”, sebagai contoh akan diinputkan sebanyak 3 data. Langkah menambahkan data adalah sebagai berikut:

- a. Pilih menu “Insert” sehingga muncul seperti pada gambar di bawah ini.

Column	Type	Function	Null	Value
nim	varchar(10)			
nama	varchar(50)			
prodi	varchar(50)			
alamat	text			

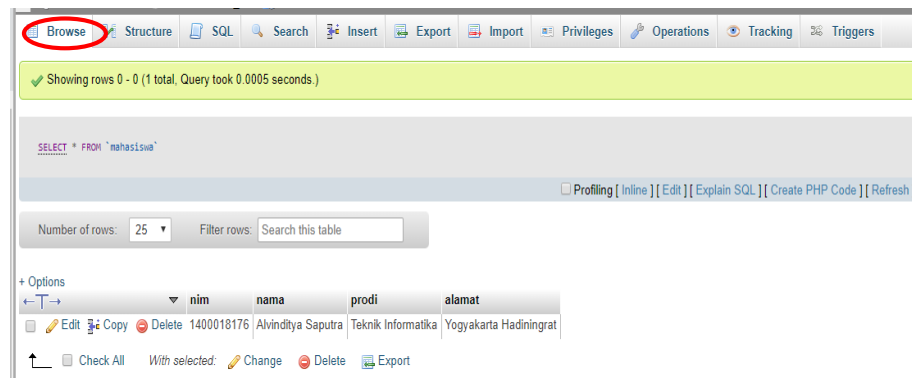
Gambar 3.1. Insert Data

- b. Kemudian isi field atau kolom isian pada bagian value kemudian pilih “Go”, seperti pada gambar di bawah ini.

Column	Type	Function	Null	Value
nim	varchar(10)			1400018176
nama	varchar(50)			Alvinditya Saputra
prodi	varchar(50)			Teknik Informatika
alamat	text			Yogyakarta Hadiningrat

Gambar 3.2. Isi Value Data

- c. Untuk memeriksa apakah data sudah masuk ke dalam *database*, dapat dilakukan dengan cara pilih pada menu “Browse”. Seperti pada gambar di bawah ini.



Gambar 3.3. Cek Data

Ulangi langkah menginputkan data hingga terdapat 3 data dalam tabel “mahasiswa”. Perlu diketahui bahwa atribut nim sebagai primary key, sehingga pada value yang diinputkan tidak boleh sama.

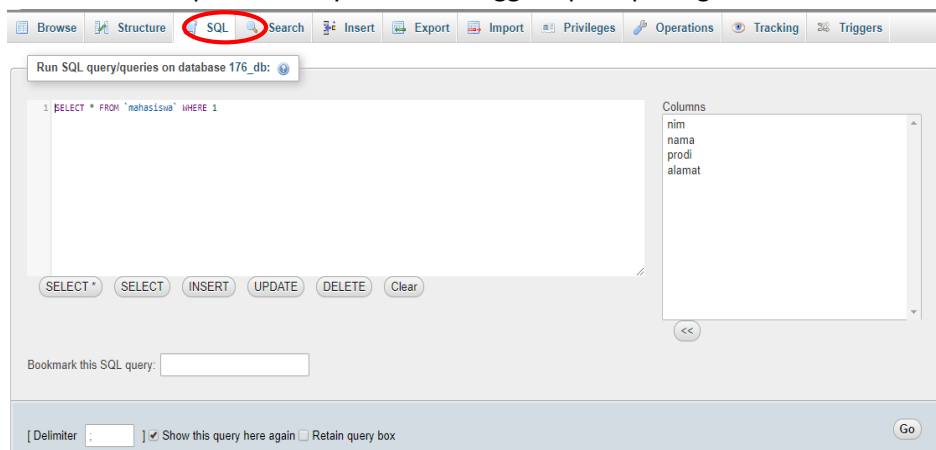
## 2. Read Data

Pada langkah ini digunakan untuk membaca atau mengambil data yang terdapat pada tabel mahasiswa. Terdapat beberapa query untuk membaca data sesuai yang ingin ditampilkan.

- SELECT \*** : Untuk membaca data dengan seluruh atribut yang ditampilkan. Query : **SELECT \* FROM nama\_tabel.**
- SELECT atribut\_1, atribut\_2 FROM nama\_tabel.** : Untuk membaca data dengan satu atau lebih atribut yang ditampilkan. Query : **SELECT atribut\_1, atribut\_2 FROM nama\_tabel.**
- SELECT ... WHERE** : Untuk membaca data dengan suatu kondisi yang diinginkan. Query : **SELECT \* FROM nama\_tabel WHERE id = 1.**

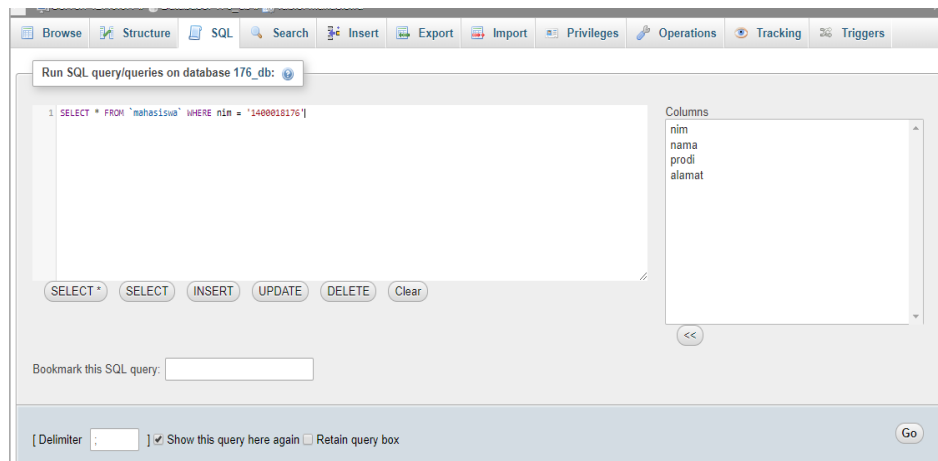
Langkah menampilkan data adalah sebagai berikut :

- Pilih menu SQL pada PHPMyAdmin sehingga seperti pada gambar dibawah ini.

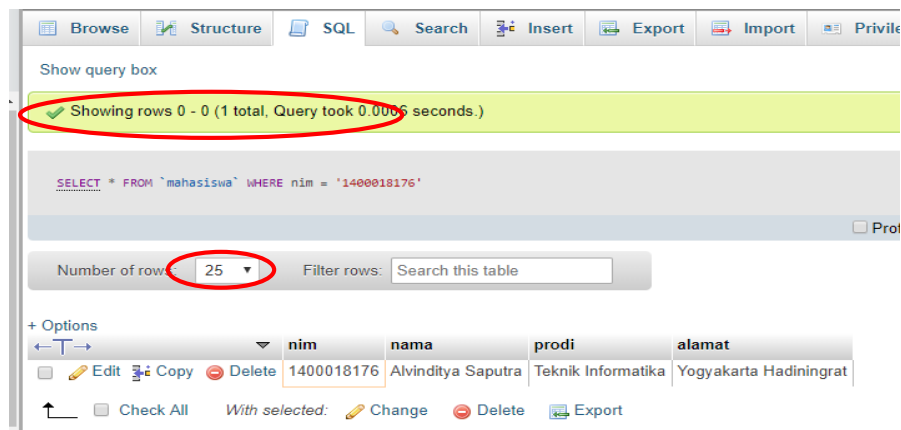


Gambar 3.4. Perintah SQL

- Misalkan data yang akan ditampilkan yaitu mahasiswa dengan nim 1400018176. Maka query yang dibutuhkan yaitu **SELECT \* FROM mahasiswa WHERE nim = '1400018176'**. Kemudian pilih “Go”, sehingga tampak seperti pada gambar di bawah ini.



Gambar 3.5. Tampil Data

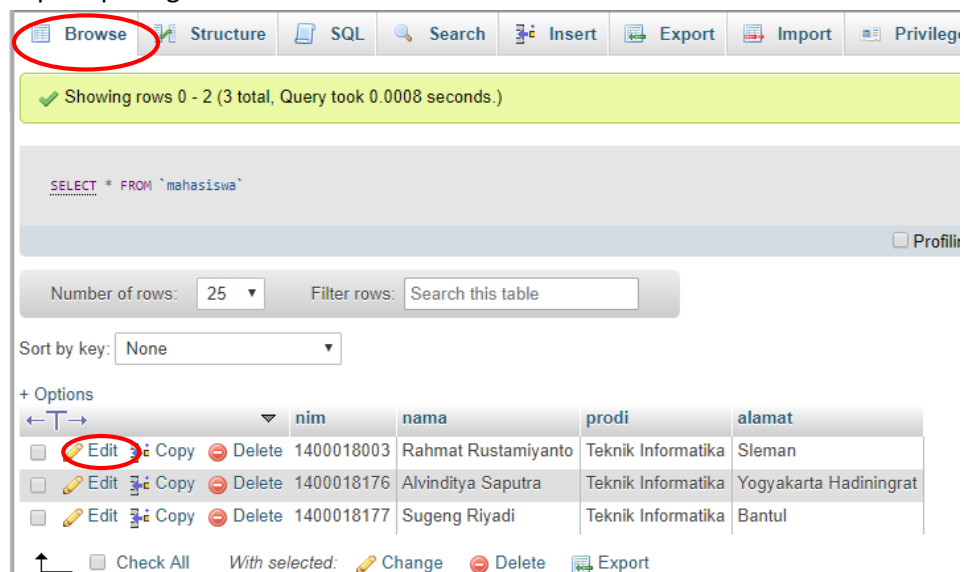


Gambar 3.6. Lihat Data

### 3. Update Data

Pada langkah ini digunakan untuk mengubah atau memodifikasi nilai atribut atau data dalam tabel atau entitas. Langkah mengubah data atau *update data* adalah sebagai berikut :

- Pilih menu “Browse” kemudian klik “edit” pada value atau field yang akan diubah datanya. Seperti pada gambar di bawah ini.



Gambar 3.7. Ubah Data

- b. Kemudian isi field yang akan diubah, misalkan data yang akan diubah yaitu pada field alamat. Kemudian klik “Go” sehingga seperti pada gambar dibawah ini.

Column	Type	Function	Null	Value
nim	varchar(10)			1400018003
nama	varchar(50)			Rahmat Rustamiyanto
prodi	varchar(50)			Teknik Informatika
alamat	text			Sleman Setelah diubah datanya

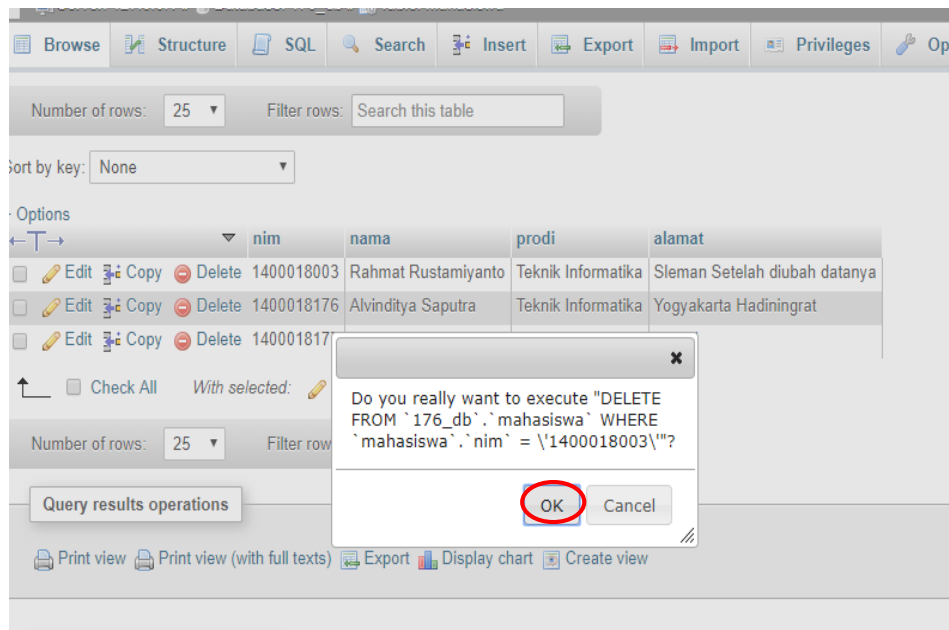
Gambar 3.8. Edit Data

#### 4. Delete Data

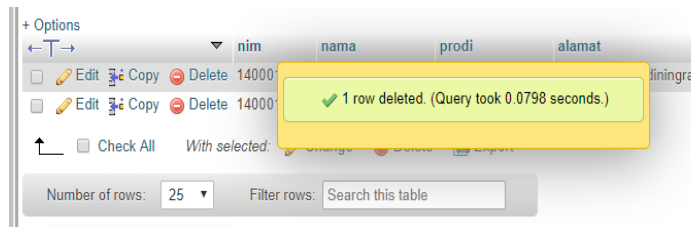
Pada langkah ini perintah *DELETE* digunakan untuk menghapus satu atau lebih data dalam sebuah tabel. Langkah menghapus data yaitu dengan klik “Delete” pada menu “Browse”. Seperti pada gambar dibawah ini.

	nim	nama	prodi	alamat
<input type="checkbox"/> Edit Copy Delete	1400018003	Rahmat Rustamiyanto	Teknik Informatika	Sleman Setelah diubah datanya
<input type="checkbox"/> Edit Copy Delete	1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat
<input type="checkbox"/> Edit Copy Delete	1400018177	Sugeng Riyadi	Teknik Informatika	Bantul

Gambar 3.9. Hapus Data



Gambar 3.10. Klik OK



Gambar 3.11. Data Berhasil Dihapus

### 3.5. TUGAS

Tugas diberikan oleh asisten praktikum.

**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 3: PENGELOLAAN DATA**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--



## PRAKTIKUM 4: KONVERSI LOGICAL MODEL KE RELATIONAL MODEL

---

Pertemuan ke : 4

Total Alokasi Waktu : 180 menit (Alokasi waktu disesuaikan dengan RPS)

- Pre-Test : 30 menit
- Praktikum : 120 menit
- Post-Test : 30 menit

Total Skor Penilaian : 100% (Bobot skor disesuaikan dengan RPS)

- Pre-Test : 20 %
  - Praktikum : 50 %
  - Responsi : 30 %
- 

### 1.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah melakukan praktek ini, mahasiswa dapat memahami, mempraktekkan cara melakukan konversi logical model ke relational model.

Indikator ketercapaian diukur dengan :

1. Mengukur pemahaman dengan memberikan asesmen berupa test tertulis dengan nama Pretest
2. Mengukur kemampuan praktek dengan cara memberikan asesmen berupa Pos Test, yaitu memberikan tugas terkait masalah dunia nyata untuk dibuatkan desain model logical ke relational model.

### 1.2. TEORI PENDUKUNG

#### a. Definisi Model Data

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 dalam memori sekunder. Hal ini akan berdampak pada bagaimana mengelompokkan data dan membentuk keseluruhan data yang terkait dalam sistem yang akan dibuat. ER Model (ERD) Yang merupakan representasi konseptual basis data harus dipetakan ke dalam relational model (relasi tabel) agar secara langsung dapat diimplementasikan ke basis data. Dalam relational model dikenal beberapa istilah yaitu:

- a. Relasi: Sebuah tabel yang terdiri dari beberapa kolom dan beberapa baris.
- b. Atribut: Kolom pada sebuah relasi (field).
- c. Domain: Kumpulan nilai yang valid untuk satu atau lebih atribut
- d. Tupel: Baris pada sebuah relasi (record).
- e. Derajat (degree): Jumlah atribut dalam sebuah relasi (jumlah field)
- f. Cardinality: Jumlah tupel dalam sebuah relasi (jumlah record)

Tabel 4.1. menjelaskan beberapa kosakata yang umum digunakan. Beberapa istilah yang sama memiliki pengertian berbeda tergantung pada model yang digunakan. Istilah dalam kolom paling kiri di bawah (ER Mode) bukan merupakan empat komponen dalam konstruksi ER model.

Tabel 4.1. .Kosa kata / istilah dalam beberapa ragam model

ER Model	Relational Model	Database	Traditional Programmer
Entity	Relation	Tabel	File
Entity Instan	Tuple	Row	Record
Atribut	Attribut	Kolom	Field
Identifier	Key	Key	Key/Link

**b. Algoritma Pemetakan model Relasi Entitas (ERD) Ke Relasi Tabel (Relational Model).**

Di dalam basis data yang menjadi pusat perhatian dan intisari sistem adalah tabel dan relasinya. Istilah tabel ini muncul dari abstraksi data pada level fisik. Tabel ini sama artinya dengan entitas dari model data pada level konseptual. Setiap orang bisa membuat tabel tetapi membuat tabel yang baik tidak semua orang dapat melakukannya. Kebutuhan akan membuat tabel yang baik ini melahirkan beberapa teori atau metode antara lain ialah pemetakan ER to table dan Normalisasi.

Uraian materi di bawah ini menjelaskan pemetakan ER model ke relasi tabel sedangkan Algoritma atau Langkah-langkah yang dilakukan untuk memetakan ER diagram ke tabel relasional yaitu sebagai berikut:

- 1) Untuk setiap entitas kuat, buat tabel baru Entitas Kuat yang menyertakan *seluruh simple atribut* dan *simple atribut* dari *composite atribut* yang ada. Pilih salah satu atribut kunci sebagai *primary key*.
- 2) Untuk setiap entitas lemah Entitas Lemah, buat tabel baru Entitas Lemah dengan mengikutsertakan seluruh *simple atribut*. Tambahkan *primary key* dari entitas kuatnya (owner entity type) yang akan digunakan sebagai *primary key* bersama-sama *partial key* dari entitas lemah.
- 3) Untuk setiap multivalued atribut R, buatlah tabel baru R yang menyertakan atribut dari multivalued tersebut. Tambahkan *primary key* dari relasi yang memiliki multivalued tersebut. Kedua atribut tersebut membentuk *primary key* dari tabel R.
- 4) Untuk setiap relasi binary 1:1, tambahkan *primary key* dari sisi yang lebih “ringan” ke sisi (entitas) yang lebih “berat”. Suatu sisi dianggap lebih “berat” timbangannya apabila mempunyai partisipasi total. Tambahkan juga *simple atribut* yang terdapat pada relasi tersebut ke sisi yang lebih “berat”. Apabila kedua partisipasi adalah sama-sama total atau sama-sama partial, maka dua entitas tersebut boleh digabung menjadi satu tabel.
- 5) Untuk setiap relasi binary 1:N yang tidak melibatkan entitas lemah, tentukan mana sisi yang lebih “berat” (sisi N). Tambahkan *primary key* dari sisi yang “ringan” ke tabel sisi

yang lebih “berat”. Tambahkan juga seluruh simple atribut yang terdapat pada relasi biner tersebut.

- 6) Untuk setiap relasi binary M:N, buatlah tabel baru R dengan atribut seluruh simple atribut yang terdapat pada relasi biner tersebut. Tambahkan primary key yang terdapat pada kedua sisi ke tabel R. Kedua foreign key yang didapat dari kedua sisi tersebut digabung menjadi satu membentuk primary key dari tabel R.
- 7) Untuk setiap relasi lebih dari dua entitas, n-nary (ternary), meliputi dua alternatif yaitu:
  - a. Buatlah tabel R yang menyertakan seluruh primary key dari entitas yang ikut serta. Sejumlah n foreign key tersebut akan membentuk primary key untuk tabel R. Tambahkan seluruh simple atribut yang terdapat pada relasi n-ary tersebut
  - b. Mengubah bentuk relasi ternary menjadi entitas lemah, kemudian memperbaiki relasi yang terjadi antara entitas lemah tersebut dengan entitas-entitas kuatnya dan melakukan algoritma pemetakan sesuai dengan aturan mapping.

### 1.3. ALAT DAN BAHAN

Dalam praktek Basis Data ini, alat dan bahan yang digunakan:

- a. Visio Drawing
- b. Komputer dengan periperalnya

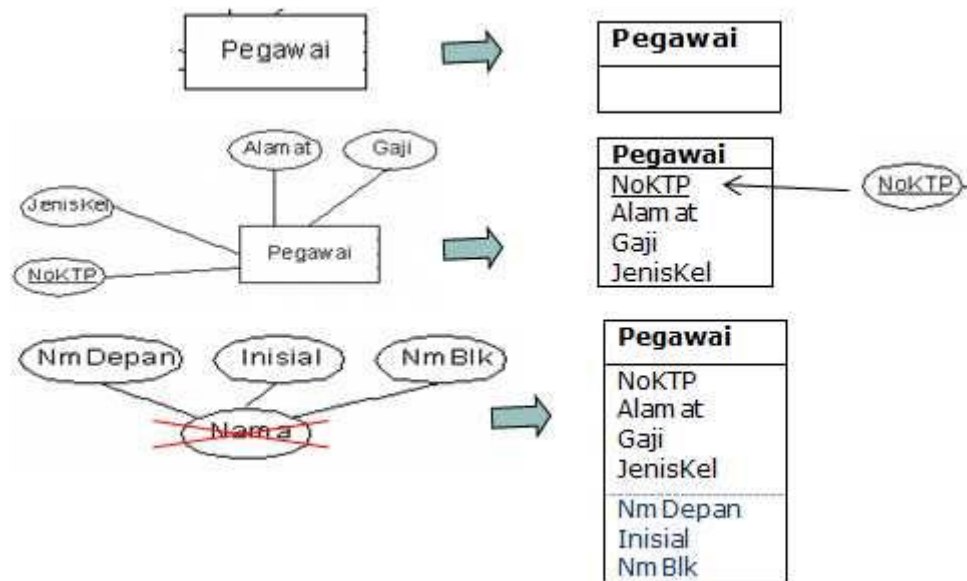
### 1.4. LANGKAH-LANGKAH PRAKTIKUM

- a. Mahasiswa menjalankan aplikasi Visio Drawing
- b. Mahasiswa membaca kasus sebagai berikut:  
System informasi perusahaan pada PT. XYZ. System dapat digunakan oleh seluruh karyawan dalam perusahaan tersebut. Proses yang terjadi dalam system berupa, penolahan data pegawai, pengolahan gaji, laporan gaji, pencarian data pegawai dan laporan dengan berbagai kategori.
- c. Membuat desain relasi model dengan menggunakan Visio Drawing
- d. Membuat relasi table dengan visio drawing

### 1.5. TUGAS PRAKTIKUM

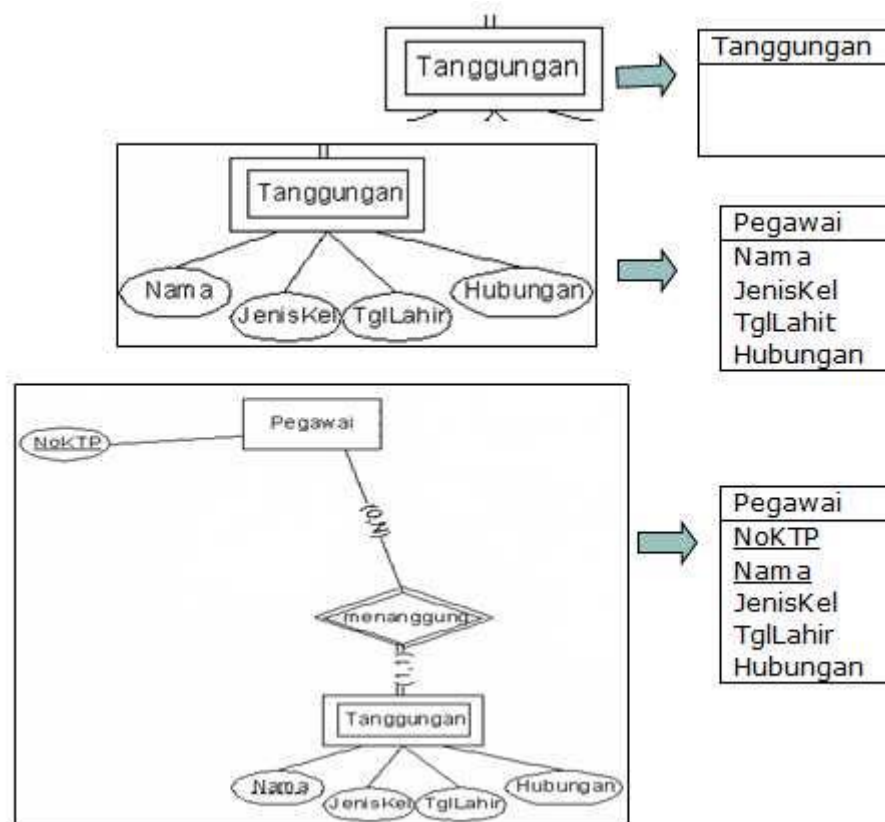
Mengubah model logika fisiknya kemudian diubah menjadi model data relational. Caranya sebagai berikut:

- a) Berdasarkan algoritma nomor 1 aturan tentang entitas kuat maka lakukan beberapa langkah dibawah ini:
  - 1) Untuk setiap entitas kuat Entitas Kuat, buat tabel baru Eks.
  - 2) Sertakan seluruh simple atribut.
  - 3) Sertakan simple atribut dari composite atribut yang ada.
  - 4) Pilih salah satu atribut kunci sebagai primary key.



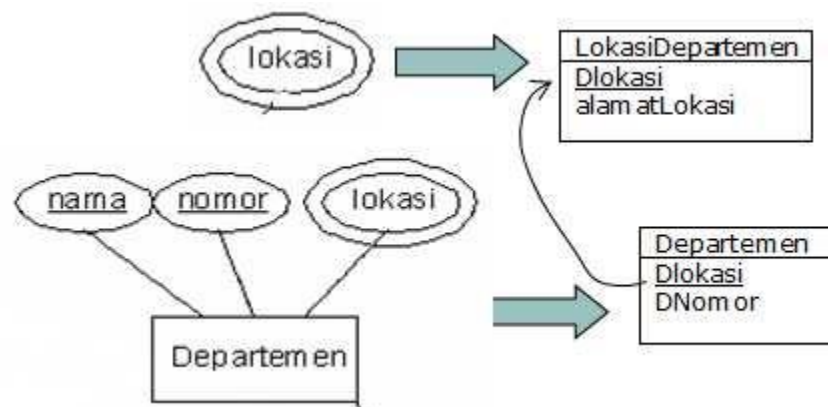
Gambar 4.1. Hasil Rancangan Entitas dari ERD

- b) Berdasarkan algoritma nomor 2 dari teori pendukung terkait aturan tentang entitas entitas lemah. Untuk setiap entitas lemah, lakukan beberapa langkah dibawah ini:
- 1) Buat tabel baru Entitas lemah.
  - 2) Sertakan seluruh simple atribut
  - 3) Tambahkan primary key dari entitas kuatnya (owner entity type) yang akan digunakan sebagai primary key bersama-sama partial key dari entitas lemah.



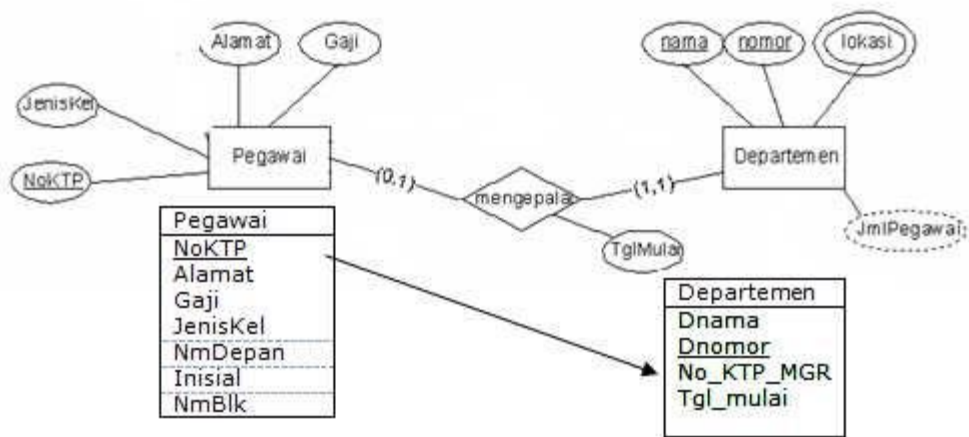
Gambar 4.2. Mapping ER ke tabel untuk entitas lemah

- c) Berdasarkan algoritma nomor 2 aturan tentang relasi multivalued atribut. Untuk setiap multivalued atribut R,
- 1) buatlah tabel baru R yang menyertakan atribut dari multivalued tersebut.
  - 2) Tambahkan primary key dari relasi yang memiliki multivalued tersebut.
- Kedua atribut tersebut membentuk primary key dari tabel R



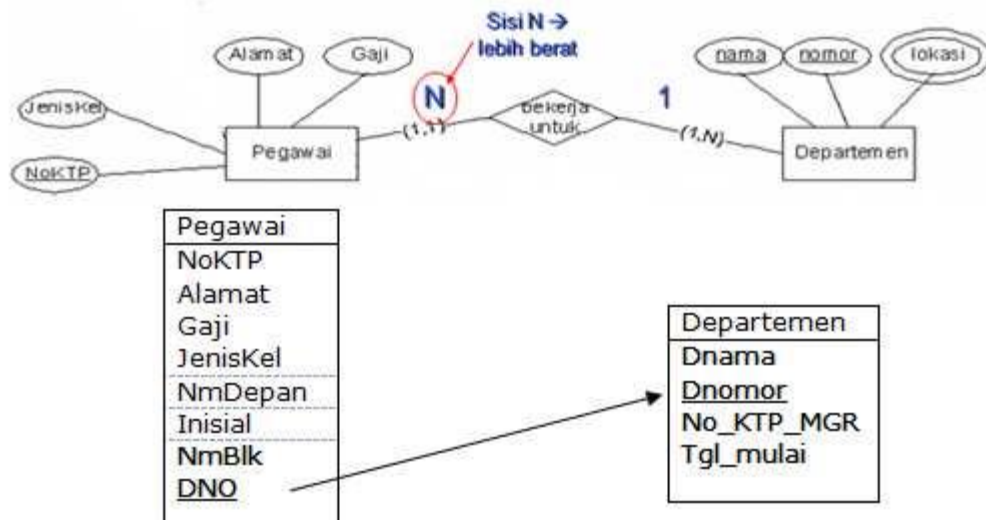
Gambar 4.3. Mapping multivalued attribute

- d) Untuk setiap relasi binary 1:1, tambahkan primary key dari sisi yang lebih “ringan” ke sisi (entitas) yang lebih “berat”. Suatu sisi dianggap lebih “berat” timbangannya apabila mempunyai partisipasi total. Tambahkan juga simple atribut yang terdapat pada relasi tersebut ke sisi yang lebih “berat”.



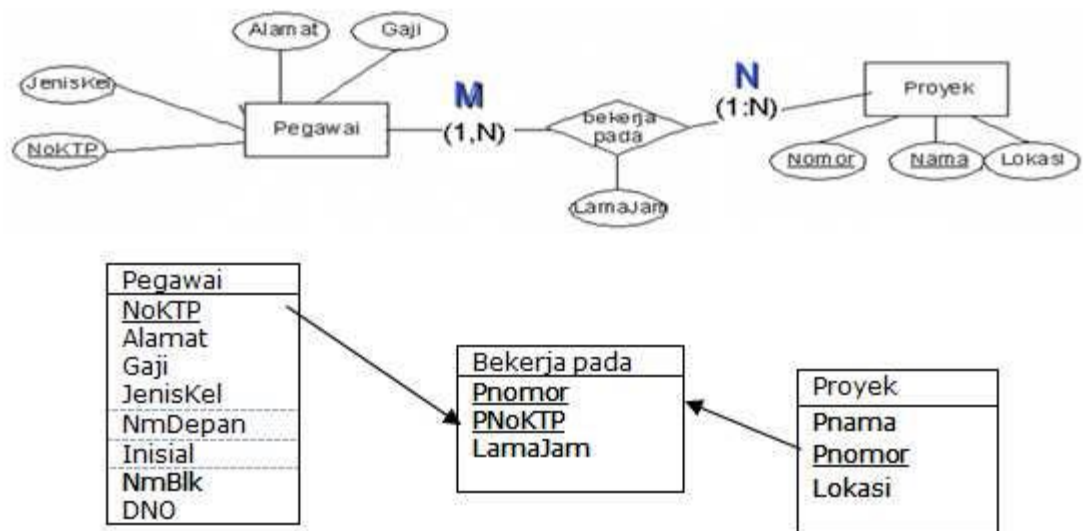
Gambar 4.4. Hasil Mapping Binary 1.1

- e) Untuk setiap relasi binary 1:N yang tidak melibatkan entitas lemah, tentukan mana sisi yang lebih “berat” (sisi N). Tambahkan primary key dari sisi yang “ringan” ke tabel sisi yang lebih “berat”. Tambahkan juga seluruh simple atribut yang terdapat pada relasi biner tersebut.



Gambar 4.5. Mapping ER to tabel relasi one to many

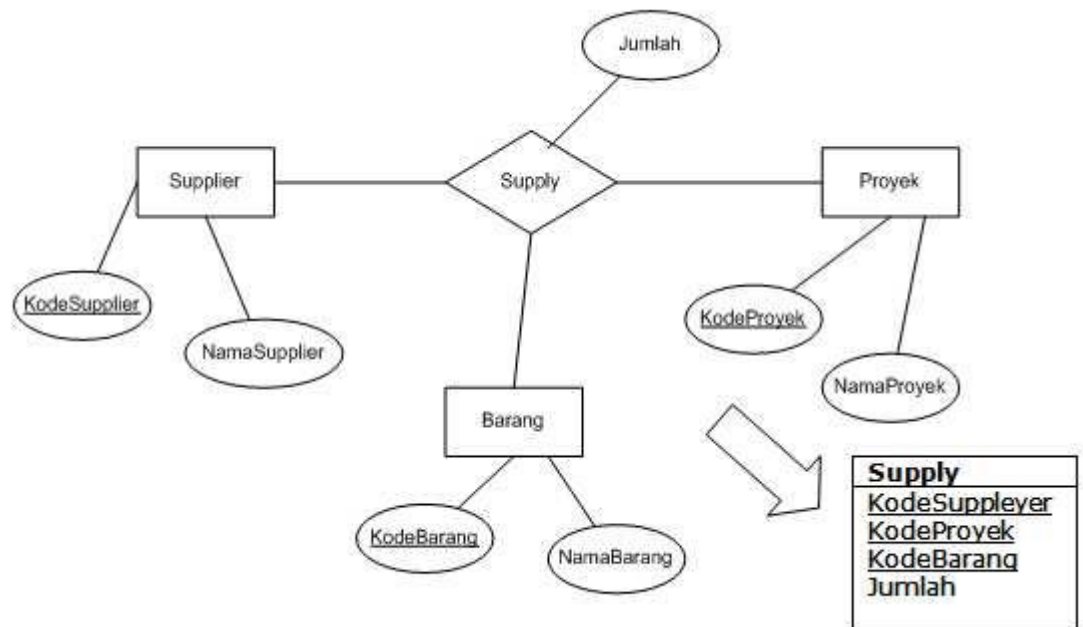
- f) Untuk setiap relasi binary M:N, buatlah tabel baru R dengan atribut seluruh simple atribut yang terdapat pada relasi biner tersebut. Tambahkan primary key yang terdapat pada kedua sisi ke tabel R. Kedua foreign key yang didapat dari kedua sisi tersebut digabung menjadi satu membentuk primary key dari tabel R.



Gambar 4.5. Mapping ER to tabel relasi one to many

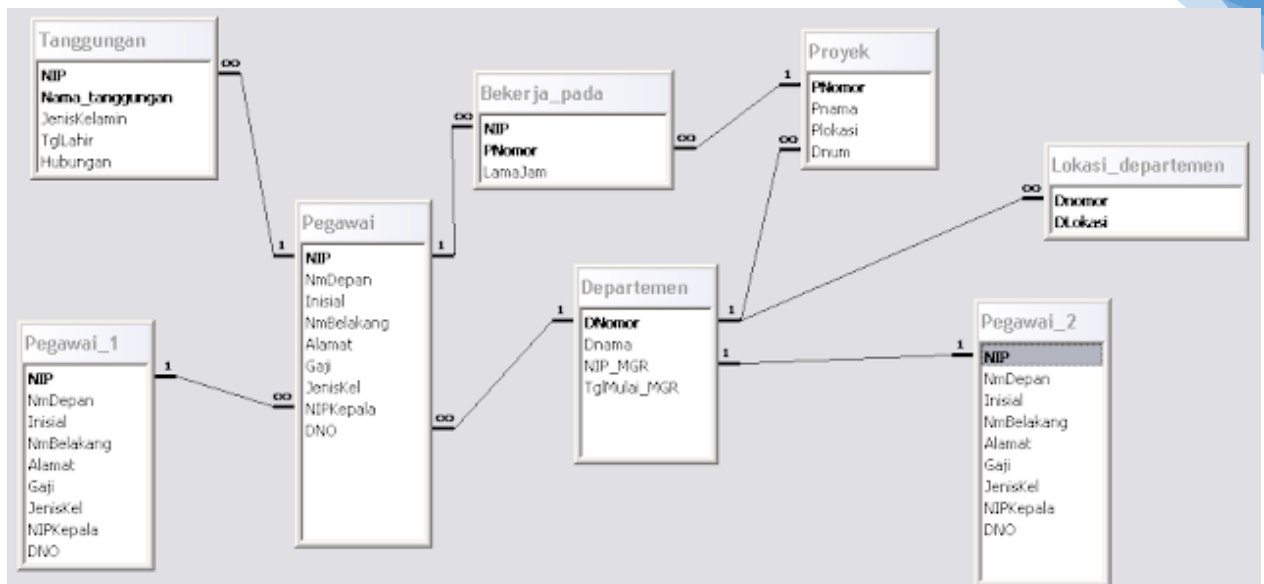
g) Untuk setiap relasi n-ary (ternary), caranya sebagai berikut:

1. Buatlah tabel R yang menyertakan seluruh primary key dari entitas yang ikut serta. Sejumlah n foreign key tersebut akan membentuk primary key untuk tabel R. Tambahkan seluruh simple atribut yang terdapat pada relasi n-ary tersebut.
2. Sama dengan proses yang dilakukan untuk langkah ke 6. Karena dalam ER-D perusahaan ini tidak ada relasi n-ary maka langkah ini tidak dilakukan.



Gambar 4.6. Mapping untuk relasi N-narry

h) Membuat hasil pemetaan ER diagram ke tabel untuk setiap relasi entitas dari ER diagram sistem basis data perusahaan XYZ dengan menggunakan Xampp.



Gambar 4.6. Relasi Tabel hasil Pemetakan ERD

- i) Cobalah membuat ER Model dan relasi ERD dengan kasus yang terdapat pada praktikum minggu sebelumnya.

### 3.6. TUGAS

Tugas diberikan oleh asisten praktikum.



**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 4: KONVERSI LOGICAL MODEL  
KE RELATIONAL MODEL**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--

## PRAKTIKUM 5: RELASI TABEL

Pertemuan ke	: 5
Total Alokasi Waktu	: 180 menit
• Pre-Test	: 30 menit
• Praktikum	: 120 menit
• Post-Test	: 30 menit

Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 50 %
• Post-Test	: 30 %

### 5.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu menerapkan:

1. *Relational*
2. *Entity*

Indikator ketercapaian diukur dengan:

1. Praktikan mampu memahami konsep dasar relasi antar entitas pada basis data.
2. Praktikan mampu membuat relasi satu atau lebih tabel dalam basis data.
3. Praktikan mampu mengintegrasikan *Foreign Key* dengan relasional.

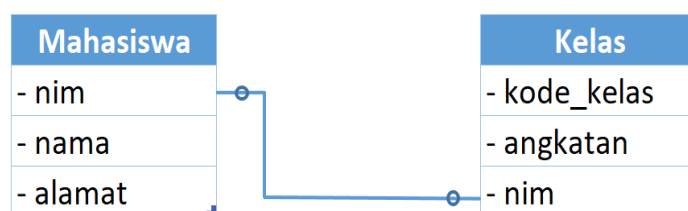
### 5.2. TEORI PENDUKUNG

#### 1. Relasi

Hubungan antara tabel yang mempresentasikan hubungan antar objek di dunia nyata. Relasi merupakan hubungan yang terjadi pada suatu tabel dengan lainnya yang mempresentasikan hubungan antar objek di dunia nyata dan berfungsi untuk mengatur mengatur operasi suatu database. Hubungan yang dapat dibentuk dapat mencakup 3 macam hubungan, yaitu:

##### a. One to One

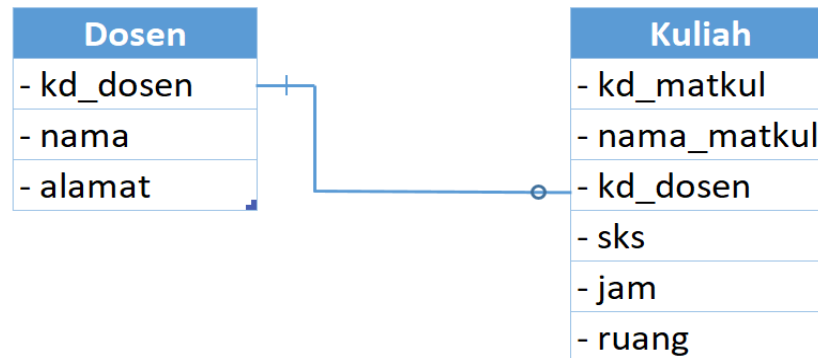
Mempunyai pengertian "Setiap baris data pada tabel pertama dihubungkan hanya ke satu baris data pada tabel ke dua". Contohnya : Mahasiswa menjadi ketua kelas di setiap kelasnya.



Gambar 5.1. Relasi 1

## b. One to Many

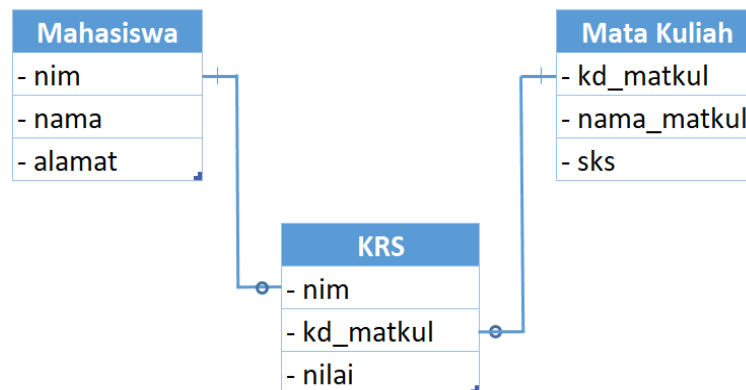
Mempunyai pengertian "Setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih data pada tabel ke dua". Contohnya : Dosen mengajar banyak mata kuliah.



Gambar 5.2. Relasi 2

## c. Many to Many

Mempunyai pengertian "Satu baris atau lebih data pada tabel pertama bisa dihubungkan ke satu atau lebih baris data pada tabel ke dua". Artinya ada banyak baris di tabel satu dan tabel dua yang saling berhubungan satu sama lain. Contohnya: Mahasiswa mengambil mata kuliah.



Gambar 5.3. Relasi 3

## 2. Primary dan Foreign Key

**Primary key** adalah field kunci / utama dari suatu tabel yang menunjukkan bahwa field yang menjadi kunci tersebut tidak bisa diisi dengan data yang sama, / dengan kata lain Primary key menjadikan tiap record **memiliki identitas sendiri-sendiri yang membedakan satu sama lainnya (unik)**. Sedangkan **foreign key** yaitu satu atribut (atau satu set atribut) yang **melengkapi satu relationship (hubungan) yang menunjukkan ke induknya**. *Foreign key* berguna untuk mendefinisikan kolom-kolom pada suatu tabel yang nilainya mengacu ke tabel lain, jadi kolom foreign key nilainya harus diambil dari nilai kolom pada tabel lain.

### 5.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. XAMPP.
3. *Browser (firefox, chrome).*

### 5.4. LANGKAH PRAKTIKUM

Langkah praktikum berisi tahapan secara rinci bagaimana praktikum dijalankan dan apa hasil yang harus dicapai dari setiap langkah.

#### 1. Tabel Utama

Membuat tabel “Dosen” dengan atribut `kode_dsn`, `nama_dsn`, dan `alamat_dsn` serta membuat tabel “Mata\_Kuliah” dengan atribut `kode_mk`, `nama_mk`, `sks`. Kemudian isi data pada masing-masing tabel sebanyak 3 data seperti pada pertemuan sebelumnya.

Tabel Dosen :

				kode_dsn	nama_dsn	alamat_dsn
<input type="checkbox"/>	Edit	Copy	Delete	50001	Slamet Widodo	Yogyakarta
<input type="checkbox"/>	Edit	Copy	Delete	50002	Wahyu Subrata	Bandung
<input type="checkbox"/>	Edit	Copy	Delete	50003	Sri Linggajati	Lampung

Gambar 5.4. Data Dosen

Tabel Mata\_Kuliah :

				kode_mk	nama_mk	sks
<input type="checkbox"/>	Edit	Copy	Delete	T1001	Basis Data	4
<input type="checkbox"/>	Edit	Copy	Delete	T1002	Pemrograman Web	3
<input type="checkbox"/>	Edit	Copy	Delete	T1003	Statistika	4

Gambar 5.5. Data Mata Kuliah

#### 2. Tabel Relasi *One to One*

Membuat tabel “Kelas” untuk mengimplementasikan jenis relasi One to One, seperti pada gambar di bawah ini.

Table name:  Add  column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
<input type="text" value="kode_kelas"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="10"/>	<input type="text" value="None"/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="checkbox"/>	<input type="text" value="PRIMARY"/>
<input type="text" value="angkatan"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="50"/>	<input type="text" value="None"/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="checkbox"/>	<input type="text" value=""/>
<input type="text" value="nim_ketua"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="10"/>	<input type="text" value="None"/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="checkbox"/>	<input type="text" value=""/>
<input type="text" value=""/>	<input type="text" value="INT"/>	<input type="text" value=""/>	<input type="text" value="None"/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="checkbox"/>	<input type="text" value=""/>

Gambar 5.6. Isi Data Tabel

Kemudian menginputkan data dengan catatan nim harus sudah ada pada tabel mahasiswa, karena tabel ini akan membaca informasi data dari tabel mahasiswa. Seperti pada contoh di bawah ini.

+ Options

	kode_kelas	angkatan	nim_ketua
<input type="checkbox"/> Edit  Copy  Delete	14_C	2014	1400018176

☐ Check All    With selected: Change Delete Export

Gambar 5.7. Tampil Data

## 5.5. TUGAS

Tugas diberikan oleh asisten praktikum.

Studi Kasus basis data di perpustakaan, buatlah salah satu contoh di bawah ini

1. Contoh Tabel relasi one to one
2. Contoh tabel relasi one to many
3. Contoh tabel relasi many to many

**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 5: RELASI TABEL**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--

## PRAKTIKUM 6: NORMALISASI & BUSSINESS RULES

Pertemuan ke : 6

Total Alokasi Waktu : 180 menit

- Pre-Test : 30 menit
- Praktikum : 120 menit
- Post-Test : 30 menit

Total Skor Penilaian : 100% (Bobot skor disesuaikan dengan RPS)

- Pre-Test : 20 %
- Praktikum : 50 %
- Post-Test : 30 %

### 6.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah melakukan praktek ini, mahasiswa mampu memahami dan menjelaskan konsep normalisasi dan *bussines rules*.

Indikator ketercapaian diukur dengan :

1. Mengukur pemahaman dengan memberikan asesmen berupa test tertulis dengan nama Pretest
2. Mengukur kemampuan praktek dengan cara memberikan asesmen berupa Pos Test, yaitu memberikan tugas terkait masalah dunia nyata untuk dibuatkan desain model Normalisasi dan model bisnisnya.

### 6.2. TEORI PENDUKUNG

#### a. Definisi Normalisasi

**Normalisasi database** merupakan suatu pendekatan sistematis untuk meminimalkan redundansi data pada suatu *database* agar *database* tersebut dapat bekerja dengan optimal. Jika anda seorang *database administrator* ketika terjadi sesuatu pada database seperti penurunan kinerja, mungkin anda akan ditanya apakah database tersebut telah di normalisasi?

#### b. Tujuan Normalisasi Database

Tujuan normalisasi *database* adalah untuk menghilangkan dan mengurangi redudansi data dan tujuan yang kedua adalah memastikan dependensi data (Data berada pada tabel yang tepat). Jika data dalam database tersebut belum di normalisasi maka akan terjadi 3 kemungkinan yang akan merugikan sistem secara keseluruhan.

1. INSERT Anomali : Situasi dimana tidak memungkinkan memasukkan beberapa jenis data secara langsung di database.
2. DELETE Anomali: Penghapusan data yang tidak sesuai dengan yang diharapkan, artinya data yang harusnya tidak terhapus mungkin ikut terhapus.
3. UPDATE Anomali: Situasi dimana nilai yang diubah menyebabkan inkonsistensi database, dalam artian data yang diubah tidak sesuai dengan yang diperintahkan atau yang diinginkan.

#### c. Normalisasi dalam Database

**Normalisasi database** terdiri dari banyak bentuk, dalam ilmu basis data ada setidaknya 9 bentuk normalisasi yang ada yaitu 1NF, 2NF, 3NF, EKNF, BCNF, 4NF, 5NF, DKNF, dan 6NF. Namun dalam prakteknya dalam dunia industri bentuk normalisasi ini yang paling sering digunakan ada sekitar 5 bentuk.

**d. Tahapan dalam Normalisasi**

Tahap-tahap normalisasi tersebut adalah:

1) Bentuk Normal ke Satu (1NF)

Syarat:

- a) Tidak ada set atribut yang berulang atau bernilai ganda, setiap atribut yang dimilikinya bersifat *atomic* (bernilai tunggal) untuk setiap baris.
- b) Telah ditentukannya *primary key* untuk tabel atau relasi.
- c) Tiap atribut hanya memiliki satu pengertian.
- d) Tiap atribut yang dapat memiliki banyak nilai sebenarnya menggambarkan entitas atau relasi yang terpisah.

2) Bentuk Normal ke Dua (2NF)

Syarat:

- a) Bentuk data telah memenuhi kriteria bentork normal ke satu.
- b) Atribut bukan kunci (*non - key atribut*) haruslah memiliki ketergantungan fungsional sepenuhnya pada *primary key*.
- c) Kunci primer hanya mengandung satu atribut.

3) Bentuk Normal ke Tiga (3NF)

Syarat:

- a) Bentuk data telah memenuhi kriteria ke dua.
- b) Tidak boleh terdapat ketergantungan transitif terhadap kunci utama atau *primary key*.

4) *Boyce-Codd Normal Form* (BCNF)

Syarat:

Semua anomali (kesalahan data) yang tersisa dari hasil penyempurnaan kebergantungan fungsional telah dihilangkan.

5) Bentuk Normal ke Empat (4NF)

Syarat:

- a) Bila dan hanya bila telah berada dalam bentuk BCNF dan tidak ada *multivalued dependency nontrivial*.
- b) *Multivalued Dependency Nontrivial (MVD)* dipakai dalam 4NF.
- c) *Dependency* ini dipakai untuk menyatakan hubungan satu (*one to many*).

6) Bentuk Normal ke Lima (5NF)

Syarat:

Semua anomali (kesalahan data) yang tertinggal telah dihilangkan.



Dari beberapa tahap normalisasi diatas, Bentuk Normal Pertama (1NF) sampai Normal ke Tiga (3NF), merupakan bentuk normal yang umum dipakai. Umumnya bila ketiga bentuk normal tersebut telah dipenuhi, maka persoalan anomali tidak akan muncul.

### 6.3. ALAT DAN BAHAN

Dalam pelaksanaan praktikum pada materi Normalisasi, menggunakan peralatan sebagai berikut:

- a. Software Visio Drawing
- b. Komputer dengan periperalnya

### 6.4. LANGKAH PRAKTIKUM

1. Bacalah kasus terkait entitas mahasiswa yang belum ternormalisasi pada Tabel 6.1. untuk proses Normalisasi dengan MySql menggunakan perintah: insert, hapus dan update.

Tabel 6.1. Entitas Mahasiswa UnNormal

NIM	NAMA	Sem	MaKul	NIDN	Nama Dosen
201001	Andika Saputra	1	Algoritma	1078523	Riyanti Anjani
			Struktur Data	1078523	Riyanti Anjani
201002	Biyanti Anggie	3	Struktur Data	1078523	Riyanti Anjani
			Orkom	1078523	Riyanti Anjani
			Metnum	1075047	Susan Savitri
201003	Naura Putri	5	Web	1075047	Susan Savitri
			Jarkom	1077021	Erwin Masadi
			Metnum	1075047	Susan Savitri

2. Dari Table 6.1 lakukanlah bentuk normalisasi tahap 1 (1 NF). Caranya dengan menggunakan fungsi hapus, insert dan update data dari table 6.1 dengan menggunakan fungsi dalam Mysql hasilnya dapat dilihat pada Tabel 6.2.

Tabel 6.2. Bentuk Normal 1 pada Entitas Mahasiswa

NIM	NAMA	Sem	MaKul	NIDN	Nama Dosen
201001	Andika Saputra	1	Algoritma	1078523	Riyanti Anjani
201001	Andika Saputra	1	Struktur Data	1078523	Riyanti Anjani
201002	Biyanti Anggie	3	Struktur Data	1078523	Riyanti Anjani
201002	Biyanti Anggie	3	Orkom	1078523	Riyanti Anjani
201002	Biyanti Anggie	3	Metnum	1075047	Susan Savitri
201003	Naura Putri	5	Web	1075047	Susan Savitri
201003	Naura Putri	5	Jarkom	1077021	Erwin Masadi
201003	Naura Putri	5	Metnum	1075047	Susan Savitri

3. Dari Table 6.2 dilakukan normalisasi lagi karena adanya kemunculan data dengan primary yang sama berulang. Caranya dengan memecah table 6.2 menjadi Tabel dosen dan table mahasiswa (sebagai entitas baru). Untuk itu kita perlu membuat table baru. Dalam MySql fungsi membuat table dengan perintah Create. Hasilnya dapat dilihat pada Table 6.3, Tabel 6.4., Tabel 6.5. dan Tabel 6.6.

Tabel 6.3. Tabel Dosen belum Ternormalisasi

NIDN	NAMA DOSEN	MaKul_1	MaKul_2	MaKul_3
1078523	Riyanti Anjani	Algoritma	Struktur Data	Orkom
1075047	Susan Savitri	Metnum	Web	-
1077021	Erwin Masadi	Jarkom	-	-

Tabel 6.4. Tabel Dosen Ternormalisasi bentuk 1

NIDN	NAMA DOSEN	MaKul
1078523	Riyanti Anjani	Algoritma
1078523	Riyanti Anjani	Struktur Data
1078523	Riyanti Anjani	Orkom
1075023	Susan Savitri	Metnum
1075023	Susan Savitri	Web
1077021	Erwin Masadi	Jarkom

Tabel 6.5. Tabel Mahasiswa Normalisasi bentuk 1

NIM	NAMA	Sem	MaKul	NIDN	Nama Dosen
201001	Andika Saputra	1	Algoritma	1078523	Riyanti Anjani
201001	Andika Saputra	1	Struktur Data	1078523	Riyanti Anjani
201002	Biyanti Anggie	3	Struktur Data	1078523	Riyanti Anjani
201002	Biyanti Anggie	3	Orkom	1078523	Riyanti Anjani
201002	Biyanti Anggie	3	Metnum	1075047	Susan Savitri
201003	Naura Putri	5	Web	1075047	Susan Savitri
201003	Naura Putri	5	Jarkom	1077021	Erwin Masadi
201003	Naura Putri	5	Metnum	1075047	Susan Savitri

4. Merubah ke bentuk Normalisasi 2.

Untuk merubah ke bentuk Normalisasi bentuk 2, dapat dilakukan dengan cara: menentukan Atribut Kunci atau Primary Key.

Untuk Tabel 6.5. Entitas Mahasiswa dapat diubah menjadi Normalisasi bentuk ke 2, caranya:

- a) Tentukan Primary Key (PK) : NIM dengan atributnya sebagai berikut:
- Atribut yang bergantung dengan PK : Nama, Semester
  - Atribut yang tidak bergantung dengan PK: Makul, NIDN, NamaDosen
- b) Untuk memenuhi 2NF, atribut yang tidak bergantung dengan primary key dipecah menjadi entitas baru, sehingga entitas mahasiswa dipecah menjadi 2 entitas, yaitu; **Entitas Mahasiswa 2NF** (Nim, Nama, Semester) dan **Entitas Ambil\_MK** (Makul, Nidn, NamaDosen).

Tabel 6.6. Entitas Mahasiswa bentuk 2 belum Ternormalisasi

NIM	NAMA	Sem
201001	Andika Saputra	1
201001	Andika Saputra	1
201002	Biyanti Anggie	3
201002	Biyanti Anggie	3
201002	Biyanti Anggie	3
201003	Naura Putri	5
201003	Naura Putri	5
201003	Naura Putri	5

Terjadi kerangkapan data yang tidak diperlukan (redudansi), untuk itu, data yang sama bisa dihilangkan.

Tabel 6.7. Entias Mahasiswa Normalisasi bentuk 2

NIM	NAMA	Sem
2010001	Andika Saputra	1
2010002	Biyanti Anggie	3
2010003	Naura Putri	5

Untuk entitas Ambil MK hasil Normalisasi Bentuk 2 disajikan pada Tabel 6.8.

Tabel 6.8. Tabel Entitas Ambik MK Normalisasi 2

NIM	MaKul	NIDN	Nama Dosen
201001	Algoritma	1078523	Riyanti Anjani
201001	Struktur Data	1078523	Riyanti Anjani
201002	Struktur Data	1078523	Riyanti Anjani
201002	Orkom	1078523	Riyanti Anjani
201002	Metnum	1075047	Susan Savitri
201003	Web	1075047	Susan Savitri
201003	Jarkom	1077021	Erwin Masadi

### 5. Mengubah menjadi Tabel Normalisasi bentuk 3

Agar Tabel 7.8 Entitas ambil Kuliah Normal 2 dapat diubah menjadi Normalisasi bentuk 3, maka dilakukan cara sebagai berikut:

1. Menentukan Primary Key (PK) : NIDN
2. Menentukan atribut yang bergantung dengan PK : Nama Dosen
3. Menentukan Atribut yang bergantung transitif dengan PK: NIM, Makul
4. Untuk memenuhi 2NF, Entitas **Ambil\_MK** 2NF dipecah menjadi 2 Entitas, yaitu; Entitas Dosen 3NF (NIDN, Nama Dosen) dan Entitas Ambil\_MK 3NF (Nim, Makul).

Tabel 6.9. Tabel Entitas Dosen Bentuk Normal 3

NIDN	Nama Dosen	Menjadi	NIDN	Nama Dosen
1078523	Riyanti Anjani		1078523	Riyanti Anjani
1078523	Riyanti Anjani		1075047	Susan Savitri
1078523	Riyanti Anjani		1077021	Erwin Masadi
1078523	Riyanti Anjani			
1075047	Susan Savitri			
1075047	Susan Savitri			
1077021	Erwin Masadi			

Tabel 6.10 Tabel Ambil MK Bentuk Normal 3

NIM	Makul	menjadi	NIM	Makul	NIDN
201001	Algoritma		201001	Algoritma	1078523
201001	Struktur Data		201001	Struktur Data	1078523
201002	Struktur Data		201002	Struktur Data	1078523
201002	Orkom		201002	Orkom	1078523
201002	Metnum		201002	Metnum	1075047
201003	Web		201003	Web	1075047
201003	Jarkom		201003	Jarkom	1077021
201003	Metnum		201003	Metnum	1078523

6. Lakukanlah dengan membuat Query untuk membuat table normalisasi 1 sampai Normalisasi 3 dengan menggunakan perintah-perintah Insert, Delet dan Update.

## 6.5. TUGAS

Perhatikan Tabel 6.11. ini merupakan table bentuk Tidak Normal dari suatu fatur penjualan.

Tabel 6.11. Bentuk Tidak Normal dari Tabel Penjualan

No Fatur	Tanggal	Kd-Pelanggan	Nama	Kode Barang	Nama Barang	Harga	QTY
F-01	12/10/20	P-001	Tommy	K-001	Mie	10.000	5
				K-002	Gula	20.000	2
F-02	15/10/20	P-002	Susi	K-001	Mie	10.000	3
				K-003	Garam	8.000	2
F-03	16/10/20	P-003	Yanti	K-004	Tepung	15.000	1
				K-002	Gula	20.000	1

Berdasarkan table 6.11 tersebut, lakukanlah normalisasi dengan membuaat table menggunakan fungsi Create, dan melakukan normalisasi menggunakan perintah insert, delet dan update dengan MySql.

**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 6: NORMALISASI & BUSSINESS RULES**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--

## PRAKTIKUM 7: DDL dan DML

---

Pertemuan ke : 7

Total Alokasi Waktu : 180 menit (Alokasi waktu disesuaikan dengan RPS)

- Pre-Test : 30 menit
- Praktikum : 120 menit
- Post-Test : 30 menit
- dst

Total Skor Penilaian : 100% (Bobot skor disesuaikan dengan RPS)

- Pre-Test : 20 %
  - Praktikum : 50 %
  - Post-Test : 30 %
  - dst
- 

### 7.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu menerapkan:

1. *Create*
2. *Alter*
3. *Drop*
4. *Insert*
5. *Update*
6. *Delete*
7. *Select*

Indikator ketercapaian diukur dengan praktikan mampu memahami perintah DDL dan DML.

### 7.2. TEORI PENDUKUNG

Data Definition Language memiliki fungsi untuk melakukan hal-hal berikut :

1. Membuat/menghapus database, dinyatakan dengan perintah CREATE DATABASE dan DROP DATABASE
2. Membuat/menghapus table, dinyatakan dengan perintah CREATE TABLE dan DROP TABLE
3. Memodifikasi table, dinyatakan dengan perintah ALTER TABLE

Sedangkan Data Manipulation Language terdiri atas :

1. Mengisi tabel dengan data, dinyatakan dengan perintah INSERT
2. Mengedit data pada tabel, dinyatakan dengan perintah UPDATE
3. Menghapus data pada tabel, dinyatakan dengan perintah DELETE
4. Mencari data pada tabel, dinyatakan dengan perintah SELECT

### 7.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.

2. XAMPP.
3. Browser (firefox, chrome).

#### 7.4. LANGKAH PRAKTIKUM

Langkah praktikum berisi tahapan secara rinci bagaimana praktikum dijalankan dan apa hasil yang harus dicapai dari setiap langkah.

##### 1) Perintah DDL

1. Menampilkan database yang ada pada mysql  
SHOW databases;
2. Membuat database  
CREATE database <nama\_database>;  
CREATE database akademik;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| test      |
+-----+
3 rows in set (0.02 sec)

mysql> create database akademik;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| akademik      |
| mysql      |
| test      |
+-----+
4 rows in set (0.00 sec)
```

Gambar 7.1. Database MySQL

3. Menggunakan database  
USE <nama\_database>;  
USE akademik;
4. Menghapus database  
DROP <nama\_database>;  
DROP akademik;

```
+-----+
| Database |
+-----+
| information_schema |
| akademik      |
| mysql      |
| test      |
+-----+
4 rows in set (0.00 sec)

mysql> use akademik;
Database changed
mysql> drop database akademik;
Query OK, 0 rows affected (0.02 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| test      |
+-----+
3 rows in set (0.00 sec)
```

Gambar 7.2. Show Database



5. Melihat tabel apa yang sudah ada dalam database yang aktif

SHOW tables;

```
mysql> use akademik;
Database changed
mysql> show tables;
Empty set (0.00 sec)
```

Gambar 7.3. Show Tables

6. Membuat tabel

CREATE TABLE <nama\_tabel> (<nama\_kolom><tipe data>);

CREATE TABLE mhs (NIM CHAR (8) NOT NULL PRIMARY KEY, nama\_mhs VARCHAR (30), alamat VARCHAR (35));

PRIMARY KEY adalah kunci utama, dalam setiap tabel harus ada minimal satu kolom yang dijadikan sebagai PRIMARY KEY.

NOT NULL artinya setiap kolom tidak boleh kosong, sedangkan jika diijinkan untuk dikosongkan dapat menggunakan parameter NULL.

7. Melihat struktur tabel

DESCRIBE <nama\_tabel>;

DESC <nama\_tabel>;

DESCRIBE mhs;

DESC mhs;

```
mysql> create table mhs (nim char(8) not null primary key, nama_mhs varchar(30),
alamat varchar(35));
Query OK, 0 rows affected (0.08 sec)

mysql> desc;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near '' at
line 1
mysql> desc mh;
ERROR 1146 (42S02): Table 'akademik.mh' doesn't exist
mysql> desc mhs;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim   | char(8) | NO | PRI | NULL | |
| nama_mhs | varchar(30) | YES | | NULL | |
| alamat | varchar(35) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _
```

Gambar 7.4. Desc Tabel

8. Mengubah struktur tabel

Ada saatnya kita sadar kalo ternyata struktur tabel yang pernah dibuat perlu penyempurnaan, bisa dalam hal penambahan kolom, pengubahan lebar kolom, penghapusan kolom, dan sebagainya, sehingga dengan penggunaan perintah ALTER ini maka kita dapat mengubah kekurangan atau kesalahan pada saat kita membuat tabel.

ALTER TABLE nama\_tabel [ spesifikasi perubahan ]

Parameter [ spesifikasi perubahan ] adalah pilihan yang digunakan untuk mengubah struktur tabel yaitu CHANGE, ADD, DROP.

ALTER TABLE nama\_tabel jenis\_pengubahan

- a. Menambah kolom baru

Parameter yang digunakan adalah ADD.

ALTER TABLE nama\_tabel ADD kolom\_baru tipe(panjang) [FIRST | AFTER kolom\_lama]  
 FIRST artinya kolom yang baru akan ditambahkan pada urutan pertama. AFTER artinya kita meletakkan kolom yang baru setelah kolom yang ditunjuk.

Misalnya kita ingin menambah kolom baru yaitu no\_telp pada tabel mhs dengan posisi setelah kolom/field alamat dengan tipe INT :

ALTER TABLE mhs ADD no\_telp INT AFTER alamat;

```
mysql> alter table mhs add no_telp int after alamat;
Query OK, 0 rows affected (0.19 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc mhs;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim    | char(8) | NO   | PRI | NULL    |       |
| nama_mhs | varchar(30) | YES |     | NULL    |       |
| alamat | varchar(35) | YES |     | NULL    |       |
| no_telp | int(11) | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

Gambar 7.5. Tambah Atribut

b. Mengubah nama kolom

Parameter yang digunakan adalah CHANGE.

ALTER TABEL nama\_tabel CHANGE kolom\_lama kolom\_baru tipe (panjang)

ALTER TABLE mhs CHANGE alamat alamat\_mhs VARCHAR (35)

```
mysql> alter table mhs change alamat alamat_mhs varchar(35);
Query OK, 0 rows affected (0.25 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc mhs;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim    | char(8) | NO   | PRI | NULL    |       |
| nama_mhs | varchar(30) | YES |     | NULL    |       |
| alamat_mhs | varchar(35) | YES |     | NULL    |       |
| no_telp | int(11) | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Gambar 7.6. Ubah Nama Atribut

c. Mengubah nama table

Parameter yang digunakan adalah RENAME.

ALTER TABLE nama\_lama RENAME [TO] nama\_baru;

ALTER TABLE mhs RENAME TO mahasiswa;

```
mysql> alter table mhs rename to mahasiswa;
Query OK, 0 rows affected (0.06 sec)

mysql> show tables;
+-----+
| Tables_in_akademik |
+-----+
| mahasiswa           |
+-----+
1 row in set (0.00 sec)
```

Gambar 7.7. Ubah Nama Tabel

d. Menghapus atau menghilangkan komponen pada tabel

Menghapus ini dapat mencakup menghilangkan primary key, kolom, tabel, index pada kolom.

Untuk menghilangkan PRIMARY KEY

ALTER TABLE nama\_tabel DROP PRIMARY KEY;  
 Untuk menghilangkan salah satu kolom pada tabel  
 ALTER TABLE mahasiswa DROP no\_telp;

```
mysql> alter table mahasiswa drop no_telp;
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim   | char(8) | NO | PRI | NULL | |
| nama_mhs | varchar(30) | YES | | NULL | |
| alamat_mhs | varchar(35) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Gambar 7.8. Fungsi DROP untuk Menghilangkan Kolom pada Tabel

## 2) Perintah DML

### a. Mengisi tabel

Ada beberapa cara dalam memasukkan data yaitu dengan menyamakan kolom dan data, menyebutkan kolom, tanpa menyebutkan kolom, memasukkan hanya sebagai pada kolom. Menyamakan Kolom dan Data, perintahnya sebagai berikut:

```
INSERT INTO nama_tabel SET
kolom_pertama = 'data_kolom_pertama',
kolom_kedua = 'data_kolom_kedua',
kolom_terakhir = 'data kolom terakhir';
INSERT INTO mhs SET NIM = '11010010', nama_mhs = 'M Rauf', alamat_mhs = 'Puri Kencana B2 Sleman';
```

Menyebutkan Kolom, perintahnya sebagai berikut:

```
INSERT INTO nama_tabel (kolom_pertama, kolom_kedua, kolom_terakhir) VALUES
(data_kolom_pertama, data_kolom_kedua, data_kolom_terakhir);
INSERT INTO mhs (NIM, nama_mhs, alamat_mhs)
VALUES ('11010010', 'M Rauf', 'Puri Kencana B2 Sleman')
```

Tanpa Menyebutkan Kolom

```
INSERT INTO nama_tabel VALUES (data_kolom_pertama,
data_kolom_kedua, data_kolom_terakhir);
INSERT INTO mhs VALUES ('11010010', 'M Rauf', 'Puri Kencana B2 Sleman')
```

```
mysql> insert into mhs values ('11010010', 'M Rauf', 'Puri Kencana B2 Sleman');
Query OK, 1 row affected (0.03 sec)

mysql> insert into mhs values ('11010011', 'Rafifah Azzahra', 'Blunyah Gede 227 J
ogja');
Query OK, 1 row affected (0.33 sec)
```

### b. Melihat isi tabel

Perintah ini digunakan untuk menyeleksi atau memilih atau menampilkan data-data yang ada dalam tabel. baik menampilkan semua kolom, sebagian kolom, serta berdasarkan kondisi.

1. Perintah untuk menampilkan data apa adanya, tanpa syarat, tanpa pemilihan kolom dan tanpa urutan :

```
SELECT * FROM <nama_tabel>
```

```
SELECT * FROM mhs;
```

## 2. Membatasi jumlah record yang dibaca

Untuk membatasi record yang muncul atau untuk mencari record dengan kriteria tertentu, digunakan klausa where. Misal untuk melihat nama mahasiswa dengan nim = '11010010'

```
SELECT * from mhs WHERE nim = '11010010'
```

Melihat data mahasiswa yang bernama 'M Rauf'

```
SELECT * from mhs where nama_mhs = 'M Rauf'
```

Melihat data mahasiswa yang beralamat di Sleman

```
SELECT * from mhs where alamat_mhs like '%Sleman'
```

Pada kriteria alamat, terlihat ada penggunaan karakter '%'. Karakter ini mengandung makna, apa pun teks yang ada akan memenuhi kriteria. Jadi '%Sleman' artinya semua string yang diakhiri kata Sleman.

```
mysql> select * from mhs;
+----+-----+-----+
| nim   | nama_mhs | alamat_mhs |
+----+-----+-----+
| 11010010 | M Rauf   | Puri Kencana B2 Sleman |
| 11010011 | Rafifah Azzahra | Blunyah Gede 227 Jogja |
+----+-----+-----+
2 rows in set (0.02 sec)

mysql> select * from mhs where nim='11010010';
+----+-----+-----+
| nim   | nama_mhs | alamat_mhs |
+----+-----+-----+
| 11010010 | M Rauf   | Puri Kencana B2 Sleman |
+----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 7.9. Perintah Tampil Data

## 3. Membatasi jumlah field yang dibaca

Untuk melihat field-field tertentu dari tabel, gantikan karakter '\*' dengan nama field yang dikehendaki.

Melihat hanya nim dan nama mahasiswa

```
SELECT nim, nama_mhs from mhs;
```

### c. Menampilkan data berurutan

Klausa order by digunakan untuk mengurutkan data yang diminta dengan query. Misal diminta untuk menampilkan nim dan nama mahasiswa yangurut oleh nim secara menaik:

```
SELECT nim, nama_mhs from mhs order by nim asc;
```

Jika urut menurun:

```
SELECT nim, nama_mhs from mhs order by nim desc;
```

```
mysql> select nim, nama_mhs from mhs order by nim asc;
+-----+-----+
| nim   | nama_mhs |
+-----+-----+
| 11010010 | M Rauf |
| 11010011 | Rafifah Azzahra |
| 11010012 | M Tedy Farhan |
| 11010013 | Shafa Dian |
| 11010014 | Austin A Cetta |
+-----+-----+
5 rows in set (0.00 sec)

mysql> select nim, nama_mhs from mhs order by nim desc;
+-----+-----+
| nim   | nama_mhs |
+-----+-----+
| 11010014 | Austin A Cetta |
| 11010013 | Shafa Dian |
| 11010012 | M Tedy Farhan |
| 11010011 | Rafifah Azzahra |
| 11010010 | M Rauf |
+-----+-----+
5 rows in set (0.00 sec)
```

Gambar 7.10. Tampil Data Berdasarkan Urutan

## d. Mengubah isi data tabel

UPDATE nama\_tabel SET

kolom\_pertama = 'data\_kolom\_pertama',

kolom\_kedua = 'data\_kolom\_kedua',

kolom\_terakhir = 'data\_kolom\_terakhir',

WHERE kondisi

Penggunaan perintah UPDATE tanpa klausa WHERE mengakibatkan semua data dalam satu kolom akan diubah.

Misal akan diubah nama 'M Rauf' menjadi 'Muhammad Rauf'

UPDATE mhs SET nama\_mhs = 'M Rauf' where nim = '11010010'

```
mysql> update mhs set nama_mhs = 'Muhammad Rauf' where nim='11010010';
Query OK, 1 row affected (0.34 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select nim, nama_mhs from mhs order by nim desc;
+-----+-----+
| nim   | nama_mhs |
+-----+-----+
| 11010014 | Austin A Cetta |
| 11010013 | Shafa Dian |
| 11010012 | M Tedy Farhan |
| 11010011 | Rafifah Azzahra |
| 11010010 | Muhammad Rauf |
+-----+-----+
5 rows in set (0.00 sec)
```

Gambar 7.11. Tampil Data Urut Abjad

```
UPDATE employees
SET job_id = (SELECT job_id
FROM employees
WHERE employee_id = 205),
salary = (SELECT salary
FROM employees
WHERE employee_id = 205)
WHERE employee_id = 114;
1 rows updated
```

Gambar 7.12. Update nilai employees

**e. Menghapus isi tabel**

DELETE FROM nama\_tabel WHERE kondisi

```
DELETE FROM departments
WHERE department_name = 'Finance';
1 rows deleted
```

Apabila tidak menggunakan klausa WHERE maka akan menyebabkan semua record dalam tabel terhapus. Misal akan dihapus data mahasiswa bernama 'M Rauf'

DELETE FROM mhs WHERE nim = '11010012'

## 7.5. TUGAS

Tugas diberikan asisten praktikum.

**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 7: DDL DAN DML**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--

## PRAKTIKUM 8: FUNGSI AGREGASI

Pertemuan ke : 8

Total Alokasi Waktu : 180 menit (Alokasi waktu disesuaikan dengan RPS)

- Pre-Test : 30 menit
- Praktikum : 120 menit
- Post-Test : 30 menit

Total Skor Penilaian : 100% (Bobot skor disesuaikan dengan RPS)

- Pre-Test : 20 %
- Praktikum : 50 %
- Post-Test : 30 %

### 8.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu menerapkan fungsi:

1. *AVG*
2. *Count*
3. *MAX*
4. *MIN*
5. *SUM*

Indikator ketercapaian diukur dengan praktikan mampu menerapkan fungsi agregasi dengan perintah SQL

### 8.2. TEORI PENDUKUNG

Fungsi agregasi meliputi :

1. Menghitung banyak record
2. Menghitung total nilai suatu atribut
3. Menghitung rata-rata nilai atribut
4. Mencari nilai terbesar dari nilai atribut
5. Mencari nilai terkecil dari nilai atribut

Berikut bahasa query formal untuk fungsi di atas :

Tabel 8.1. Query Formal

KLAUSA	PENJELASAN
AVG	Sama dengan
COUNT	Mengetahui jumlah record
MAX	Mengetahui nilai maximal
MIN	Mengetahui nilai minimal
SUM	Menghitung jumlah data



Berikut beberapa operator yang biasanya diikuti Klausa WHERE :

Tabel 8.2. Operator MySQL

OPERATOR	PENJELASAN
=	Sama dengan
< >, !=	Tidak sama dengan
<	Kurang dari
>	Lebih besar dari
< =	Kurang dari atau sama dengan
> =	Lebih dari atau sama dengan
! >	Tidak lebih besar dari
! <	Tidak lebih kecil dari
BETWEEN	Antara dua nilai yang ditentukan
LIKE	Menyesuaikan nilai yang ditentukan
IS NULL	Nilainya adalah NULL
IN	Nilainya ditentukan dalam sebuah daftar
NOT	Negasi dari sebuah operator perbandingan
AND	Merangkai kriteria pencarian
OR	Memastikan bahwa criteria pencarian adalah eksklusif

### 8.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. XAMPP.
3. Browser (firefox, chrome).

### 8.4. LANGKAH PRAKTIKUM

Langkah praktikum berisi tahapan secara rinci bagaimana praktikum dijalankan dan apa hasil yang harus dicapai dari setiap langkah.

Buat tabel mata\_kuliah dan isikan seperti gambar di bawah :

```
mysql> select * from mata_kuliah;
+----+-----+-----+-----+
| kode_kul | nama_kul | sks | sem |
+----+-----+-----+-----+
| IT0101 | Logika Inf | 3 | 1 |
| IT0102 | Studi Islam I | 2 | 1 |
| IT0103 | Kalkulus Inf | 3 | 1 |
| IT0301 | Sertifikasi | 0 | 3 |
| IT0401 | Basis Data | 3 | 4 |
| IT0801 | Tugas Akhir | 6 | 8 |
+----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Gambar 8.3. Tampil Seluruh Data

1. Menampilkan data mata kuliah yang dilaksanakan di semester 1.  
SELECT \* FROM mata\_kuliah WHERE sem = 1;
2. Menampilkan data mata kuliah yang dilaksanakan selain semester 1.

SELECT \* FROM mata\_kuliah WHERE sem <>1;

3. Menampilkan data mata kuliah yang mengandung judul informatika

SELECT \* FROM mat\_kul WHERE nama\_kul LIKE '%informatika%'

```
mysql> select * from mata_kuliah where sem=1;
+----+-----+-----+-----+
| kode_kul | nama_kul | sks | sem |
+----+-----+-----+-----+
| IT0101 | Logika Inf | 3 | 1 |
| IT0102 | Studi Islam I | 2 | 1 |
| IT0103 | Kalkulus Inf | 3 | 1 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from mata_kuliah where sem<>1;
+----+-----+-----+-----+
| kode_kul | nama_kul | sks | sem |
+----+-----+-----+-----+
| IT0301 | Sertifikasi | 0 | 3 |
| IT0401 | Basis Data | 3 | 4 |
| IT0801 | Tugas Akhir | 6 | 8 |
+----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> select * from mata_kuliah where nama_kul like '%Inf';
+----+-----+-----+-----+
| kode_kul | nama_kul | sks | sem |
+----+-----+-----+-----+
| IT0101 | Logika Inf | 3 | 1 |
| IT0103 | Kalkulus Inf | 3 | 1 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Gambar 8.4. Tampil Data dengan Kondisi

4. Menampilkan data mata kuliah yang mengandung judul informatika dan sksnya = 3

SELECT \* FROM mat\_kul WHERE nama\_kul LIKE '%informatika%' AND sks = 3

5. Menghitung jumlah data mata kuliah

SELECT COUNT(\*) from mat\_kul;

6. Menghitung sks yang paling sedikit, sks yang paling banyak dan rata-rata sks

SELECT MIN (sks), MAX (sks), AVG (sks) from mat\_kul;

7. Menghitung total jumlah sks

SELECT SUM (sks) from mat\_kul;

```
mysql> select count(*) from mata_kuliah;
+-----+
| count(*) |
+-----+
| 6 |
+-----+
1 row in set (0.00 sec)

mysql> select min(sks), max(sks), avg(sks) from mata_kuliah;
+-----+-----+-----+
| min(sks) | max(sks) | avg(sks) |
+-----+-----+-----+
| 0 | 6 | 2.83333333333333 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select sum(sks) from mata_kuliah;
+-----+
| sum(sks) |
+-----+
| 17 |
+-----+
1 row in set (0.00 sec)
```

Gambar 8.5. Tampil Data dengan Kondisi SUM

## 8.5. TUGAS

Tugas diberikan asisten praktikum.

**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 8 : FUNGSI AGREGASI**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--

## PRAKTIKUM 9: QUERY DARI RELASI TABEL

Pertemuan ke : 9

Total Alokasi Waktu : 180 menit (Alokasi waktu disesuaikan dengan RPS)

- Pre-Test : 30 menit
- Praktikum : 120 menit
- Post-Test : 30 menit

Total Skor Penilaian : 100% (Bobot skor disesuaikan dengan RPS)

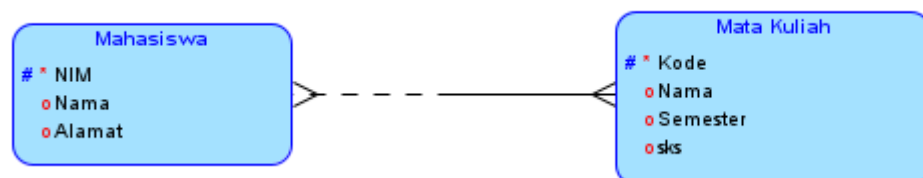
- Pre-Test : 20 %
- Praktikum : 50 %
- Post-Test : 30 %

### 9.1. TUJUAN DAN INDIKATOR CAPAIAN

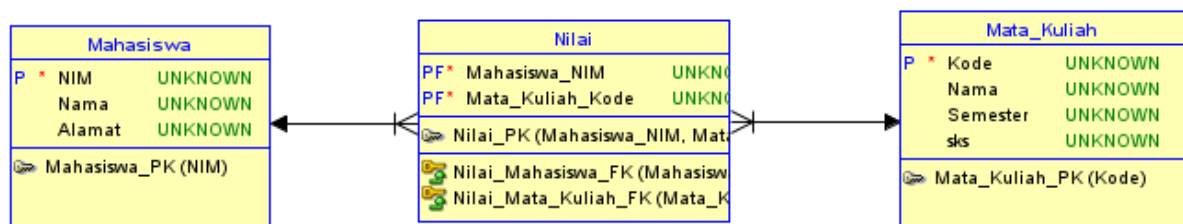
Setelah mengikuti praktikum ini mahasiswa diharapkan mampu menerapkan Relasi antar tabel Indikator ketercapaian diukur dengan dapat melakukan pencarian data antar tabel.

### 9.2. TEORI PENDUKUNG

Pada perancangan basis data dengan ERD akan mencerminkan table yang akan dibentuk pada basis data fisik. ERD merupakan relasional model yang dibentuk dari logical mode lberikut gambarannya :



Gambar 9.1. Logical Model



Gambar 9.2. Relasional Model

Query terhadap 2 tabel atau lebih tidak bisa dilakukan sembarangan. Tabel-tabel yang menjadi sumber query harus memiliki keterhubungan (relasi). NIM adalah primary key dari tabel mhs dan kode\_kul adalah primary key dari mata\_kuliah. Tabel nilai adalah tabel yang menjadi relasi antara tabel mhs dan mata\_kuliah, sehingga primary key dari tabel nilai diambil dari primary key tabel mhs dan mata\_kuliah,

yaitu nim dan kode\_kul. Artinya nim adalah *foreign key* ke tabel mhs dan juga kode\_kul adalah *foreign key* ke tabel mata\_kuliah.

Perintah untuk menampilkan data dari beberapa tabel, yaitu :

```
SELECT
    <tabel1>.<kolom_1>, <tabel1>.<kolom_2>, <tabel1>.<kolom_n>,
    <tabel2>.<kolom_1>, <tabel2>.<kolom_2>, <tabel2>.<kolom_n>,
    <tabeln>.<kolom_1>, <tabeln>.<kolom_2>, <tabeln>.<kolom_n>
FROM
    <tabel1>, <tabel2>, <tabeln>
WHERE
    <tabel1>.<kolom_x> = <tabel2>.<kolom_y> and
    <tabel2>.<kolom_y> = <tabeln>.<kolom_n> and
    <tabel1>.<kolomx> = <tabeln>.<kolomn>
```

Kondisi yang dituliskan pada WHERE adalah atribut yang menjadi relasi yang menghubungkan antara tabel 1, tabel 2 sampai tabel n.

Untuk contoh kasus tabel di atas, perintah WHERE dapat ditulis :

WHERE mhs.nim = nilai.nim and nilai.kode\_kul = mata\_kuliah.kode\_kul

### 9.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. DB MARIA / MYSQL.
3. Browser.

### 9.4. LANGKAH PRAKTIKUM

Langkah praktikum berisi tahapan secara rinci bagaimana praktikum dijalankan dan apa hasil yang harus dicapai dari setiap langkah.

Buatlah tabel berikut :

Mhs (nim, nama\_mhs, alamat\_mhs)

Mata\_Kuliah (kode\_kul, nama\_kul, sks, semester)

Nilai (nim, kode\_kul, nilai)

```
mysql> show tables;
+-----+
| Tables_in_akademik |
+-----+
| mata_kuliah         |
| mhs                 |
| nilai               |
+-----+
3 rows in set (0.00 sec)
```

Gambar 9.3. Show Tabel

1. Membuat tabel nilai dengan menyatakan bahwa nim dan kode\_kul adalah merupakan foreign key dari tabel mhs dan mata\_kuliah.

```
mysql> create table nilai (nim varchar(8), kode_kul varchar(6), nilai int, foreign key fk_nim(nim) references mhs(nim), foreign key fk_kode_kul(kode_kul) references mata_kuliah(kode_kul));
Query OK, 0 rows affected (0.09 sec)

mysql> desc nilai;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim    | varchar(8) | YES  | MUL | NULL    |       |
| kode_kul | varchar(6) | YES  | MUL | NULL    |       |
| nilai  | int(11)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> insert into nilai values ('11010010', 'IT0101', 100);
Query OK, 1 row affected (0.03 sec)

mysql> insert into nilai values ('11010020', 'IT0101', 100);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails ('akademik/nilai', CONSTRAINT 'nilai_ibfk_1' FOREIGN KEY ('nim') REFERENCES 'mhs' ('nim'))
```

Gambar 9.4. Masukkan Data

2. Isikan tabel seperti isian gambar di bawah :

```
mysql> select * from mhs;
+-----+-----+-----+
| nim    | nama_mhs | alamat_mhs |
+-----+-----+-----+
| 11010010 | M Rauf | Puri Kencana B2 Sleman |
| 11010011 | Rafifah Azzahra | Blunyah Gede 227 Sleman |
| 11010012 | M Tedy Farhan | Jl Boegenville IV No.9 Jambi |
| 11010013 | Shafa Dian | Jl Boegenville IV No.9 Jambi |
| 11010014 | Austin A Cetta | Palagan Regency B3 Jogja |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Gambar 9.5. Tampil Data Mahasiswa

```
mysql> select * from mata_kuliah;
+-----+-----+-----+-----+
| kode_kul | nama_kul | sks | sem |
+-----+-----+-----+-----+
| IT0101 | Logika Inf | 3 | 1 |
| IT0102 | Studi Islam I | 2 | 1 |
| IT0103 | Kalkulus Inf | 3 | 1 |
| IT0301 | Sertifikasi | 0 | 3 |
| IT0401 | Basis Data | 3 | 4 |
| IT0001 | Tugas Akhir | 6 | 8 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Gambar 9.6. Tampil Data Mata Kuliah

```
mysql> select * from nilai;
+-----+-----+-----+
| nim    | kode_kul | nilai |
+-----+-----+-----+
| 11010010 | IT0101 | 100 |
| 11010011 | IT0101 | 100 |
| 11010010 | IT0401 | 90 |
| 11010011 | IT0401 | 100 |
| 11010012 | IT0401 | 80 |
| 11010013 | IT0401 | 85 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

Gambar 9.7. Tampil Data Nilai

3. Menampilkan hanya nama mahasiswa dan nilai untuk mata kuliah dengan kode IT0401

```
mysql> select a.nama_mhs, b.nilai from mhs a, nilai b where a.nim = b.nim and
b.kode_kul = 'IT0401';
```

nama_mhs	nilai
M Rauf	90
Rafifah Azzahra	100
M Tedy Farhan	80
Shafa Dian	85

4 rows in set (0.00 sec)

Gambar 9.8. Tampil Data Hanya Atribut Tertentu 1

4. Menampilkan hanya nama mata kuliah berikut nilainya

```
mysql> select a.nama_kul, b.nilai from mata_kuliah a, nilai b where a.kode_kul =
b.kode_kul;
```

nama_kul	nilai
Logika Inf	100
Logika Inf	100
Basis Data	90
Basis Data	100
Basis Data	80
Basis Data	85

6 rows in set (0.00 sec)

Gambar 9.9. Tampil Data Hanya Atribut Tertentu 2

5. Menampilkan nama mata kuliah, nama mahasiswa dan nilai yang diperoleh

```
mysql> select a.nama_kul, b.nilai, c.nama_mhs from mata_kuliah a, nilai b, mhs
c where a.kode_kul = b.kode_kul and b.nim = c.nim;
```

nama_kul	nilai	nama_mhs
Logika Inf	100	M Rauf
Logika Inf	100	Rafifah Azzahra
Basis Data	90	M Rauf
Basis Data	100	Rafifah Azzahra
Basis Data	80	M Tedy Farhan
Basis Data	85	Shafa Dian

6 rows in set (0.00 sec)

Gambar 9.10. Tampil Data Hanya Atribut Tertentu 3

6. Menampilkan nama mata kuliah, nama mahasiswa dan nilai yang diperoleh yang diurutkan dari nilai terkecil (kolom tabel disesuaikan seperti yang dihasilkan).

```
mysql> select a.nama_kul as 'Nama Mata Kuliah', b.nilai as 'Nilai Mata Kuliah',
c.nama_mhs as 'Nama Mahasiswa' from mata_kuliah a, nilai b, mhs c where a.kode_kul = b.kode_kul and b.nim = c.nim order by nilai asc;
```

Nama Mata Kuliah	Nilai Mata Kuliah	Nama Mahasiswa
Basis Data	80	M Tedy Farhan
Basis Data	85	Shafa Dian
Basis Data	90	M Rauf
Logika Inf	100	Rafifah Azzahra
Basis Data	100	Rafifah Azzahra
Logika Inf	100	M Rauf

6 rows in set (0.00 sec)

Gambar 9.11. Tampil Data Hanya Atribut Tertentu 4

## 9.5. TUGAS

Tugas dibagikan asisten praktikum.

**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 9: QUERY DARI RELASI TABEL**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--



## PRAKTIKUM 10: RELASI TABEL DENGAN JOIN

Pertemuan ke : 10

Total Alokasi Waktu : 180 menit (Alokasi waktu disesuaikan dengan RPS)

- Pre-Test : 30 menit
- Praktikum : 120 menit
- Post-Test : 30 menit

Total Skor Penilaian : 100% (Bobot skor disesuaikan dengan RPS)

- Pre-Test : 20 %
- Praktikum : 50 %
- Post-Test : 30 %

### 10.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu menerapkan:

1. *Inner join*
2. *Left join*
3. *Right join*
4. *Full join*
5. *Union*

Indikator ketercapaian diukur dengan praktikan dapat melakukan pencarian data antar tabel.

### 10.2. TEORI PENDUKUNG

Mekanisme join dipergunakan untuk mencari data dari beberapa tabel berdasarkan hubungan logis tabel-tabel tersebut. Macam-macam Join, adalah:

1. Inner Join merupakan himpunan dalam yaitu hasil gabungan dari dua buah tabel yang saling berelasi untuk semua record yang berpasangan
2. Full Outer Join mengembalikan semua baris dari kedua tabel
3. Left Outer Join menghasilkan semua baris tabel di sebelah kiri pernyataan, dan baris-baris bersesuaian dari tabel sebelah kanan pernyataan
4. Right Outer Join menghasilkan semua baris tabel di sebelah kiri pernyataan, dan baris-baris bersesuaian dari tabel sebelah kiri pernyataan
5. Union dipergunakan untuk menggabungkan dua buah operasi query ke dalam satu buah *cursor*

### 10.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. *DB Maria / Mysql*.
3. *Browser*.

## 10.4. LANGKAH PRAKTIKUM

Langkah praktikum berisi tahapan secara rinci bagaimana praktikum dijalankan dan apa hasil yang harus dicapai dari setiap langkah. Buatlah table mahasiswa, nilai, dan mata\_kuliah seperti pada Gambar 10.1.

```
mysql> select * from mhs;
```

nim	nama_mhs	alamat_mhs
11010010	Muhammad Rauf	Puri Kencana B2 Sleman
11010011	Rafifah Azzahra	Blunyah Gede 227 Sleman
11010012	M Tedy Farhan	Jl Boegenville IU No.9 Jambi
11010013	Shafa Dian	Jl Boegenville IU No.9 Jambi
11010014	Austin A Cetta	Palagan Regency B3 Jogja
11010015	Luna Ramadhan	Jl Keparakan No.1 Samarinda

```
6 rows in set (0.00 sec)
```

```
mysql> select * from nilai;
```

nim	kode_kul	nilai
11010010	IT0101	100
11010011	IT0101	100
11010010	IT0401	90
11010011	IT0401	100
11010012	IT0401	80
11010013	IT0401	85

```
6 rows in set (0.00 sec)
```

```
mysql> select * from mata_kuliah;
```

kode_kul	nama_kul	sks	sem
IT0101	Logika Inf	3	1
IT0102	Studi Islam I	2	1
IT0103	Kalkulus Inf	3	1
IT0301	Sertifikasi	0	3
IT0401	Basis Data	3	4
IT0801	Tugas Akhir	6	8

```
6 rows in set (0.00 sec)
```

Gambar 10.1. Tabel Mahasiswa, Nilai, Mata kuliah

### 1. Inner Join

Menampilkan nama mahasiswa, kode mata kuliah berikut nilai yang diperoleh  
Dengan perintah relasi atribut antar tabel

```
mysql> select mhs.nim, mhs.nama_mhs, nilai.kode_kul, nilai.nilai
-> from mhs, nilai
-> where mhs.nim = nilai.nim;
```

nim	nama_mhs	kode_kul	nilai
11010010	Muhammad Rauf	IT0101	100
11010010	Muhammad Rauf	IT0401	90
11010011	Rafifah Azzahra	IT0101	100
11010011	Rafifah Azzahra	IT0401	100
11010012	M Tedy Farhan	IT0401	80
11010013	Shafa Dian	IT0401	85

```
6 rows in set (0.00 sec)
```

Gambar 10.2. Perintah Inner Join 1

Dengan perintah inner join

```
mysql> select mhs.nim, mhs.nama_mhs, nilai.kode_kul, nilai.nilai
-> from mhs inner join nilai
-> on mhs.nim = nilai.nim;
```

nim	nama_mhs	kode_kul	nilai
11010010	Muhammad Rauf	IT0101	100
11010010	Muhammad Rauf	IT0401	90
11010011	Rafifah Azzahra	IT0101	100
11010011	Rafifah Azzahra	IT0401	100
11010012	M Tedy Farhan	IT0401	80
11010013	Shafa Dian	IT0401	85

5 rows in set (0.00 sec)

Gambar 10.3. Perintah Inner Join 2

Dua gambar di atas menghasilkan hasil yang sama. Terlihat bahwa pada tabel mhs dan mata\_kuliah, nim adalah atribut yang menghubungkan ke dua tabel tersebut. Pada perintah di atas hanya record yang berpasangan yang dimunculkan. Pada tabel mhs terdapat record dengan nim '11010014' dan '11010015' yang tidak terdapat pada tabel nilai, sehingga tidak dimunculkan.

## 2. Left Join

Menampilkan semua data mahasiswa yang ada di tabel mhs berikut data nilai mahasiswa

```
mysql> select mhs.nim, mhs.nama_mhs, nilai.kode_kul, nilai.nilai from mhs left
join nilai on mhs.nim = nilai.nim;
```

nim	nama_mhs	kode_kul	nilai
11010010	Muhammad Rauf	IT0101	100
11010010	Muhammad Rauf	IT0401	90
11010011	Rafifah Azzahra	IT0101	100
11010011	Rafifah Azzahra	IT0401	100
11010012	M Tedy Farhan	IT0401	80
11010013	Shafa Dian	IT0401	85
11010014	Austin A Cetta	NULL	NULL
11010015	Luna Ramadhan	NULL	NULL

8 rows in set (0.00 sec)

Gambar 10.4. Perintah Left Join

Terlihat bahwa data tabel sebelah kanan akan diisi dengan NULL karena mahasiswa yang bersangkutan tidak ada nilainya, sementara data dari tabel sebelah kiri akan ditampilkan seluruhnya.

## 3. Right Join

Menampilkan semua data nilai yang ada di tabel nilai berikut nama mahasiswa.

```
mysql> select mhs.nim, mhs.nama_mhs, nilai.kode_kul, nilai.nilai from mhs right
join nilai on mhs.nim = nilai.nim;
```

nim	nama_mhs	kode_kul	nilai
11010010	Muhammad Rauf	IT0101	100
11010011	Rafifah Azzahra	IT0101	100
11010010	Muhammad Rauf	IT0401	90
11010011	Rafifah Azzahra	IT0401	100
11010012	M Tedy Farhan	IT0401	80
11010013	Shafa Dian	IT0401	85

5 rows in set (0.00 sec)

Gambar 10.5. Perintah Right Join

Akan terlihat bahwa data dari tabel sebelah kanan yaitu tabel nilai akan ditampilkan seluruhnya.

#### 4. Union

```
mysql> select * from mhs union select * from nilai;
```

nim	nama_mhs	alamat_mhs
11010010	Muhammad Rauf	Puri Kencana B2 Sleman
11010011	Rafifah Azzahra	Blunyah Gede 227 Sleman
11010012	M Tedy Farhan	Jl Boegenville IV No.9 Jambi
11010013	Shafa Dian	Jl Boegenville IV No.9 Jambi
11010014	Austin A Cetta	Palagan Regency B3 Jogja
11010015	Luna Ramadhan	Jl Keparakan No.1 Samarinda
11010010	IT0101	100
11010011	IT0101	100
11010010	IT0401	90
11010011	IT0401	100
11010012	IT0401	80
11010013	IT0401	85

12 rows in set (0.02 sec)

Gambar 10.6. Perintah Union

## 10.5. TUGAS

Tugas diberikan asisten praktikum.

**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 10: RELASI TABEL DENGAN JOIN**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--

## PRAKTIKUM 11: SUBQUERY

---

Pertemuan ke	: 11
Total Alokasi Waktu	: 180 menit
• Pre-Test	: 30 menit
• Praktikum	: 120 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 50 %
• Post-Test	: 30 %

---

### 11.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu menerapkan:

1. *IN*
2. *NOT IN*
3. *EXIST*
4. Operator Perbandingan

Indikator ketercapaian diukur dengan:

1. Praktikan memahami konsep SubQuery.
2. Praktikan dapat membuat perintah SubQuery.

### 11.2. TEORI PENDUKUNG

Sub Query adalah suatu query yang menjadi bagian dari suatu query, digunakan untuk menangani masalah yang kompleks yang mungkin sulit untuk dilakukan hanya dengan sebuah query. Sub Query menyediakan cara alternatif untuk melakukan operasi yang membutuhkan join atau union yang rumit.

Beberapa aturan :

1. Dalam sebuah query boleh memiliki lebih dari 1 sub query.
2. Sebuah sub query boleh memiliki sub query lagi.
3. Operator perbandingan yang dapat digunakan adalah =, >, <, >=, <=, <>, !=, <=>, IN, ANY, SOME, ALL, EXISTS, NOT EXISTS

Untuk operator =, >, <, >=, <=, <>, !=, <=> sub querynya hanya boleh memiliki 1 baris, jika barisnya memiliki lebih dari 1 baris akan menampilkan pesan "Subquery returns more than 1 row".

Untuk operator IN akan memeriksa apakah suatu nilai di outer query ada dalam sebuah hasil sub query. Sub query boleh memiliki data lebih dari 1 baris. Operator IN bisa disamakan dengan operator "= ANY". Lawan hasil dari operasi IN adalah NOT IN. Operator NOT IN bisa disamakan dengan "<> ALL"

Untuk operator EXISTS digunakan untuk memeriksa apakah subquery memiliki baris atau tidak. Jika minimal ada 1 baris (walaupun hanya berisi NULL), maka akan bernilai TRUE. NOT EXISTS adalah kebalikan dari EXISTS.

### 11.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. *DB Maria / Mysql*.
3. *Browser*.

### 11.4. LANGKAH PRAKTIKUM

1. Buatlah dan Isikan tabel nilai seperti Gambar 11.1:

```
mysql> select * from nilai;
+-----+-----+-----+
| nim      | kode_kul | nilai |
+-----+-----+-----+
| 11010010 | IT0101   | 80    |
| 11010011 | IT0101   | 75    |
| 11010010 | IT0401   | 70    |
| 11010011 | IT0401   | 75    |
| 11010012 | IT0401   | 70    |
| 11010013 | IT0401   | 70    |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

Gambar 11.1 Tabel Nilai Mahasiswa

2. Menampilkan data nilai mahasiswa yang nilainya melebihi rata-rata nilai mata kuliah secara keseluruhan. Langkahnya:
  - Membuat Query untuk mencari nilai rata-rata mata kuliah. Query ini akan digunakan sebagai acuan untuk mencari nilai yang memiliki nilai di atas rata rata, sehingga query dengan model seperti ini dinamakan Sub Query.
  - Dari Sub Query tersebut akan di masukan dalam Query utama (Main Query). Main Query ini untuk mencari nilai yang memiliki nilai di atas rata rata.
  - Lihat gambar berikut ini:

Sub Query : `SELECT avg(nilai) from nilai`

```
mysql> select avg(nilai) from nilai;
+-----+
| avg(nilai) |
+-----+
| 73.3333    |
+-----+
1 row in set (0.05 sec)
```

Gambar 11.2. Sub Query

Main Query : `SELECT nim, nilai FROM nilai WHERE nilai> (SELECT avg(nilai) from nilai)`

```
mysql> select nim, nilai from nilai where nilai>(select avg(nilai) from nilai);
+-----+-----+
| nim      | nilai |
+-----+-----+
| 11010010 | 80    |
| 11010011 | 75    |
| 11010011 | 75    |
+-----+-----+
3 rows in set (0.03 sec)
```

Gambar 11.3. Main Query

3. Carilah data nilai yang nilainya sama dengan nilai terbesar.  
Query membutuhkan Sub Query karena untuk dapat mencari data nilai yang diinginkan, maka nilai terbesar harus dicari terlebih dahulu.

Query ini bisa dilakukan dengan menggunakan ORDER BY dan LIMIT, tetapi hanya akan menghasilkan 1 baris saja. Bagaimana jika data yang sesuai dengan kriteria lebih dari 1 baris.

```
mysql> select * from nilai;
```

nim	kode_kul	nilai
11010010	IT0101	80
11010011	IT0101	75
11010010	IT0401	70
11010011	IT0401	75
11010012	IT0401	70
11010013	IT0401	80

```
6 rows in set (0.00 sec)
```

Gambar 11.4. Tabel Nilai

```
SELECT nim, nilai, kode_kul
FROM nilai WHERE nilai=(SELECT MAX(nilai) FROM nilai)
```

```
mysql> select nim, nilai, kode_kul from nilai where nilai = (select max(nilai) f
rom nilai);
```

nim	nilai	kode_kul
11010010	80	IT0101
11010013	80	IT0401

```
2 rows in set (0.00 sec)
```

Gambar 11.5. Tabel hasil Query tanpa Limit

```
SELECT nim, nilai, kode_kul
FROM nilai ORDER BY nilai ASC LIMIT 1
```

Query pertama mungkin menampilkan data nilai lebih dari 1 baris ketika baris yang nilainya sama dengan nilai MAX(nilai) lebih dari 1 baris.

Query kedua hanya akan menampilkan 1 baris saja karena ada penggunaan LIMIT. Kekurangan dari SQL ini adalah ketika ada data nilai yang sama-sama memiliki nilai sama dengan MAX(nilai) lebih dari 1 baris.

```
mysql> select * from nilai order by nilai desc limit 1;
```

nim	kode_kul	nilai
11010010	IT0101	80

```
1 row in set (0.00 sec)
```

Gambar 11.6. Tabel hasil Query dengan Limit

#### 4. Menampilkan data mahasiswa yang mengambil mata kuliah 'IT0401'

Query tersebut membutuhkan sub query karena harus melakukan perbandingan data mahasiswa dengan data mahasiswa yang ada di tabel kuliah. Berarti data mahasiswa yang ada di tabel kuliah harus dicari terlebih dahulu.

```
SELECT nim, nama_mhs
FROM mhs
WHERE nim IN (SELECT nim FROM kuliah WHERE kode_kul = 'IT100')
```



Sub query bekerja untuk mencari data mhs yang telah terdaftar di tabel kuliah (mengambil mata kuliah tersebut)

```
mysql> select nim, nama_mhs from mhs where nim in (select nim from nilai where
kode_kul = 'IT0401');
+-----+-----+
| nim      | nama_mhs |
+-----+-----+
| 11010010 | Muhammad Rauf |
| 11010011 | Rafifah Azzahra |
| 11010012 | M Tedy Farhan |
| 11010013 | Shafa Dian |
+-----+-----+
4 rows in set (0.00 sec)
```

Gambar 11.7. Tabel hasil Query perbandingan dua table dengan IN

5. Menampilkan data mahasiswa yang tidak mengambil mata kuliah basis data

```
mysql> select nim, nama_mhs from mhs where nim not in (select nim from nilai wh
ere kode_kul = 'IT0401');
+-----+-----+
| nim      | nama_mhs |
+-----+-----+
| 11010014 | Austin A Cetta |
| 11010015 | Luna Ramadhan |
+-----+-----+
2 rows in set (0.00 sec)
```

Gambar 11.8. Tabel hasil Query Perbandingan dua Tabel dengan NOT IN

## 11.5. TUGAS

Tugas diberikan asisten praktikum.

**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 11: SUBQUERY**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--

## PRAKTIKUM 12: TRIGGER

---

Pertemuan ke	: 12
Total Alokasi Waktu	: 180 menit
• Pre-Test	: 30 menit
• Praktikum	: 120 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100%
• Pre-Test	: 20 %
• Praktikum	: 50 %
• Post-Test	: 30 %

---

### 12.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu memahami:

1. konsep trigger di dalam basis data.
2. implementasi trigger dalam merespon suatu kejadian.

Indikator ketercapaian diukur dengan:

1. Praktikan memahami konsep trigger.
2. Praktikan dapat mengimplementasikan konsep Trigger.

### 12.2. TEORI PENDUKUNG

Trigger dapat didefinisikan sebagai himpunan kode (prosedural) yang dieksekusi secara otomatis sebagai respon atas suatu kejadian berkaitan dengan tabel basis data. Kejadian (event) yang dapat membangkitkan Trigger umumnya berupa pernyataan INSERT, UPDATE, dan DELETE. Berdasarkan ruang lingkupnya, Trigger diklasifikasikan menjadi dua jenis: row trigger dan statement Trigger. Trigger baris (row) mendefinisikan aksi untuk setiap baris tabel; trigger pernyataan hanya berlaku untuk setiap pernyataan INSERT, UPDATE, atau DELETE.

Dari sisi yang lain, Trigger dapat dibedakan menjadi beberapa jenis; namun umumnya ada dua jenis: Trigger BEFORE dan AFTER. Sesuai penamaannya, jenis-jenis ini merepresentasikan waktu eksekusi Trigger— misalnya sebelum ataukah sesudah pernyataan yang berkorespondensi. Adakalanya trigger dipandang sebagai bentuk spesifik dari stored procedure (terkait pendefinisian body). Bagaimanapun, trigger akan dipanggil (secara otomatis) ketika event terjadi, sedangkan stored procedure harus dipanggil secara eksplisit. Ada beberapa event yang dapat digunakan untuk melakukan eksekusi Trigger yaitu:

- BEFORE INSERT – dijalankan ketika data di masukan ke dalam table.
- AFTER INSERT – dijalankan setelah data masuk ke dalam table.
- BEFORE UPDATE – dijalankan sebelum proses update data.
- AFTER UPDATE – dijalankan setelah proses update data.
- BEFORE DELETE – dijalankan sebelum proses delete data.
- AFTER DELETE – dijalankan setelah proses delete data.

MySQL mendukung fitur trigger termasuk juga stored procedure dan view sejak versi 5.0.2. Sebagaimana objek-objek lainnya, trigger diciptakan menggunakan pernyataan CREATE. Sintaks pendefinisian trigger diperlihatkan sebagai berikut:

```
CREATE
[DEFINER = { user | CURRENT_USER }]
TRIGGER trigger_name trigger_time trigger_event
ON tbl_name FOR EACH ROW trigger_stmt
```

MySQL tidak mengizinkan multiple trigger dengan waktu aksi dan event sama per tabel. Misalkan di tabel A sudah didefinisikan trigger AFTER INSERT, maka kita tidak boleh mendefinisikan trigger AFTER INSERT lagi; namun AFTER EDIT, AFTER DELETE, atau BEFORE (INSERT, EDIT, dan DELETE) bisa diterima.

### 12.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. *Komputer.*
2. *XAMPP (Mysql, Apache, PhpMyAdmin)*
3. *Browser.*

### 12.4. LANGKAH PRAKTIKUM

1. Buatlah dan Isikan tabel produk dan table log\_harga seperti tabel 12.1 dan 12.2:

Tabel 12.1 Tabel produk

Field	Type	Null	Key	Default	Extra
kode_produk	varchar(6)	NO	PRI	NULL	
nama_produk	varchar(100)	NO		NULL	
harga	int(11)	NO		NULL	

Tabel 12.2. Tabel log\_harga

Field	Type	Null	Key	Default	Extra
log_id	int(11)	NO	PRI	NULL	auto_increment
kode_produk	varchar(8)	NO		NULL	
harga_lama	int(11)	NO		NULL	
harga_baru	int(11)	NO		NULL	
waktu_perubahan	datetime	NO		NULL	

2. Buatlah Trigger untuk mencatat perubahan harga Ketika ada perintah update pada produk. Langkahnya dengan membuat sintaks sebagai berikut:

```
DELIMITER $$
CREATE TRIGGER before_produk_update
BEFORE UPDATE ON produk
FOR EACH ROW
BEGIN
INSERT INTO log_harga_produk
set kode_produk = OLD.kode_produk,
harga_baru=new.harga,
```

```

    harga_lama=old.harga,
    waktu_perubahan = NOW();
END$$
DELIMITER ;

```

Penjelasan sintaks:

- baris 2 – Kita membuat sebuah Trigger baru dengan nama before\_produk\_update
- baris 3 – Pada Trigger ini kita menggunakan event BEFORE UPDATE
- baris 6 – Query SQL untuk melakukan insert data ke tabel log\_harga\_produk

### 3. Isikan data

Isikan data ke table produk dengan perintah sebagai berikut:

```

INSERT INTO `produk` VALUES ('BR001', 'SEMINGGU JAGO CODEIGNITER',120000);
INSERT INTO `produk` VALUES ('BR002',' SEMINGGU JAGO PHP MYSQL',80000);

```

Setelah itu, cek isi table dengan perintah:

```
SELECT * FROM produk;
```

```
mysql> select * from produk;
```

kode_produk	nama_produk	harga
BR001	SEMINGGU JAGO CODEIGNITER	120000
BR002	SEMINGGU JAGO PHP MYSQL	80000

### 4. Kemudian lakukan UPDATE Data produk dengan tujuan mengganti harga BR001 dari 12000 menjadi 90000 dengan perintah:

```
UPDATE produk SET harga=90000 WHERE kode_produk='BR001'
```

Lakukan pengecekan terhadap perintah di atas dengan memberi perintah:

```
SELECT * FROM produk;
```

```
mysql> select * from produk;
```

kode_produk	nama_produk	harga
BR001	SEMINGGU JAGO CODEIGNITER	90000
BR002	SEMINGGU JAGO PHP MYSQL	80000

### 5. Kemudian lakukan pengecekan pada table log\_harga\_produk dengan perintah:

```
SELECT * FROM log_harga_produk
```

Maka akan didapatkan hasil:

```
mysql> select * from log_harga_produk;
```

log_id	kode_produk	harga_lama	harga_baru	waktu_perubahan
7	BR001	120000	90000	2016-10-27 01:18:56

6. Jika ingin mendapatkan informasi mengenai list Trigger yang ada pada suatu basisdata, maka dapat dilakukan dengan memberikan perintah: SHOW TRIGGERS
7. Sedangkan untuk melakukan penghapusan suatu Trigger dari Basisdata yang sudah dibuat yaitu dengan perintah:

```
DROP TRIGGER nama_trigger;
//contoh implementasinya
DROP TRIGGER before_produk_update;
```

## 12.5. TUGAS

Tugas diberikan asisten praktikum.

Silahkan lakukan percobaan dengan perintah Trigger:

BEFORE DELETE – dijalankan sebelum proses delete data.

AFTER DELETE – dijalankan setelah proses delete data

**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 12: TRIGGER**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--

## PRAKTIKUM 13: PHP dan MySQL

---

Pertemuan ke	: 13
Total Alokasi Waktu	: 180 menit
• Pre-Test	: 30 menit
• Praktikum	: 120 menit
• Post-Test	: 30 menit
Total Skor Penilaian	: 100% (Bobot skor disesuaikan dengan RPS)
• Pre-Test	: 20 %
• Praktikum	: 50 %
• Post-Test	: 30 %

---

### 13.1. TUJUAN DAN INDIKATOR CAPAIAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu menerapkan:

1. *PHP*
2. *MySQL*
3. *HTML*

Indikator ketercapaian diukur dengan:

1. Praktikan mampu membuat koneksi pada *database* menggunakan *PHP* objek.
2. Praktikan mampu membuat query CRUD pada *database* menggunakan *PHP* objek.

### 13.2. TEORI PENDUKUNG

**PHP** adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP disebut bahasa pemrograman **server side** karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client).

Objek adalah sekumpulan software yang terdiri dari variable dan method-method yang terkait. Objek juga merupakan benda nyata yang di buat berdasarkan rancangan yang di definisikan di dalam class. Object adalah instance dari class. Jika class secara umum mepresentasikan (template) sebuah object, sebuah instance adalah representasi nyata dari class itu sendiri.

Class adalah prototype, atau blueprint, atau rancangan yang mendefinisikan variable dan method-methode pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program yang akan di jalankan, di dalamnya berisi atribut / type data dan method untuk menjalankan suatu program. Sedangkan method merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu object. Method didefinisikan pada class akan tetapi dipanggil melalui object.



### 13.3. ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. XAMPP.
3. Browser (*firefox, chrome*).

### 13.4. LANGKAH PRAKTIKUM

Langkah praktikum berisi tahapan secara rinci bagaimana praktikum dijalankan dan apa hasil yang harus dicapai dari setiap langkah.

#### 1. Membuat Class pada PHP

Membuat folder yang berada pada "C:\\xampp\\htdocs\\" yang digunakan untuk mengakses file php didalamnya melalui web browser. Membuat file php yang akan digunakan sebagai tempat pembuatan class dalam PHP, misalkan dengan nama 'database.php'.

Tabel 13.1. Buat Database

<?php
<b>Class Database</b> {
<b>function __construct</b> () {
\$this->db = <b>new</b> mysqli("localhost", "root", "", "176_db");
}
}
?>

#### 2. Menampilkan Data Mahasiswa

Membuat function yang digunakan untuk menampilkan data mahasiswa.

Tabel 13.2. Tampil Data

<b>function tampilMhs</b> () {
\$array = <b>array</b> ();
\$query = \$this->db-> <b>query</b> ("SELECT * FROM mahasiswa");
<b>while</b> (\$data = mysqli_fetch_array(\$query)) {
\$array[] = \$data;
}
<b>return</b> \$array;
}

Kemudian membuat file index.php yang akan digunakan untuk menampilkan data dari function tampilMhs ke dalam web.

Tabel 13.3. Inisialisasi

<?php
<b>require</b> 'database.php';
\$objek = <b>new</b> Database();
\$dataMhs = \$objek->tampilMhs();
?>

Source code diatas berfungsi untuk memanggil file database.php kemudian dibuat objek dari class Database. Kemudian membuat source code HTML dan PHP untuk menampilkan hasil dari data yang akan ditampilkan.

Tabel 13.4. Buat Tabel

<table border="1">
<tr>
<th>NIM</th>
<th>Nama</th>
<th>Program Studi</th>
<th>Alamat</th>
</tr>
<?php foreach (\$dataMhs as \$x) {?
<tr>
<td><?php echo \$x['nim'];?></td>
<td><?php echo \$x['nama'];?></td>
<td><?php echo \$x['prodi'];?></td>
<td><?php echo \$x['alamat'];?></td>
</tr>
<?php } ?>
</table>

Fungsi foreach digunakan untuk memecah data array yang dikembalikan oleh function tampil Mhs pada class Database.

NIM	Nama	Program Studi	Alamat
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul

Gambar 13.1. Tampil Data di Web

### 3. Menambah Data Mahasiswa

Hal yang harus dilakukan terlebih dahulu yaitu membuat form input mahasiswa dengan menambahkan source code dibawah ini pada index.php.

Tabel 13.5. Tambah Data

<form method="POST" action="">
<table>
<tr>
<td>NIM</td>
<td>:</td>
<td><input type="text" name="nim"></td>
</tr>
<tr>
<td>Alamat</td>
<td>:</td>

<td><input type="text" name="nama"></td>
</tr>
<tr>
<td>Program Studi</td>
<td>:</td>
<td><input type="text" name="prodi"></td>
</tr>
<tr>
<td>Alamat</td>
<td>:</td>
<td><input type="text" name="alamat"></td>
</tr>
<tr><td colspan="3" align="right"><input type="submit" name="simpan" value="SIMPAN"></td></tr>
</table>
</form>

NIM	Nama	Program Studi	Alamat
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul

NIM :

Alamat :

Program Studi :

Alamat :

Gambar 13.2. Input Data

Kemudian membuat function pada class database yang berfungsi untuk menginputkan data ke dalam database.

Tabel 13.6. Isi Data

<b>function tambahMhs(\$nim, \$nama, \$prodi, \$alamat){</b>
<b>\$insert = \$this-&gt;db-&gt;query("INSERT INTO mahasiswa (nim, nama, prodi, alamat) VALUES ('\$nim','\$nama','\$prodi','\$alamat')");</b>
<b>if (\$insert) {</b>
header('location:index.php');
<b>}else{</b>
<b>echo "data gagal ditambahkan";</b>
<b>echo "&lt;br&gt;\$nim \$nama \$prodi \$alamat";</b>
<b>echo \$this-&gt;db-&gt;error;</b>
<b>}</b>
<b>}</b>

Kemudian menambahkan source code pada index.php yang berfungsi untuk mengantarkan data pada function yang sudah dibuat pada langkah sebelumnya.

Tabel 13.7. Inisialisasi Variabel

```
<?php
if (isset($_POST['simpan'])) {
    $nim = $_POST['nim'];
    $nama = $_POST['nama'];
    $prodi = $_POST['prodi'];
    $alamat = $_POST['alamat'];
    $objek->tambahMhs($nim, $nama, $prodi, $alamat);
}
?>
```

NIM	Nama	Program Studi	Alamat
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul

NIM :

Alamat :

Program Studi :

Alamat :

Gambar 13.3. Tambah Data di Web

NIM	Nama	Program Studi	Alamat
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul
1400018182	Andi Eko	Teknik Informatika	Lampung

NIM :

Alamat :

Program Studi :

Alamat :

Gambar 13.4. Tampil Data Setelah Edit

#### 4. Mengubah dan Menghapus Data Mahasiswa

Mengubah syntax html pada index.php yang menampilkan tabel data mahasiswa, seperti di bawah ini.

Tabel 13.8. Codingan Edit Data

<tr>
<th>NIM</th>
<th>Nama</th>
<th>Program Studi</th>
<th>Alamat</th>
<th>Aksi</th>
</tr>
<?php foreach (\$dataMhs as \$x) {?>
<tr>
<td><?php echo \$x['nim'];?></td>
<td><?php echo \$x['nama'];?></td>
<td><?php echo \$x['prodi'];?></td>
<td><?php echo \$x['alamat'];?></td>
<td><a href="edit.php?nim=<?php echo \$x['nim'];?>">Edit</a>   <a href="index.php?aksi=delete&nim=<?php echo \$x['nim'];?>">Hapus</a></td>
</tr>

Kemudian membuat function update pada class Database.

<b>function</b> updateMhs(\$nim, \$nama, \$prodi, \$alamat){
\$insert = \$this->db->query("UPDATE mahasiswa SET nama = '\$nama', prodi = '\$prodi', alamat = '\$alamat' WHERE nim = '\$nim'");
<b>if</b> (\$insert) {
header('location:index.php');
} <b>else</b> {
<b>echo</b> "data gagal ditambahkan";
<b>echo</b> " \$nim \$nama \$prodi \$alamat";
<b>echo</b> \$this->db->error;
}
}

Kemudian membuat file edit.php dengan syntax html dan php seperti di bawah ini.

<?php
<b>require</b> 'database.php';
\$objek = <b>new</b> Database();
\$mhs = \$objek->tampilMhsWhere(\$_GET['nim']);
<b>foreach</b> (\$mhs as \$x) {
}
<b>if</b> (isset(\$_POST['simpan'])) {
\$nim = \$_POST['nim'];
\$nama = \$_POST['nama'];
\$prodi = \$_POST['prodi'];

\$alamat = \$_POST['alamat'];
\$objek->updateMhs(\$nim, \$nama, \$prodi, \$alamat);
}
?>
<form method="POST" action="">
<table>
<tr>
<td>NIM</td>
<td>:</td>
<td><input type="text" name="nim" value="<?php echo \$x['nim'];?>" readonly></td>
</tr>
<tr>
<td>Alamat</td>
<td>:</td>
<td><input type="text" name="nama" value="<?php echo \$x['nama'];?>"></td>
</tr>
<tr>
<td>Program Studi</td>
<td>:</td>
<td><input type="text" name="prodi" value="<?php echo \$x['prodi'];?>"></td>
</tr>
<tr>
<td>Alamat</td>
<td>:</td>
<td><input type="text" name="alamat" value="<?php echo \$x['alamat'];?>"></td>
</tr>
<tr><td colspan="3" align="right"><input type="submit" name="simpan" value="SIMPAN"></td></tr>
</table>
</form>

NIM	Nama	Program Studi	Alamat	Aksi
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat	<a href="#">Edit</a>   <a href="#">Hapus</a>
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul	<a href="#">Edit</a>   <a href="#">Hapus</a>
1400018182	Andi Eko	Teknik Informatika	Lampung	<a href="#">Edit</a>   <a href="#">Hapus</a>

NIM :

Alamat :

Program Studi :

Alamat :

NIM	Nama	Program Studi	Alamat	Aksi
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat	<a href="#">Edit</a>   <a href="#">Hapus</a>
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul	<a href="#">Edit</a>   <a href="#">Hapus</a>
1400018182	Andi Eko Suryanto	Teknik Informatika	Lampung	<a href="#">Edit</a>   <a href="#">Hapus</a>

NIM :

Alamat :

Program Studi :

Alamat :

Gambar 13.5. Tampil Data Edit

Kemudian membuat function hapus pada class database yang berfungsi untuk menghapus data berdasarkan nim.

Tabel 13.9. Hapus Data

<b>function</b> hapusMhs(\$nim){
\$insert = \$this->db->query("DELETE FROM mahasiswa WHERE nim = '\$nim'");
<b>if</b> (\$insert) {
header('location:index.php');
} <b>else</b> {
<b>echo</b> "data gagal dihapus";
<b>echo</b> " \$nim";
<b>echo</b> \$this->db->error;
}
}

Kemudian menambahkan source code php pada index.php seperti di bawah ini.

```

if (isset($_GET['aksi'])) {
    if ($_GET['aksi'] == 'delete') {
        $objek->deleteMhs($_GET['nim']);
    }
}

```

NIM	Nama	Program Studi	Alamat	Aksi
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat	<a href="#">Edit</a>   <a href="#">Hapus</a>
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul	<a href="#">Edit</a>   <a href="#">Hapus</a>
1400018182	Andi Eko Suryanto	Teknik Informatika	Lampung	<a href="#">Edit</a>   <a href="#">Hapus</a>

NIM :   
 Alamat :   
 Program Studi :   
 Alamat :

NIM	Nama	Program Studi	Alamat	Aksi
1400018176	Alvinditya Saputra	Teknik Informatika	Yogyakarta Hadiningrat	<a href="#">Edit</a>   <a href="#">Hapus</a>
1400018177	Sugeng Riyadi	Teknik Informatika	Bantul	<a href="#">Edit</a>   <a href="#">Hapus</a>

Gambar 13.6. Tampil Data Hapus

### 13.5. TUGAS

Tugas diberikan oleh asisten praktikum.



**LEMBAR JAWABAN PRE-TEST / POST-TEST / EVALUASI PRAKTIKUM 13: PHP DAN MySQL**

<b>Nama :</b> <b>NIM :</b>	<b>Asisten:</b> <b>Paraf Asisten:</b>	<b>Tanggal:</b> <b>Nilai:</b>
-------------------------------	--	----------------------------------

--

## DAFTAR PUSTAKA

1. Silberschatz Abraham et.al. 1986. *Database Systems Concepts*. United State: McGraw-Hill.
2. Ramez Elmasri et.al. 1989. *Fundamentals of Database Systems*. Addison Wesley Publishing Company.
3. Oracle Academy Presentation, 2020
4. Modul Praktikum Basisdata 2017, Teknik Elektro UM
5. <https://belajarphp.net/tutorial-trigger-mysql/>

