



LABORATORIUM
S1 INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN

PETUNJUK PRAKTIKUM EDISI KURIKULUM OBE

BASIS DATA



Penyusun:

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

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

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

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

Miftahurahma Rosyda, S.Kom., M.Eng.

2023

HAK CIPTA

PETUNJUK PRAKTIKUM NAMA PRAKTIKUM

Copyright© 2023,

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

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

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

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

Miftahurahma Rosyda, S.Kom., M.Eng.

Hak Cipta dilindungi Undang-Undang

Dilarang mengutip, memperbanyak atau mengedarkan isi buku ini, baik sebagian maupun seluruhnya, dalam bentuk apapun, tanpa izin tertulis dari pemilik hak cipta dan penerbit.

Diterbitkan oleh:

Program Studi Informatika

Fakultas Teknologi Industri

Universitas Ahmad Dahlan

Jalan Ring Road Selatan, Tamanan, Banguntapan, Bantul Yogyakarta 55166

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

Dewi Soyusiawati, S.T., M.T.

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

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

Miftahurahma Rosyda, S.Kom., M.Eng.

Editor : Laboratorium S1 Informatika, Universitas Ahmad Dahlan

Desain sampul : Laboratorium S1 Informatika, Universitas Ahmad Dahlan

Tata letak : Laboratorium S1 Informatika, Universitas Ahmad Dahlan

Ukuran/Halaman : 21 x 29,7 cm /128 halaman

Didistribusikan oleh:



Laboratorium S1 Informatika

Universitas Ahmad Dahlan

Jalan Ring Road Selatan, Tamanan, Banguntapan, Bantul Yogyakarta 55166

Indonesia

KATA PENGANTAR

Puji dan syukur ke hadirat Allah SWT yang memberikan rahmat dan hidayah sehingga penyusunan revisi kedua Petunjuk Praktikum Basis Data Edisi Kurikulum *Outcome Based Education* (OBE) ini akhirnya bisa diselesaikan. Petunjuk Praktikum ini disusun sebagai panduan untuk pelaksanaan praktikum mata kuliah Basis Data di lingkungan Program Studi S1 Informatika Universitas Ahmad Dahlan.

Materi yang disajikan sudah diurutkan disesuaikan dengan RPS berbasis OBE matakuliah Basis Data, sehingga mahasiswa dapat dengan mudah memahami. Setiap pertemuan diberikan terlebih dahulu penjelasan mengenai teori terkait materi yang akan dilakukan dan dilanjutkan dengan kegiatan praktikum.

Penyusun menyadari masih ada kekurangan pada penulisan ini, 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 maupun tidak, terhadap terselesaiannya petunjuk praktikum ini.

Yogyakarta, 10 Juli 2023

Penyusun

DAFTAR PENYUSUN

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

	<p>NIDN : 0528058401 NIPM : 60160979 Jabatan Fungsional: Lektor Bidang Minat : Sistem Informasi Email : jefree.fahana[at]tif.uad.ac.id Penyusun dan editor Bab 1 dan 2 Petunjuk Praktikum Basisdata Edisi OBE</p>
---	--

Dewi Soysiwaty, S.T., M.T

	<p>NIDN : 0530077601 NIPM : 19760730 200409 011 0951361 Jabatan Fungsional: Lektor Bidang Minat : Sistem Cerdas, Pemrosesan Bahasa Alami Email : dewi.soyusiwaty[at]tif.uad.ac.id Penyusun dan editor Bab 1 dan 2 Petunjuk Praktikum Basisdata Edisi OBE</p>
---	---

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

	<p>NIDN : 0516127501 NIPM : 19751216 200103 011 0880702 Jabatan Fungsional: Lektor Kepala Bidang Keahlian : Sistem Cerdas Email : sri.winiarti[at]tif.uad.ac.id Penyusun dan editor Bab 1 dan 2 Petunjuk Praktikum Basisdata Edisi OBE</p>
---	---

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

	<p>NIDN : 0522018302 NIPM : 19830122 201104 011 1100819 Jabatan Fungsional: Lektor Bidang Keahlian : Sistem Cerdas Email : annahendri[at]tif.uad.ac.id Penyusun dan editor Bab 1 dan 2 Petunjuk Praktikum Basisdata Edisi OBE</p>
---	--

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

	<p>NIDN : 0515069001 NIPM : 19900615 201908 011 1029280 Jabatan Fungsional: Asisten Ahli Bidang Keahlian : Sistem cerdas, bioinformatika, big data Email : miftahurrahma.rosyda[at]tif.uad.ac.id Penyusun dan editor Bab 1 dan 2 Petunjuk Praktikum Basisdata Edisi OBE</p>
---	--

KONTRIBUSI PENULIS

Nomor Bab	Daftar Penulis
Bab I	Jefree Fahana, S.T., M.Kom.
Bab II	Dewi Soyusiawaty, S.T., M.T
Bab III	Ir. Sri Winiarti, S.T., M.Cs.
Bab IV	Anna Hendri Soleliza Jones, S.Kom., M.Cs.
Bab V	Miftahurrahma Rosyda S.Kom., M.Eng.
Bab VI	Jefree Fahana, S.T., M.Kom., Miftahurrahma Rosyda S.Kom., M.Eng.
Bab VII	Dewi Soyusiawaty, S.T., M.T, Anna Hendri Soleliza Jones, S.Kom., M.Cs.
Bab VIII	Dewi Soyusiawaty, S.T., M.T
Bab IX	Ir. Sri Winiarti, S.T., M.Cs.
Bab X	Miftahurrahma Rosyda S.Kom., M.Eng.
Bab XI	Jefree Fahana, S.T., M.Kom.
Bab XII	Anna Hendri Soleliza Jones, S.Kom., M.Cs.

HALAMAN REVISI

Yang bertanda tangan di bawah ini:

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

NIPM : 19840528 201606 111 1011850

Jabatan : Dosen Pengampu Mata Kuliah Basisdata

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

No	Keterangan Revisi	Tanggal Revisi	Nomor Modul
1	a. Tuliskan revisi disini. b. dst.		PP/018/V/R1
2	a. Penataan ulang struktur praktikum b. Menambahkan materi Trigger dan Keamanan Data c. Menambahkan materi trend basisdata (NO SQL)	10 Juli 2023	PP/018/V/R2

Yogyakarta, 10 Juli 2023

Penyusun

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

NIPM. 19840528 201606 111 1011850

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Murein Miksa Mardhia S.T., M.T.
NIPM : 19891019 201606 011 1236278
Jabatan : Kepala Laboratorium S1 Informatika

Menerangkan dengan sesungguhnya bahwa Petunjuk Praktikum ini telah direview dan akan digunakan untuk pelaksanaan praktikum di Semester Gasal Tahun Akademik 2023/2024 di Laboratorium Praktikum S1 Informatika, Program Studi Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan.

Yogyakarta, 10 Juli 2023

Mengetahui,
Ketua Kelompok Keilmuan

Nama Ketua KK
NIP/NIY.

Kepala Laboratorium Praktikum
S1 Informatika



Murein Miksa Mardhia S.T., M.T.
NIPM. 19891019 201606 011 1236278

VISI DAN MISI PRODI INFORMATIKA

VISI

Menjadi program studi yang unggul dan inovatif dalam bidang rekayasa perangkat lunak dan sistem cerdas dengan dijiwai nilai-nilai Islam

MISI

1. Mengimplementasikan nilai-nilai AIK pada semua aspek kegiatan.
2. Memajukan ilmu pengetahuan dan teknologi Rekayasa Perangkat Lunak dan Sistem cerdas melalui pendidikan, penelitian, dan pengabdian kepada masyarakat.
3. Mengembangkan kerjasama dalam pendidikan, penelitian, dan pengabdian kepada masyarakat di tingkat lokal, nasional, maupun internasional.
4. Menyelenggarakan tata kelola program studi yang unggul dan inovatif.
5. Berperan aktif dalam kegiatan yang menunjang profesi dosen.

TATA TERTIB LABORATORIUM S1 INFORMATIKA

DOSEN/KOORDINATOR PRAKTIKUM

1. Dosen harus hadir saat praktikum minimal 15 menit di awal kegiatan praktikum untuk mengisi materi 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 yang akan mengikuti inhal diwajibkan mendaftarkan diri dan membayar administrasi inhal kepada laboran inhal paling lambat H-1 jadwal inhal.
4. 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.
5. 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).
6. Praktikan dilarang meninggalkan kegiatan praktikum tanpa seizin Asisten atau Laboran.
7. Praktikan harus meletakkan sepatu dan tas pada rak/loker yang telah disediakan.
8. Selama praktikum dilarang NGENET/NGE-GAME, kecuali mata praktikum yang membutuhkan atau menggunakan fasilitas Internet.
9. Praktikan dilarang melepas kabel jaringan atau kabel power praktikum tanpa sepengertahan laboran
10. Praktikan harus memiliki FILE Petunjuk praktikum dan digunakan pada saat praktikum dan harus siap sebelum praktikum berlangsung.
11. Praktikan dilarang melakukan kecurangan seperti mencontek atau menyalin pekerjaan praktikan yang lain saat praktikum berlangsung atau post-test yang menjadi tugas praktikum.

12. 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.
13. 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.
14. 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 mencatat dan merekap praktikan dengan status INHAL setiap minggu serta wajib mengumumkan mekanisme INHAL di awal pertemuan praktikum.
7. Asisten harus dapat bertindak secara profesional sebagai seorang asisten praktikum dan dapat menjadi teladan bagi praktikan.
8. 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.
9. Asisten tidak diperkenankan mengobrol sendiri apalagi sampai membuat gaduh.
10. Asisten dimohon mengkoordinasikan untuk meminta praktikan agar mematikan komputer untuk jadwal terakhir dan sudah dilakukan penilaian terhadap hasil kerja praktikan.
11. Asisten wajib untuk mematikan LCD Projector dan komputer asisten/praktikan apabila tidak digunakan.
12. 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. Pelaksanaan (waktu dan metode) INHAL sama seperti praktikum mingguan/reguler.

4. Izin praktikum mengikuti aturan izin SIMERU/KULIAH.
5. Yang tidak berkepentingan dengan praktikum dilarang mengganggu praktikan atau membuat keributan/kegaduhan.
6. Penggunaan lab diluar jam praktikum maksimal sampai pukul 21.00 dengan menunjukkan surat ijin dari Kepala Laboratorium Prodi Informatika.

Yogyakarta, 10 Juli 2023

Kepala Laboratorium Praktikum
S1 Informatika



Murein Miksa Mardhia S.T., M.T.

NIPM. 19891019 201606 011 1236278

DAFTAR ISI

HAK CIPTA	1
KATA PENGANTAR.....	2
DAFTAR PENYUSUN.....	3
KONTRIBUSI PENULIS	4
HALAMAN REVISI.....	5
HALAMAN PERNYATAAN.....	6
VISI DAN MISI PRODI INFORMATIKA.....	7
TATA TERTIB LABORATORIUM S1 INFORMATIKA	8
DAFTAR ISI	11
DAFTAR GAMBAR.....	12
DAFTAR TABEL.....	13
SKENARIO PRAKTIKUM SECARA DARING	14
PRAKTIKUM 1: PEMAHAMAN KASUS DENGAN IDENTIFIKASI ENTITAS	15
PRAKTIKUM 2: RELASI DAN PENGOLAHAN DATA	0
PRAKTIKUM 3: RELASI DALAM MODEL DATA	15
PRAKTIKUM 4: KONVERSI MODEL DATA KE TABEL.....	24
PRAKTIKUM 5: IMPLEMENTASI RELASI DAN KONVERSI TABEL.....	31
PRAKTIKUM 6: NORMALISASI	41
PRAKTIKUM 7: DDL.....	50
PRAKTIKUM 8: DML DAN FUNGSI AGREGASI.....	59
PRAKTIKUM 9: RELASI TABEL DENGAN JOIN.....	68
PRAKTIKUM 10: SUB QUERY, FUNCTION, STORE PROCEDURE	75
PRAKTIKUM 11: TRIGGER DAN KEAMANAN DATA	82
PRAKTIKUM 12: TREND DBMS	89
DAFTAR PUSTAKA.....	90

DAFTAR GAMBAR

Gambar 1.1 Label Gambar.

14

DAFTAR TABEL

SKENARIO PRAKTIKUM SECARA DARING

Nama Mata Praktikum : _____

Jumlah Pertemuan : _____

TABEL SKENARIO PRAKTIKUM DARING

Pertemuan ke	Judul Materi	Waktu (Lama praktikum sampai pengumpulan posttest)	Skenario Praktikum <i>(Dari pemberian pre-test, post-test dan pengumpulannya serta mencantumkan metode yang digunakan misal video, whatsapp group, Google meet atau lainnya)</i>
1			
2			
3	dst	dst	Dst

PRAKTIKUM 1: PEMAHAMAN KASUS DENGAN IDENTIFIKASI ENTITAS

Pertemuan ke : 1

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-05	Mampu mengkaji/menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-03	Mahasiswa dapat mengimplementasikan pemodelan data

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu mengimplementasikan entitas, atribut, kunci, relasi dan derajat relasi.

1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-05	CPMK-03	Mahasiswa memahami tabel, tipe data dan <i>primary key</i> dalam <i>Database Management System</i> .
CPL-05	CPMK-03	Mahasiswa mengenal dan dapat mengimplementasikan MySQL sebagai <i>Database Management System</i> .
CPL-07	CPMK-03	Mahasiswa mampu mengimplementasikan entitas, atribut, kunci, relasi dan derajat relasi

1.3. TEORI PENDUKUNG

1. Tabel dan Atribut

Entitas merupakan individu atau objek yang memiliki sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Entitas biasanya dalam bentuk kata benda (Oracle Academy, 2019). Entitas dapat diklasifikasikan menjadi tiga tipe, yaitu

Tabel 1.1 Tipe Entitas

Nama Tipe	Penjelasan	Contoh
Prime	Entitas Independen	Customer, Mahasiswa
Characteristic	Entitas yang muncul karena berelasi dengan entitas (<i>prime</i>) lainnya.	Pemesanan (order), Jadwal
Intersection	Entitas yang muncul karena ada dua atau lebih entitas	Item Pemesanan (order item)

Sedangkan untuk atribut adalah karakteristik atau ciri yang menjelaskan entitas sehingga dapat membedakan antara entitas satu dengan entitas lainnya (Oracle Academy, 2019). Dalam sebuah entitas pasti memiliki satu atau lebih atribut untuk mendefinisikan karakteristik dari entitas tersebut. Contoh atribut dari Entitas Mahasiswa yaitu NIM, Nama, JK.

2. Mengenal MySQL

MySQL adalah program *database server* yang mampu menerima dan mengirimkan datanya sangat cepat, multi user serta menggunakan perintah dasar SQL (Structured Query Language) (<http://mysql.com>). MySQL memiliki 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 :

- 1) MySQL merupakan Database Management System (DBMS).
- 2) MySQL sebagai Relational Database Management System (RDBMS) atau disebut dengan database Relational.
- 3) MySQL Merupakan sebuah database server yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.
- 4) MySQL merupakan sebuah database client.
- 5) MySQL mampu menerima query yang bertumpuk dalam satu permintaan atau Multi Threading.
- 6) MySQL merupakan Database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran GigaByte sekalipun.
- 7) MySQL didukung oleh driver ODBC, artinya database MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Visual Basic, PHP dan Delphi.

- 8) MySQL merupakan Database Server yang multi user, artinya database ini tidak hanya digunakan oleh satu pihak orang akan tetapi dapat digunakan oleh banyak pengguna.
- 9) MySQL mendukung field yang dijadikan sebagai kunci primer dan kunci unik (Unique).
- 10) MySQL memiliki kecepatan dalam pembuatan tabel maupun update table.

3. Pemahaman Kasus pada Sistem Informasi Perpustakaan

Saat meminjam buku di perpustakaan, 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 :

Tabel 1.2 Data disimpan pada Basis Data

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 pembayaran denda, misal tgl denda, kode buku, jml buku yang kena denda, nomor peminjam, status lunas.

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

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

1.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Tentukan entitas yang terlibat dalam studi kasus yang diberikan	25
2.	CPL-07	CPMK-03	Tentukan atribut yang terlibat dalam studi kasus yang diberikan	25
3.	CPL-07	CPMK-03	Tentukan relasi yang terlibat dalam studi kasus yang diberikan	25
4.	CPL-07	CPMK-03	Gambarkan model datanya	25

1.6. LANGKAH PRAKTIKUM

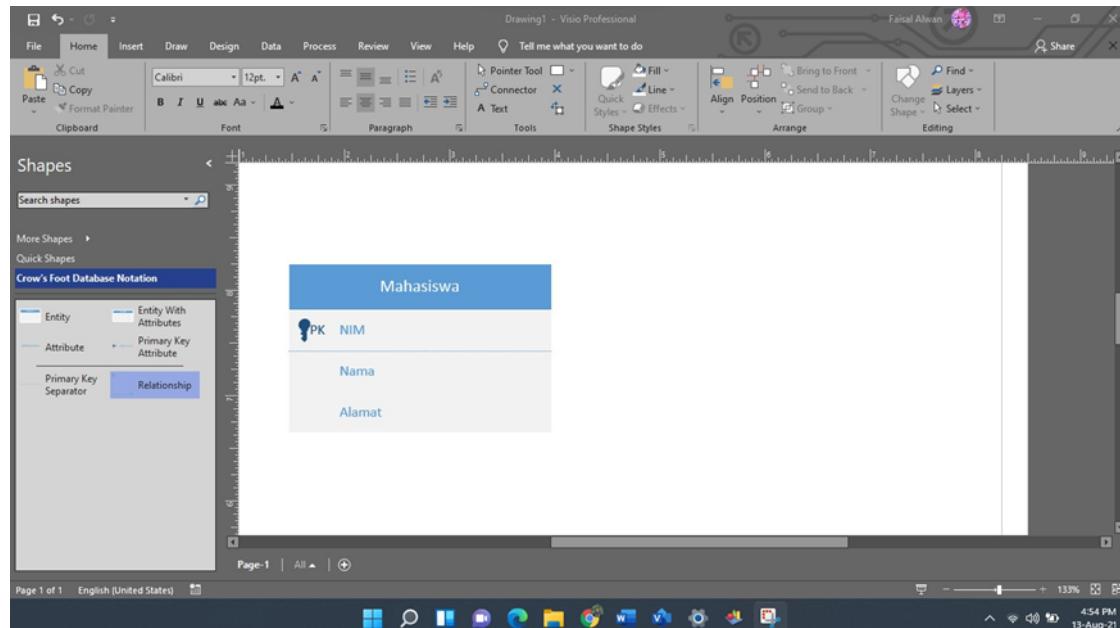
Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-07	CPMK-03	Buat database	Hasil praktikum	20
2.	CPL-07	CPMK-03	Buat tabel dilengkapi dengan atribut, tipe data, primary key	Hasil praktikum	80

Langkah-Langkah Praktikum:

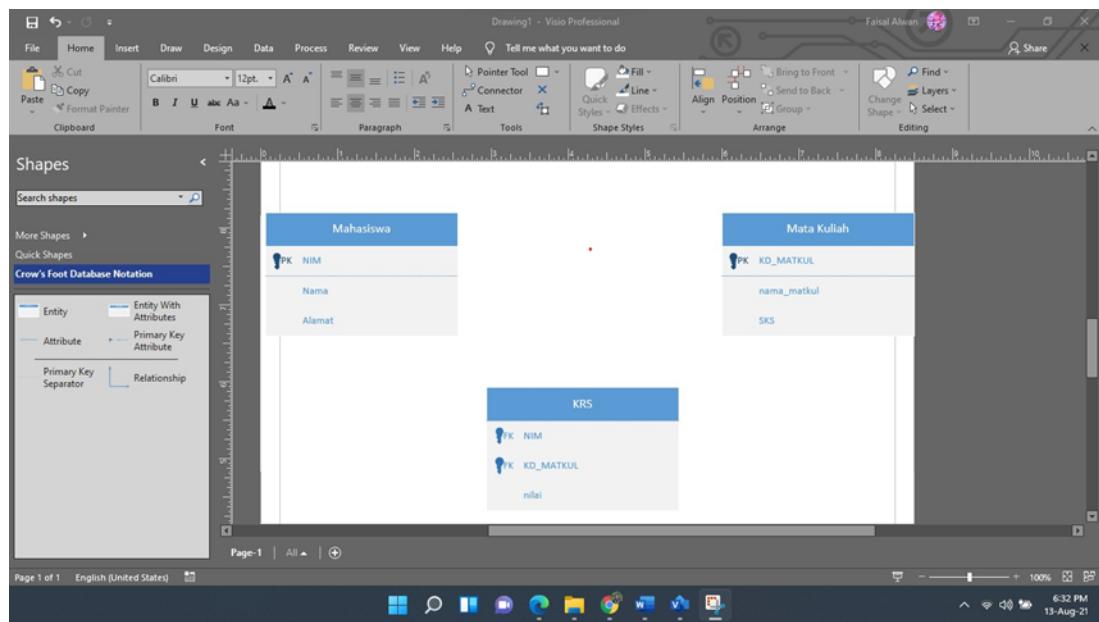
Buatlah pemodelan data menggunakan Microsoft Visio dan Powerdesigner dengan studi kasus “Mahasiswa mengambil mata kuliah”.

1. Menggunakan Visio



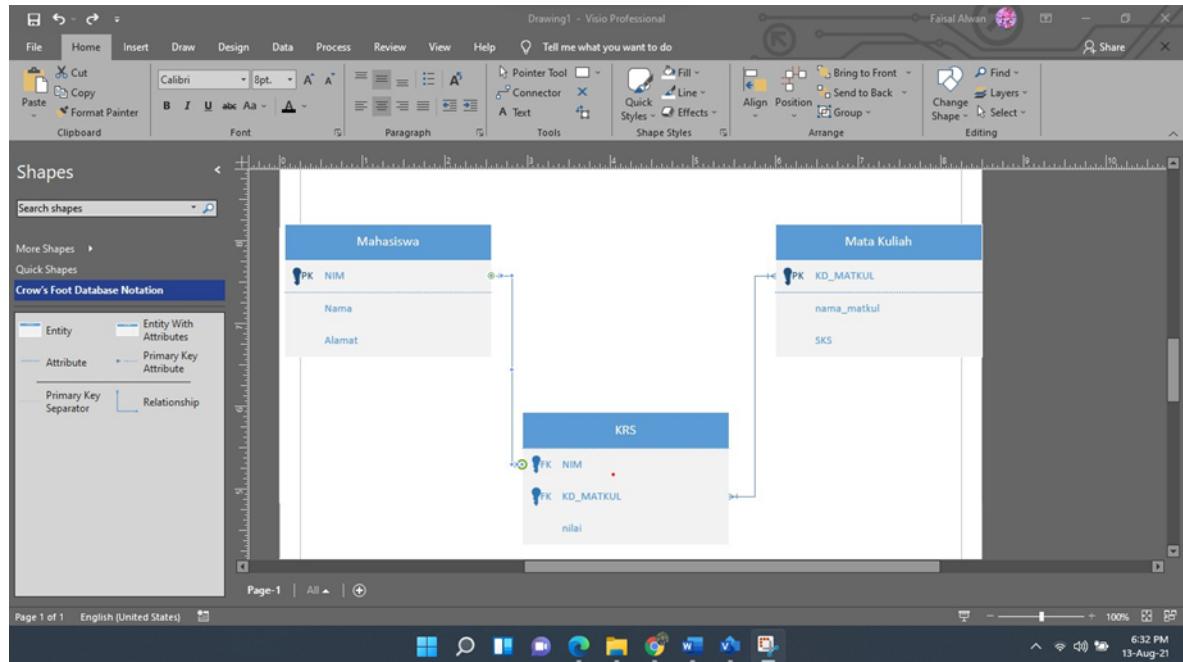
Gambar 1.1 Tampilan Awal Visio

Seperti yang telah dipraktikkan sebelumnya, buat entitas dari studi kasus di atas. Disini entitas awal yang terbentuk adalah entitas mahasiswa dan mata kuliah.



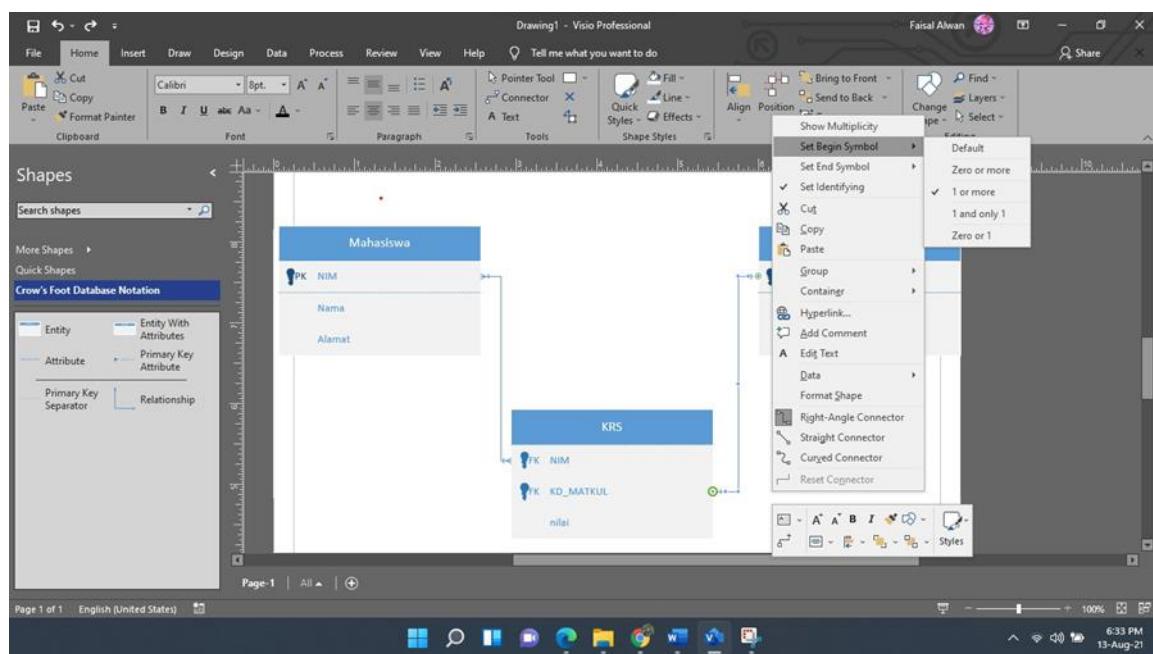
Gambar 1.2 Membuat Entitas

Karena relasi yang terbentuk adalah “many to many” maka akan dibuat 1 entitas baru yaitu entitas “SKS” yang berisi primary key dari masing – masing entitas, primary key tersebut dijadikan sebagai foreign key pada entitas baru untuk menghubungkan kedua entitas yang telah dibuat.



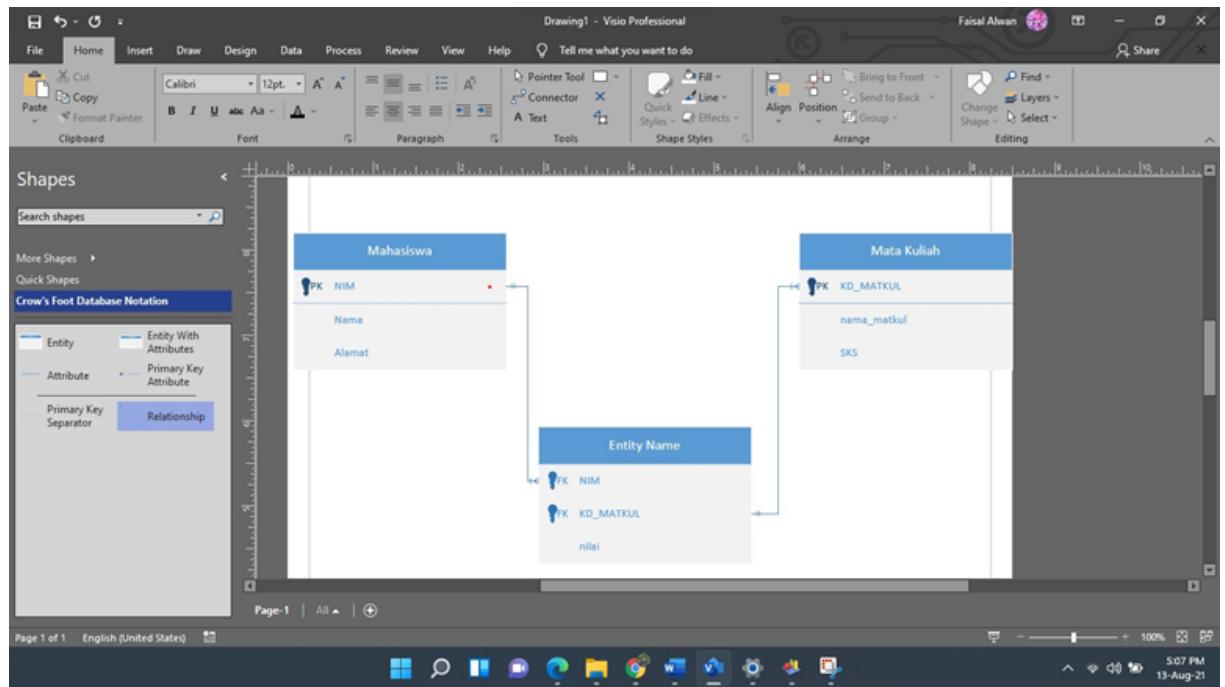
Gambar 1. 3 Relasi Pada Visio

Hubungkan kedua entitas yang telah dibuat dengan menggunakan shape “Relationship”.



Gambar 1. 4 Menghubungkan Entitas

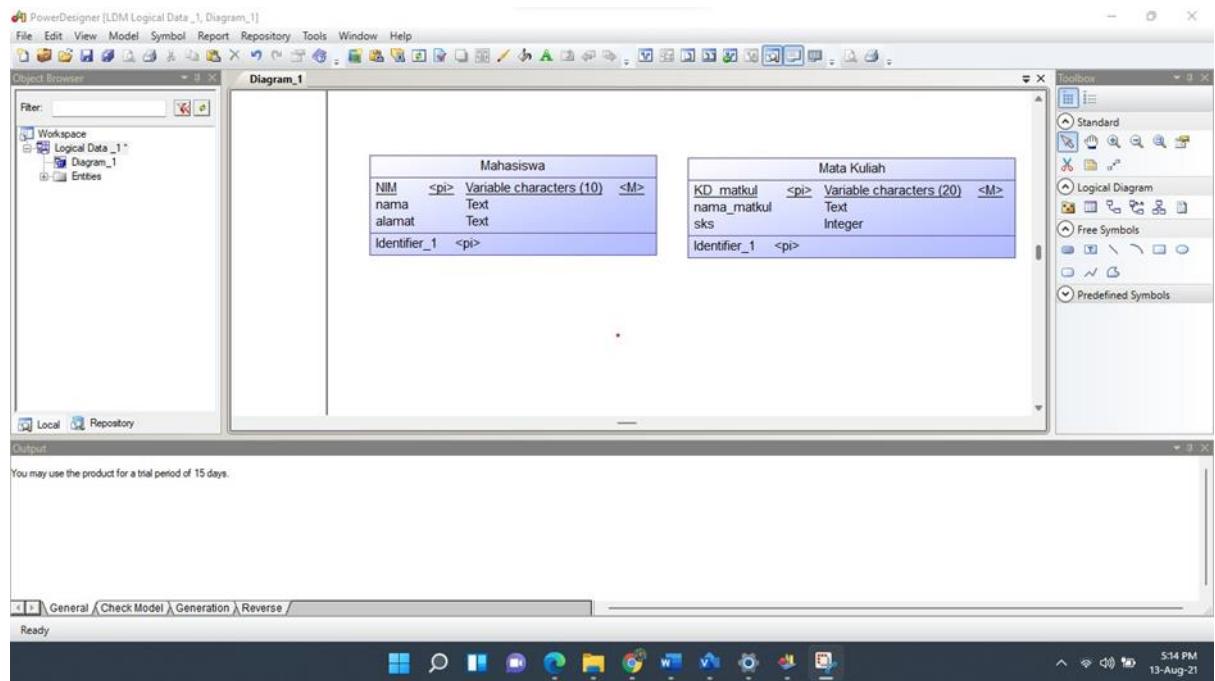
Ubah kardinalitas pada relasi yang telah dibuat dengan cara klik kanan pada relasi lalu pilih jenis kardinalitasnya.



Gambar 1. 5 Hasil Pemodelan

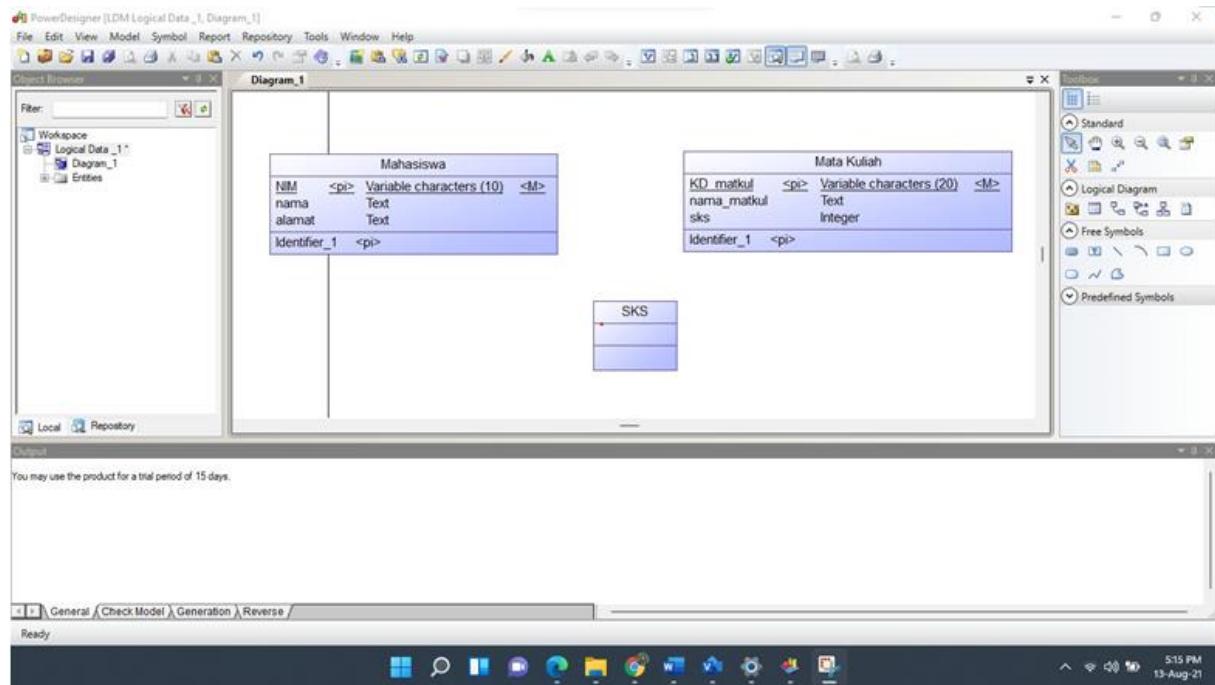
Di atas adalah hasil dari pemodelan data dari studi kasus yang ada.

2. Power Designer



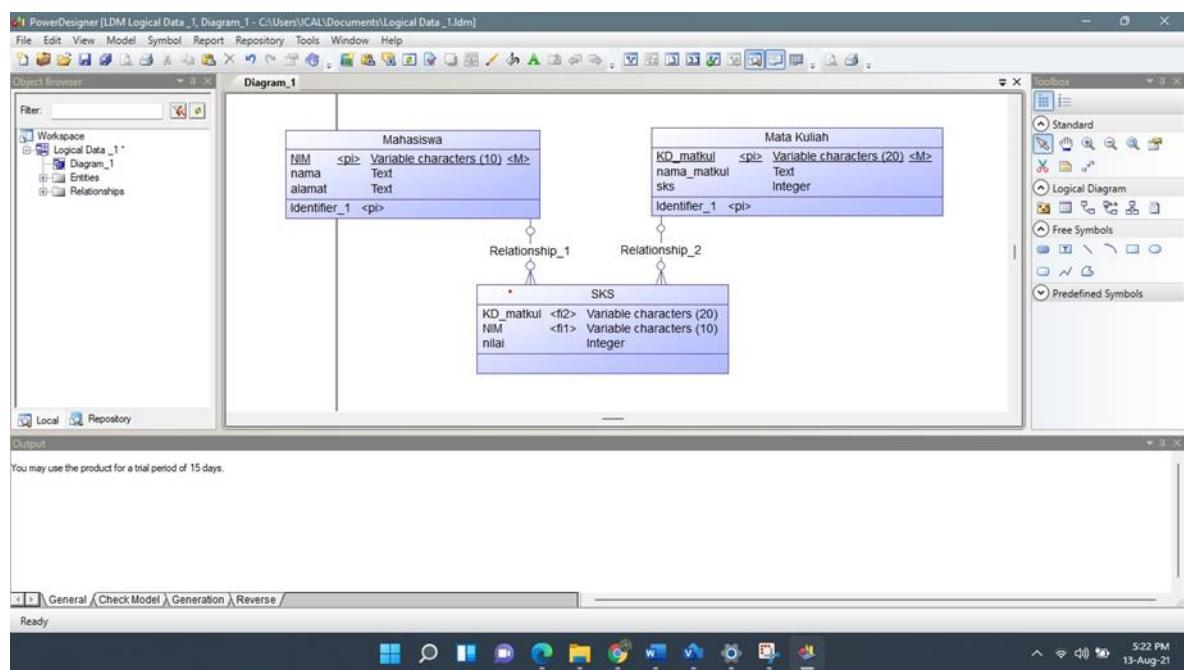
Gambar 1. 6 Tampilan Awal Power Designer

Buat entitas beserta atributnya berdasarkan studi kasus, entitas awal yang terbentuk adalah “mahasiswa” dan “mata kuliah”.



Gambar 1. 7 Buat Entitas dan Attribut

Karena relasi yang terbentuk adalah “many to many” maka akan dibuat 1 entitas baru yaitu entitas “SKS” yang nantinya berisi primary key dari masing – masing entitas, primary key tersebut dijadikan sebagai foreign key pada entitas baru untuk menghubungkan kedua entitas yang telah dibuat.

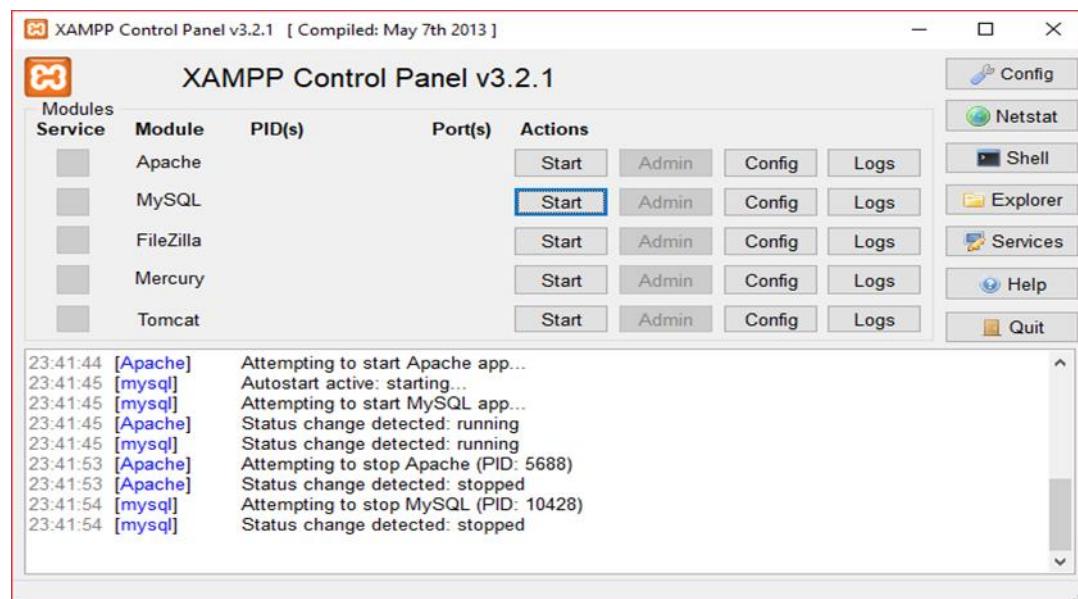


Gambar 1. 8 Primary Key dan Foreign Key

Gunakan logical diagram “Relationship” untuk menghubungkan kedua entitas dengan entitas yang baru terbentuk, lalu ubah relationship tersebut sesuai dengan keadaan pada studi kasus. Hasil akhir dari pemodelan data menggunakan Power Designer adalah seperti di atas.

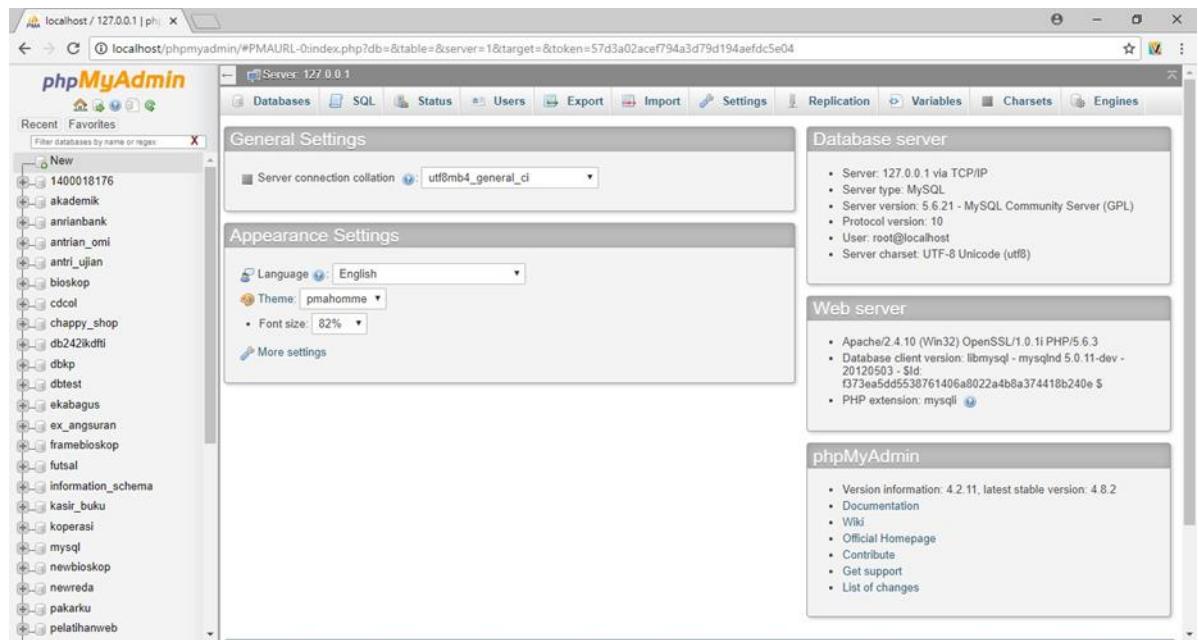
3. Mengakses PHPMyAdmin dengan XAMPP

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 1. 9 Tampilan XAMPP

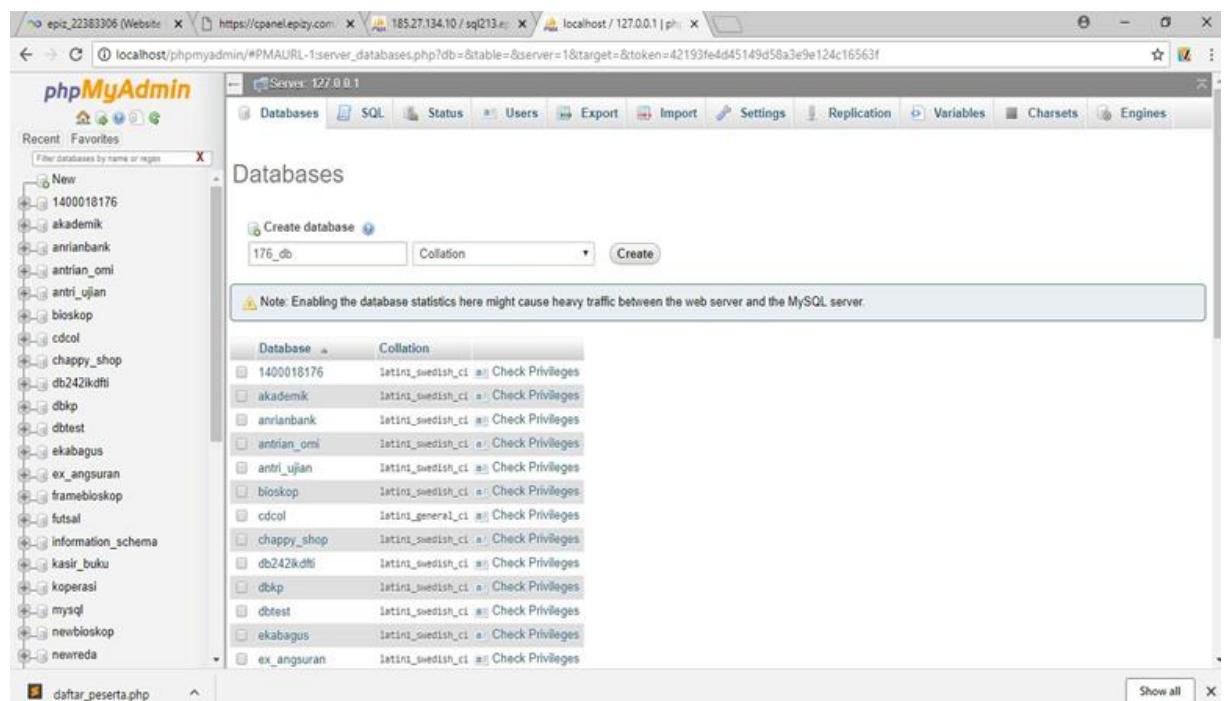
Kemudian membuka browser (Chrome, Mozilla, Opera, dll) dan mengaksesnya dengan mengetikkan “localhost/phpmyadmin” pada kolom isian URL, sehingga muncul seperti pada gambar di bawah ini.



Gambar 1. 10 Tampilan PHPMyAdmin

Membuat Database

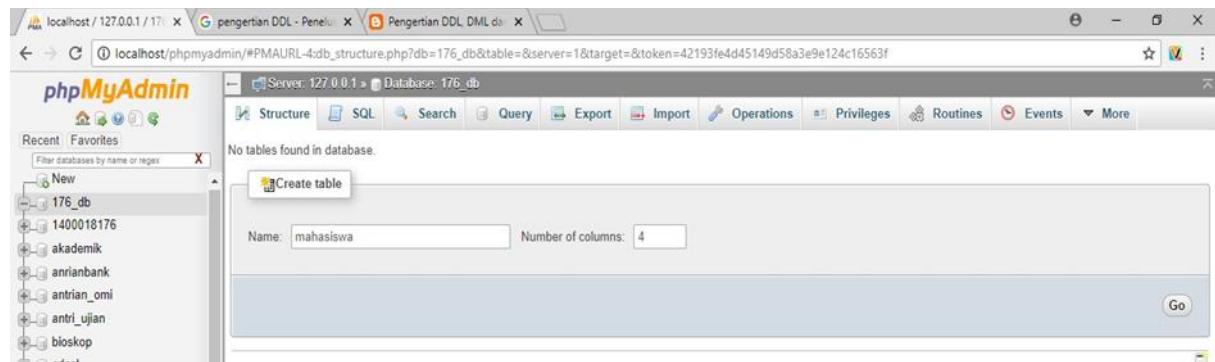
Pilih menu “new” kemudian isi nama database pada kolom yang sudah disediakan, untuk keseragaman nama database diisi dengan “3_digit_nim_terakhir_db” kemudian pilih “create”.



Gambar 1. 11 Create DB

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 1. 12 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.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A	Com
nim	VARCHAR	10	None						
nama	VARCHAR	50	None						
prodi	VARCHAR	50	None						
alamat	TEXT		None						

Gambar 1. 13 Isi Data Table

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

The screenshot shows the MySQL Workbench 'Structure' tab for a table named 'mahasiswa'. The table contains four columns: 'nim' (VARCHAR, length 10, primary key), 'nama' (VARCHAR, length 50), 'prodi' (VARCHAR, length 50), and 'alamat' (TEXT). The 'nim' column is highlighted in yellow and has a 'PRIMARY' key icon next to it. The 'Storage Engine' is set to InnoDB and the 'Collation' is set to latin1_swedish_ci.

Gambar 1. 14 Primary Key

1.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Nyatakan model data dari kasus yang diberikan	30
2.	CPL-07	CPMK-03	Implementasikan entitas dan atribut yang sudah ditentukan ke dalam Phpmyadminnya dengan ketentuan isian datanya	70

1.8. HASIL CAPAIAN PRAKTIKUM

Ditulis oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-07	CPMK-03	20%		
2.	Praktik	CPL-07	CPMK-03	30%		
3.	Post-Test	CPL-07	CPMK-03	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

PRAKTIKUM 2: RELASI DAN PENGOLAHAN DATA

Pertemuan ke : 2

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu.
CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah.
CPMK-03	Mahasiswa dapat mengimplementasikan pemodelan data.

2.1 DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu mengimplementasikan basis data dan mengolah data.

2.2 INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

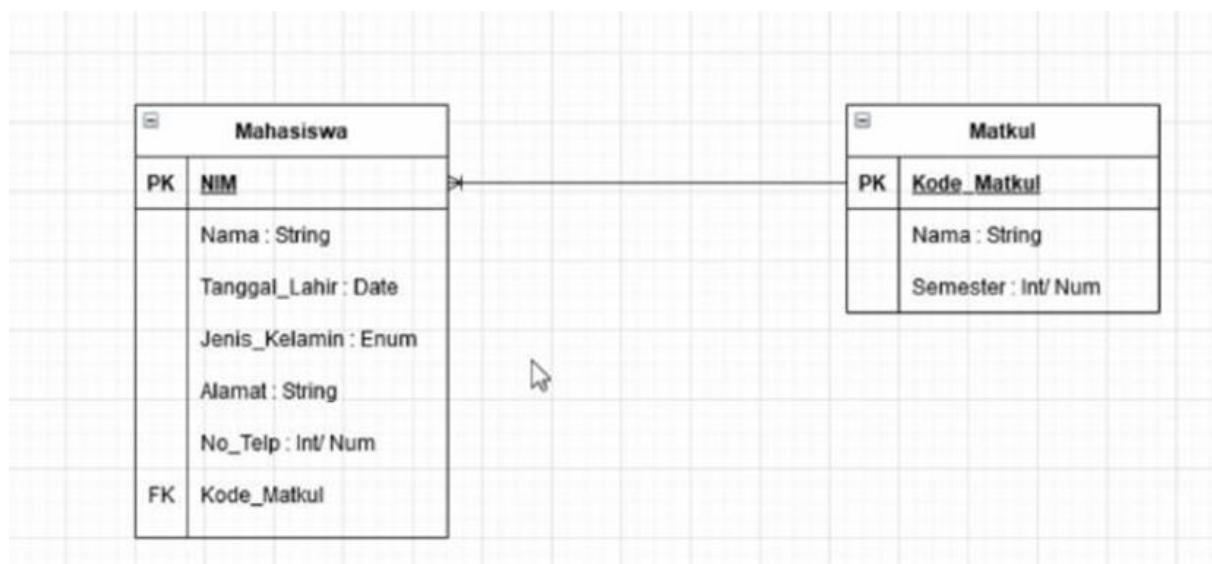
CPL-05	CPMK-03	Kemampuan mahasiswa dalam mengimplementasikan basis data dan mengolah data.
CPL-07		

2.3 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*

Berikut contoh relasi antar entitas mahasiswa dan mata kuliah.



Gambar 2.1 Gambaran Relasi

2.4 HARDWARE DAN SOFTWARE

Hardware dan *software* yang digunakan dalam praktikum ini yaitu:

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

2.5 PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-05	CPMK-03	Jelaskan operasi dasar dalam basis data	30
2.	CPL-05	CPMK-03	Tuliskan sintax dari Insert, Update dan delete.	35
3.	CPL-05	CPMK-03	Tuliskan sintax untuk memasukkan data sebagai berikut : (Nur Fatayah, 2000018451, 085338595804, nur2000018451@webmail.uad.ac.id)	35

2.6 LANGKAH PRAKTIKUM

Aturan Penilaian (**Total Skor: 100**):

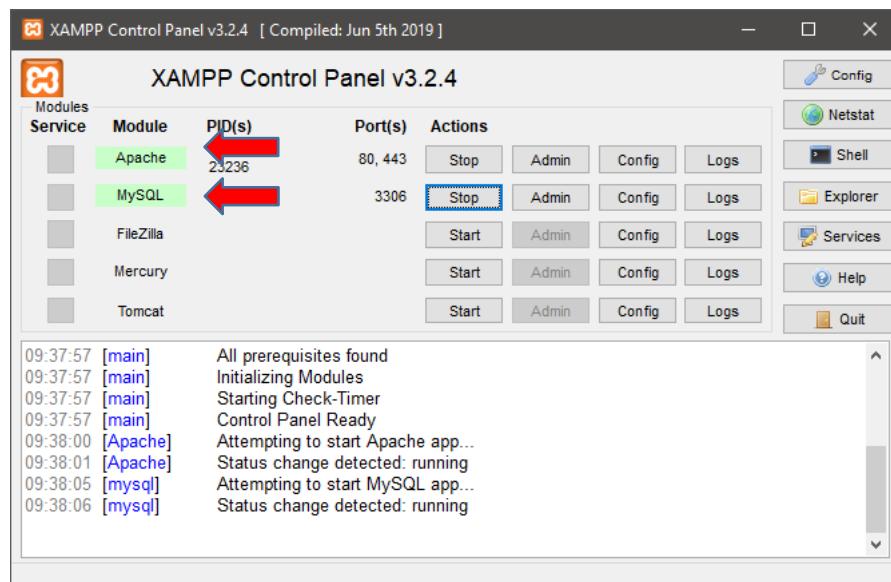
No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-05	CPMK-03	Lakukan proses insert data dari kasus	Hasil praktikum	35
2.	CPL-05	CPMK-03	Lakukan proses update data dari kasus	Hasil praktikum	35
3.	CPL-05	CPMK-03	Lakukan proses delete data dari kasus	Hasil praktikum	30

Langkah-Langkah Praktikum:

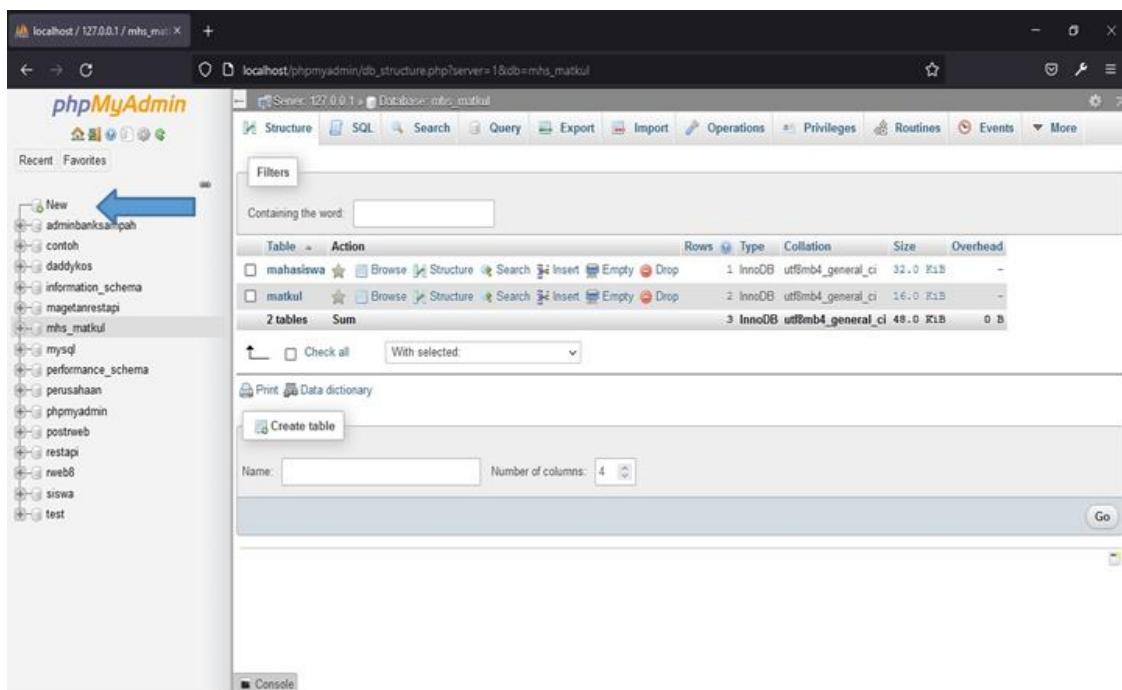
1. Pembuatan Relasi pada Mysql. Studi kasus Mahasiswa

Langkah- Langkah :

- Langkah pertama yaitu melakukan pembuatan database baru pada Mysql. Aktifkan terlebih dahulu XAMPP => Centang pada MySql => Buka <http://localhost/phpmyadmin/>

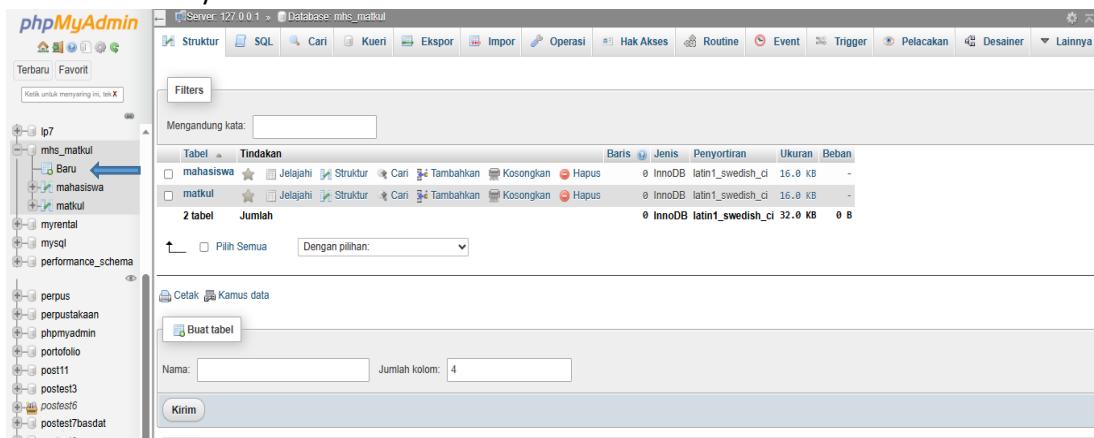


Gambar 2.2 Tampilan XAMPP

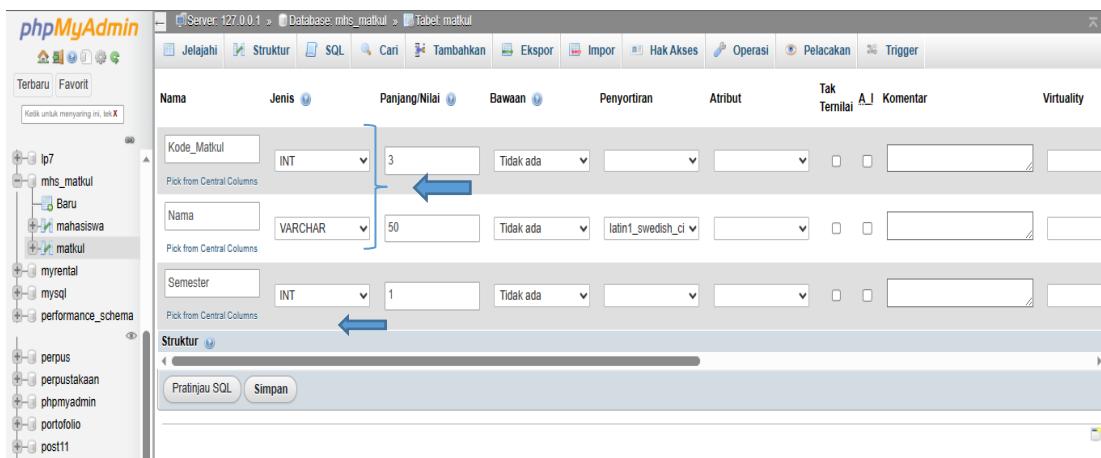


Gambar 2.3 Tampilan PHP MyAdmin

- b. Langkah selanjutnya yaitu membuat tabel. Buatlah tabel sesuai dengan rancang bangun sebelumnya.

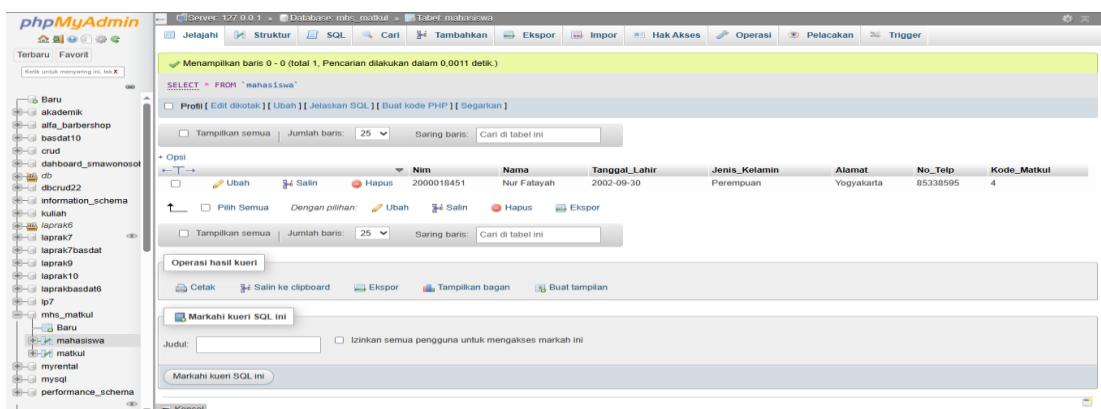


Gambar 2.4 Membuat Tabel



Gambar 2.5 Mengisi atribut

- c. Setelah semua tabel terbuat. Langkah selanjutnya yaitu melakukan relasi antar tabel. Pilih tabel Mahasiswa yang sebelumnya telah dibuat, kemudian Klik pada Structure, lalu klik Relation View dan setting relasi tabel sesuai struktur awal kita tadi.



Gambar 2.6 Tampilan Browse Tabel Mahasiswa

Struktur tabel

#	Nama	Jenis	Penyortiran	Atribut	Tak Termilai	Bawaan	Komentar	Ekstra	Tindakan
1	Nim	int(20)			Tidak	Tidak ada		Ubah Hapus Lainnya	
2	Nama	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya	
3	Tanggal_Lahir	date			Tidak	Tidak ada		Ubah Hapus Lainnya	
4	Jenis_Kelamin	enum('Laki-Laki', 'Perempuan')	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya	
5	Alamat	text	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya	
6	No_Telp	int(15)			Tidak	Tidak ada		Ubah Hapus Lainnya	
7	Kode_Matkul	int(3)			Tidak	Tidak ada		Ubah Hapus Lainnya	

Cetak Usulkan struktur tabel Lacak tabel Normalisasi

Tambahkan 1 kolom setelah Kode_Matkul Kirim

Indeks

Tindakan	Nama kunci	Jenis	Unik	Dipadatkan	Kolom	Kardinalitas	Penyortiran	Tak Termilai	Komentar
Ubah Hapus	PRIMARY	BTREE	Ya	Tidak	Nim	1	A	Tidak	
Ubah Hapus	Kode_Matkul	BTREE	Ya	Tidak	Kode_Matkul	1	A	Tidak	

Create an index on 1 columns Kirim

Gambar 2.7 Tampilan Struktur Tabel Mahasiswa

Struktur tabel

Foreign key constraints

Actions	Constraint properties	Kolom	Foreign key constraint (INNODB)	Basis data	Tabel	Kolom
	Constraint name ON DELETE RESTRICT ON UPDATE RESTRICT	Kode_Matkul	+Tambah kolom	mhs_matkul	matkul	Kode_Matkul

Pilih kolom untuk ditampilkan: Nama

Pratinjau SQL Simpan

Isi Sesuai Relasi Struktur Awal

Gambar 2.8 Tampilan Relation View/Tampilan Hubungan

Struktur tabel

#	Nama	Jenis	Penyortiran	Atribut	Tak Termilai	Bawaan	Komentar	Ekstra	Tindakan
1	Nim	int(20)			Tidak	Tidak ada		Ubah Hapus Lainnya	
2	Nama	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya	
3	Tanggal_Lahir	date			Tidak	Tidak ada		Ubah Hapus Lainnya	
4	Jenis_Kelamin	enum('Laki-Laki', 'Perempuan')	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya	
5	Alamat	text	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya	
6	No_Telp	int(15)			Tidak	Tidak ada		Ubah Hapus Lainnya	
7	Kode_Matkul	int(3)			Tidak	Tidak ada		Ubah Hapus Lainnya	

Cetak Usulkan struktur tabel Lacak tabel Normalisasi

Tambahkan 1 kolom setelah Kode_Matkul Kirim

Indeks

Tindakan	Nama kunci	Jenis	Unik	Dipadatkan	Kolom	Kardinalitas	Penyortiran	Tak Termilai	Komentar
Ubah Hapus	PRIMARY	BTREE	Ya	Tidak	Nim	0	A	Tidak	
Ubah Hapus	Kode_Matkul	BTREE	Ya	Tidak	Kode_Matkul	0	A	Tidak	

Create an index on 1 columns Kirim

Gambar 2.9 Foreign Key ditambahkan

- d. Selanjutnya, isikan terlebih dahulu data yang ada di tabel Matakuliah. Setelah itu isilah data di tabel Mahasiswa.

The screenshot shows the 'matkul' table in the 'mts_matkul' database. The table has three columns: Kode_Matkul, Nama, and Semester. There are two rows of data: one for 'Basis Data' (Semester 3) and one for 'KDJK' (Semester 4). Below the table, there are buttons for 'Ubah' (Edit), 'Salin' (Copy), and 'Hapus' (Delete). The 'Tambahkan' button is highlighted with a blue arrow.

Gambar 2.10 Tampilan Browse Matkul

This screenshot shows the 'matkul' table insert form. Two new rows are being added:

- Row 1: Kode_Matkul: 3, Nama: Basis Data, Semester: 3
- Row 2: Kode_Matkul: 4, Nama: KDJK, Semester: 4

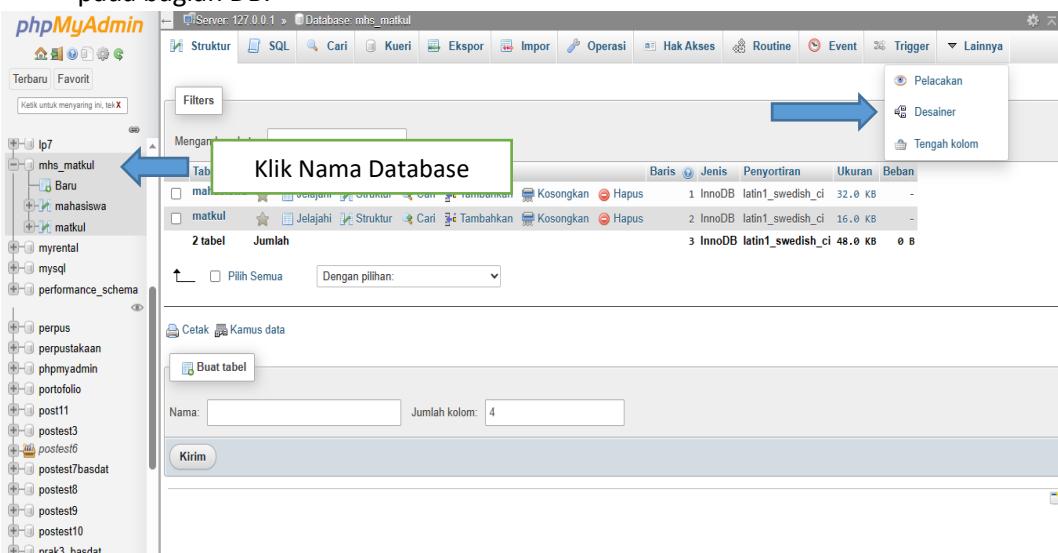
The 'Pratinjau' (Preview) button is highlighted with a blue arrow.

Gambar 2.11 Tampilan Insert Data Matkul

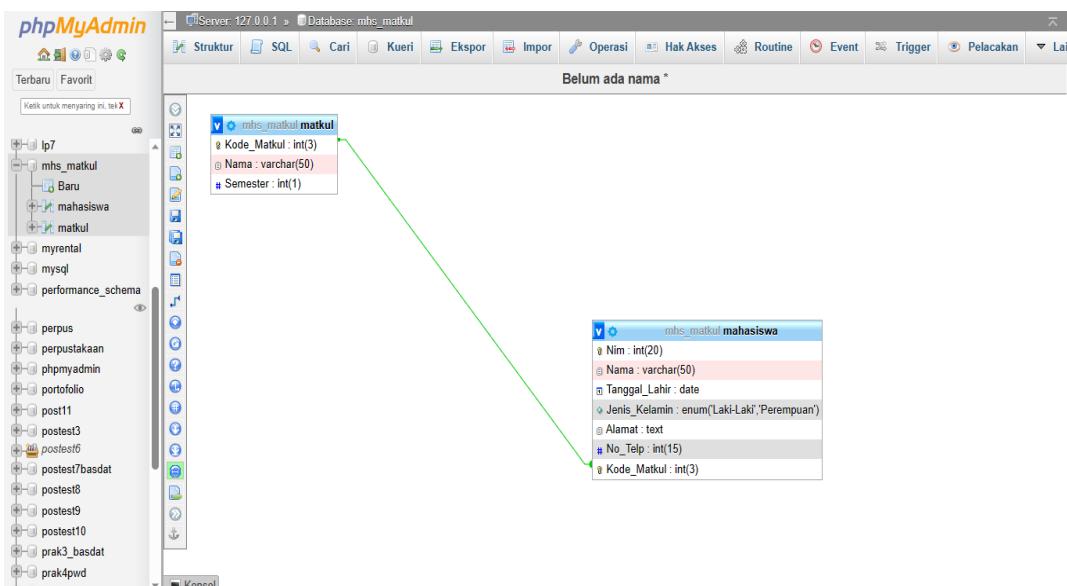
This screenshot shows the 'mahasiswa' table insert form. A dropdown menu for 'Jenis_Kelamin' is open, showing 'Perempuan'. The 'Kirim' (Send) button is highlighted with a blue arrow.

Gambar 2.12 Tampilan Insert Data Mahasiswa

- e. Maka anda dengan ini telah menyelesaikan Relasi antar table dan juga telah mengisi data pada setiap tabelnya. Untuk lebih melihat hasil relasi kita, bisa melihat di bagian Designer pada bagian DB.



Gambar 2.13 Tampilan Untuk Melihat Desainer



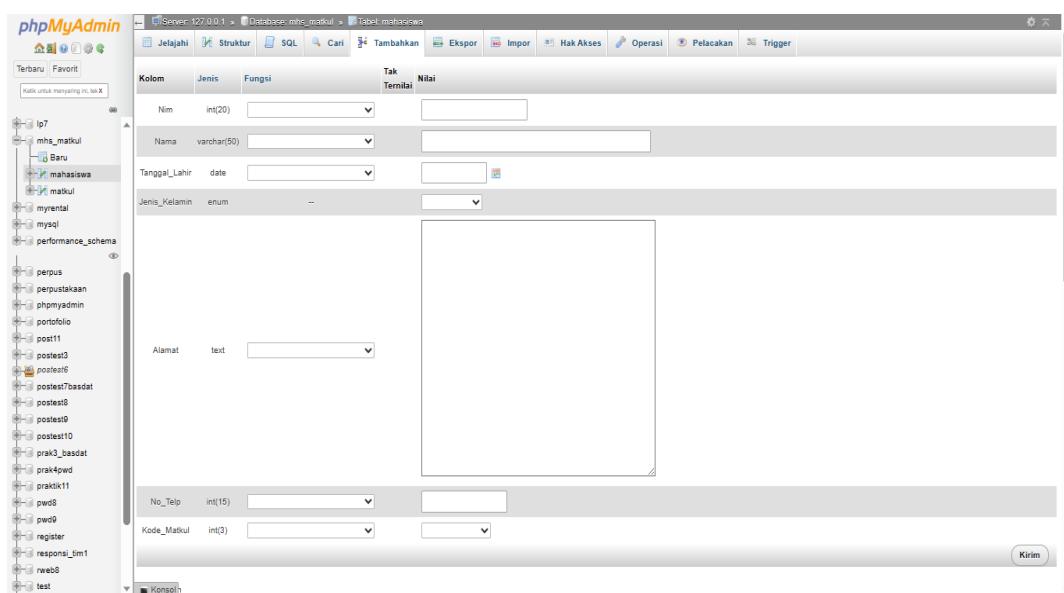
Gambar 2.14 Tampilan Designer Pada DB

2. Pengolahan Data

a. Insert Data

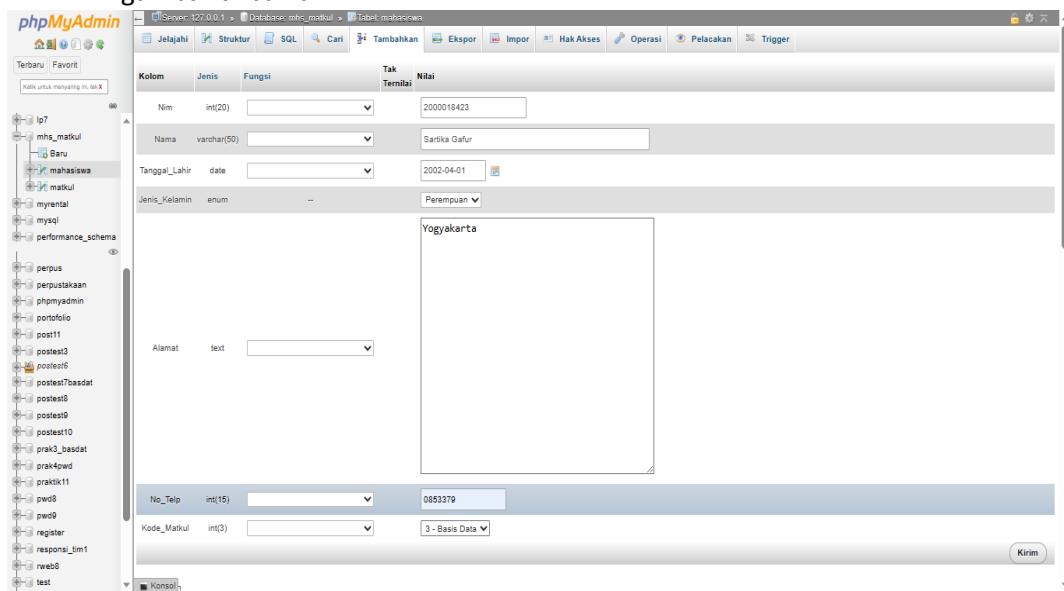
Pada langkah ini digunakan untuk menginputkan data ke dalam tabel “mahasiswa”, sebagai contoh akan diinputkan sebanyak 3 data. Langkah menambahkan data adalah sebagai berikut:

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



Gambar 2.15 Insert Data

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



Gambar 2.16 Isi Value Data

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

Nim	Nama	Tanggal_Lahir	Jenis_Kelamin	Alamat	No_Telp	Kode_Matkul
2000018423	Sartika Gefur	2002-04-01	Perempuan	Yogyakarta	853379	3
2000018451	Nur Falayah	2002-09-30	Perempuan	Yogyakarta	85338595	4

Gambar 2.17 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.

b. 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 :** 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.

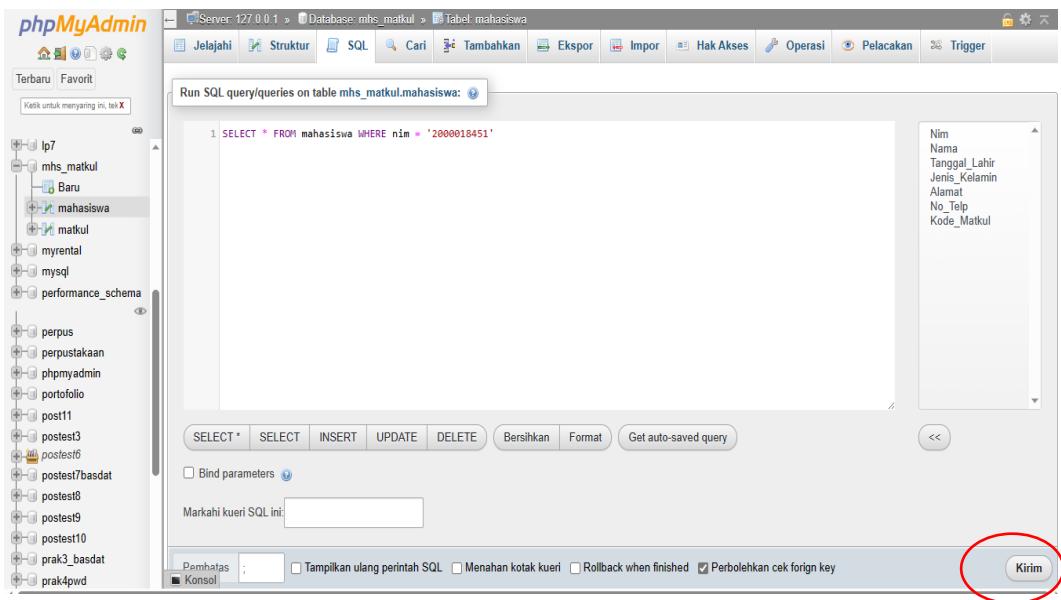
Langkah menampilkan data adalah sebagai berikut :

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

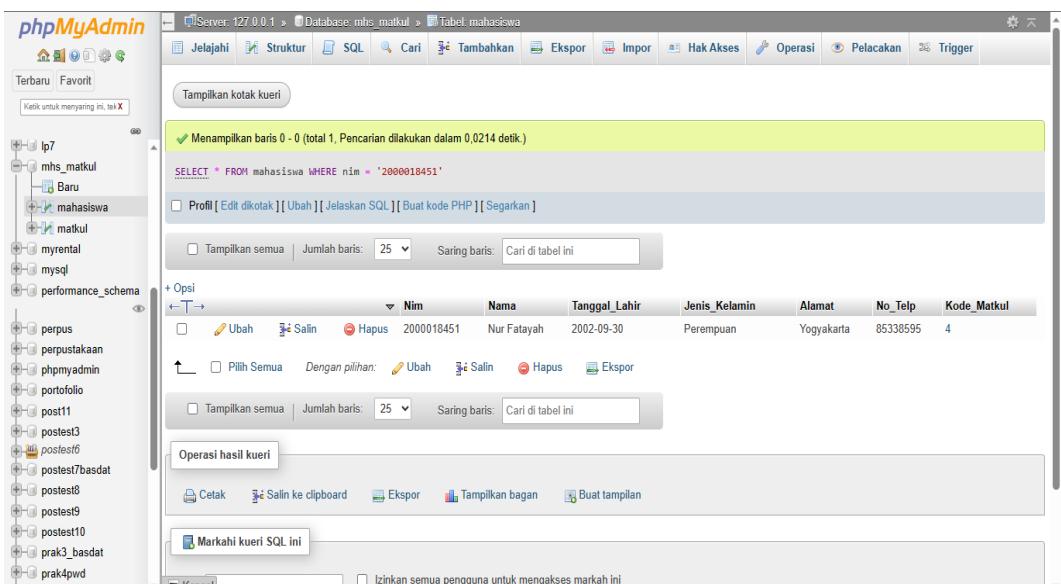
```
Run SQL query/queries on table mhs_makul.mahasiswa:
1. SELECT * FROM `mahasiswa` WHERE 1
```

Gambar 2.18 Perintah SQL

- b. Misalkan data yang akan ditampilkan yaitu mahasiswa dengan nim 2000018451. Maka query yang dibutuhkan yaitu `SELECT * FROM mahasiswa WHERE nim = '2000018451'`. Kemudian pilih "Go"/ "kirim", sehingga tampak seperti pada gambar di bawah ini.



Gambar 2.19 Tampil Data



Gambar 2.20 Lihat Data

c. 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.

	Nim	Nama	Tanggal_Lahir	Jenis_Kelamin	Alamat	No_Telp	Kode_Matkul
<input type="checkbox"/>	2000018423	Sartika Gafur	2002-04-01	Perempuan	Yogyakarta	853379	3
<input checked="" type="checkbox"/>	2000018451	Nur Fatayah	2002-09-30	Perempuan	Yogyakarta	85338595	4

Gambar 2.21 Ubah Data

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

Kolom	Jenis	Fungsi	Tak Termilai	Nilai
Nim	int(20)			2000018451
Nama	varchar(50)			Nur Fatayah
Tanggal_Lahir	date			2002-09-30
Jenis_Kelamin	enum			Perempuan
Alamat	text			Sumatra

Gambar 2.22 Edit Data

	<input type="checkbox"/> Ubah	<input type="checkbox"/> Salin	<input type="checkbox"/> Hapus	Nim	Nama	Tanggal_Lahir	Jenis_Kelamin	Alamat	No_Telp	Kode_Matkul
<input type="checkbox"/>	<input type="checkbox"/> Ubah	<input type="checkbox"/> Salin	<input type="checkbox"/> Hapus	2000018423	Sartika Gafur	2002-04-01	Perempuan	Yogyakarta	853379	3
<input checked="" type="checkbox"/>	<input type="checkbox"/> Ubah	<input type="checkbox"/> Salin	<input type="checkbox"/> Hapus	2000018451	Nur Fatayah	2002-09-30	Perempuan	Sumatra	85338595	4

Gambar 2.23 Hasil Edit Data

d. Delete Data

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

+ Opsi		Nim	Nama	Tanggal_Lahir	Jenis_Kelamin	Alamat	No_Telp	Kode_Matkul
<input type="checkbox"/>	Ubah	2000018423	Sartika Gafur	2002-04-01	Perempuan	Yogyakarta	853379	3
<input type="checkbox"/>	Ubah	2000018451	Nur Fatayah	2002-09-30	Perempuan	Sumatra	85338595	4

Gambar 2.24 Hapus Data

✓ 1 baris terpengaruh.

```
UPDATE `mahasiswa` SET `Alamat` = 'Sumatra' WHERE `mahasiswa`.`Nim` = 2000018451;
```

[Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP]

✓ Menampilkan baris 0 - 1 (total 2, Pencarian dilakukan dalam 0,0018 detik.)

SELECT * FROM `mahasiswa`

Konfirmasi

Anda yakin akan menjalankan "DELETE FROM `mahasiswa` WHERE `mahasiswa`.`Nim` = 2000018423"?

Oke Batal

Nim	Nama	Tanggal_Lahir	Jenis_Kelamin	Alamat	No_Telp	Kode_Matkul
2000018423	Sartika Gafur	2002-04-01	Perempuan	Yogyakarta	853379	3
2000018451	Nur Fatayah	2002-09-30	Perempuan	Sumatra	85338595	4

Gambar 2.25 Klik OK

✓ 1 baris terpengaruh.

```
UPDATE `mahasiswa` SET `Alamat` = 'Sumatra' WHERE `mahasiswa`.`Nim` = 2000018451;
```

[Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP]

✓ Menampilkan baris 0 - 1 (total 2, Pencarian dilakukan dalam 0,1133 detik.)

SELECT * FROM `mahasiswa`

1 baris terpengaruh. (Pencarian dilakukan dalam 0,1133 detik.)

[Edit dikotak] [Ubah] [Jelaskan SQL] [Segarkan]

Nim	Nama	Tanggal_Lahir	Jenis_Kelamin	Alamat	No_Telp	Kode_Matkul
2000018451	Nur Fatayah	2002-09-30	Perempuan	Sumatra	85338595	4

Gambar 2.26 Data Berhasil Dihapus

2.7 POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-07	CPMK-03	Implementasikan entitas dan atribut yang sudah ditentukan tadi ke dalam Phpmyadmindnya dengan ketentuan isian datanya minimal 10 data.	50
2.	CPL-07	CPMK-03	Screenshot perintah insert menggunakan syntax SQL (2 data dari masing masing table)	50

2.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-05	CPMK-03	20%		
2.	Praktik	CPL-05	CPMK-03	30%		
3.	Post-Test	CPL-07	CPMK-03	50%		
						Total Nilai

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

PRAKTIKUM 3: RELASI DALAM MODEL DATA

Pertemuan ke : 3

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-03	Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas
CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-01	Mahasiswa memahami konsep data dan informasi (ITE-IMA-02 ACM 2017)
CPMK-02	Mahasiswa memahami, menganalisa dan menyimpulkan arsitektur organisasi dan pemodelan data (ITE-IMA-03 ACM 2017, ITE-IMA-07 ACM 2017)
CPMK-03	Mahasiswa dapat mengimplementasikan pemodelan data

3.1 DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas dengan menggunakan asesmen praktik dan posttest
2. Berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya melalui asesmen Pre test, praktik dan post test
3. Mengkaji/menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu dengan asesmen Pre test, praktik dan post test

4. Memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah

3.2 INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-03 CPL-04	CPMK-01	1. Kemampuan mahasiswa meningkat dalam memahami konsep data dan informasi serta dapat berfikir logis yang diterapkan dengan cara memodelkan masalah terkait Relasi dengan menggunakan aplikasi atau tools desain model basis data yang tersedia.
CPL-03 CPL-04 CPL-05	CPMK-02	2. Kemampuan mahasiswa meningkat dalam memahami, menganalisa dan menyimpulkan arsitektur organisasi dan pemodelan data yang terkait dengan Relasi 3. Kemampuan mahasiswa meningkat dalam mengkaji/menganalisis permasalahan terkait Relasi dengan mempertimbangkan berbagai aspek 4. Kemampuan mahasiswa meningkat dalam memilih, membuat dan menerapkannya dengan menggunakan perangkat desain basis data yang tersedia untuk penyelesaian masalah terkait Relasi
CPL-05 CPL-07	CPMK-03	5. Kemampuan mahasiswa meningkat dalam menganalisa, memilih dan mengimplementasikan masalah terkait Relasi dalam pemodelan basis data dengan menggunakan tool pemodelan yang tersedia

3.3 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 Model) bukan merupakan empat komponen dalam konstruksi ER model.

Tabel 3.1 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. Kardinalitas dalam Relasi

Kardinalitas adalah mengukur kuantitas sesuatu, dalam hal ini adalah entitas. Dalam suatu relasi, itu menentukan sejauh mana satu entitas terkait dengan yang lain dengan menjawab pertanyaan, "Berapa banyak?"

Sebagai contoh:

- 1) Berapa banyak pekerjaan yang dapat dilakukan oleh satu karyawan? Satu pekerjaan saja? Atau lebih dari satu pekerjaan?
- 2) Berapa banyak karyawan yang dapat memegang satu pekerjaan tertentu? Satu karyawan saja? Atau lebih dari satu karyawan?

Catatan: Kardinalitas suatu relasi hanya menjawab apakah bilangan itu tunggal atau jamak;

Sekali lagi, organisasi yang berbeda mungkin memiliki aturan bisnis yang berbeda yang menentukan jenis relasi ini. Namun, secara umum, setiap karyawan akan memegang satu pekerjaan, dan setiap pekerjaan dapat dipegang oleh nol atau lebih karyawan.

c. 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 ini melahirkan beberapa teori atau metode antara lain ialah pemetakan ER to tabel 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, 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 multivalue tersebut. Tambahkan primary key dari relasi yang memiliki multivalue 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.

3.4 HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Buku Petunjuk Praktikum
3. Aplikasi pemodelan: visio drawing, edraw atau Draw io

3.5 PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03 CPL-04	CPMK-01	1. Untuk menentukan jenis relasi dalam suatu model data sangat bergantung dengan proses bisnis dalam suatu organisasi. Mengapa demikian..?	25
2.	CPL-03 CPL-04	CPMK-02	2. Jelaskan bagaimana peran relasi dalam menganalisa dan menyimpulkan arsitektur organisasi dan pemodelan data? 3. Jelaskan Aspek apa saja yang perlu dipertimbangkan dalam mengkaji/menganalisis permasalahan terkait Relasi? 4. Untuk memulai dalam menganalisa terkait masalah Relasi dalam pemodelan data bagaimana langkah-langkahnya..?	25 25 25
			Total Nilai	100

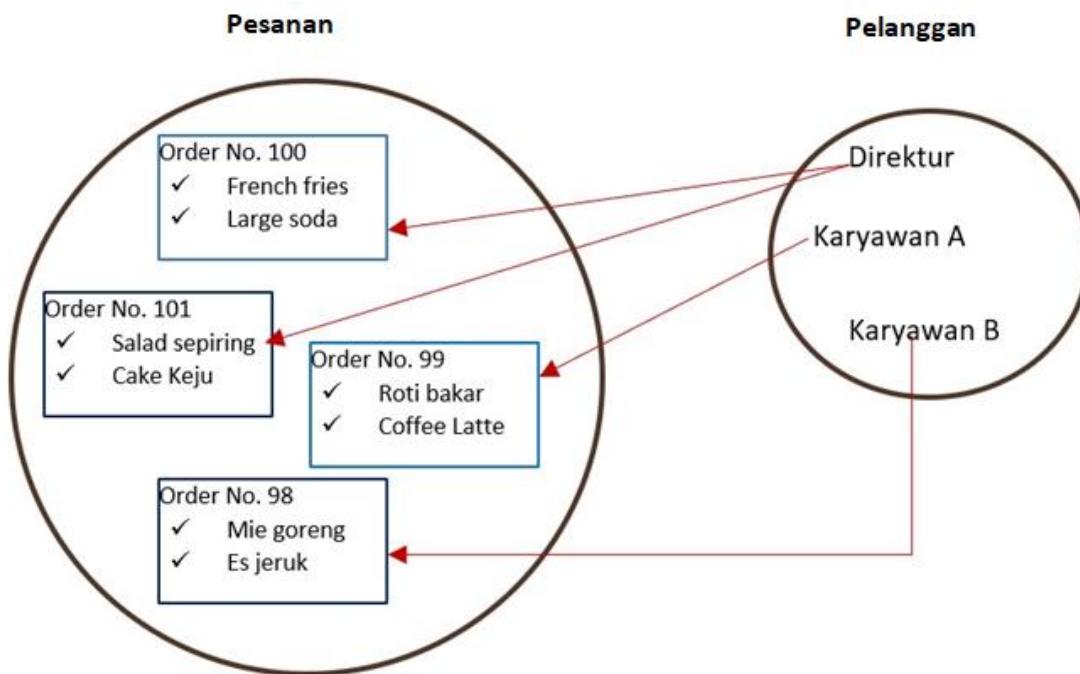
3.6 LANGKAH PRAKTIKUM

Bacalah Skenario kasus A dan Skenario Kasus B berikut ini, lalu selesaikanlah dengan langkah-langkah praktikum untuk penyelesaian masalah dalam Relasi model data, sehingga Anda bisa menjawab pertanyaan-pertanyaan terkait kasus A dan B.

Skenario Bisnis Kasus A

Di sebuah restoran XYZ, seorang pelanggan berjalan ke konter dan melakukan pemesanan. Pelanggan dapat memesan untuk dirinya sendiri saja, atau untuk dirinya sendiri dan orang lain. Misalnya,

seorang Direktur Perusahaan X memesan untuk dirinya sendiri dan karyawannya. Jika Direktur Perusahaan X sebagai pelanggan yang memiliki pesanan dan bertanggung jawab atas pembayaran selama periode waktu tertentu, pelanggan dapat memesan sebanyak mungkin seperti yang dia inginkan. Bagi Pelanggan yang memesan lebih dari 5 jenis menu akan mendapat Bonus.



Proses bisnis yang berlaku dalam menentukan Relasi untuk Skenario bisnis Kasus A dapat diidentifikasi sebagai berikut:

1. PELANGGAN melakukan PESANAN:

- ✓ Terdapat opsionalitas dan kardinalitas
- ✓ Opsional = bersifat Harus/mutlak atau bisa bersifat Mungkin?

2. Untuk Opsionalitas

- ✓ Opsionalitas akan bersifat harus, jika setiap PESANAN harus dilakukan oleh satu (dan hanya satu) PELANGGAN
- ✓ Oprionalitas bersifat Mungkin, jika setiap PELANGGAN harus melakukan satu atau lebih PESANAN

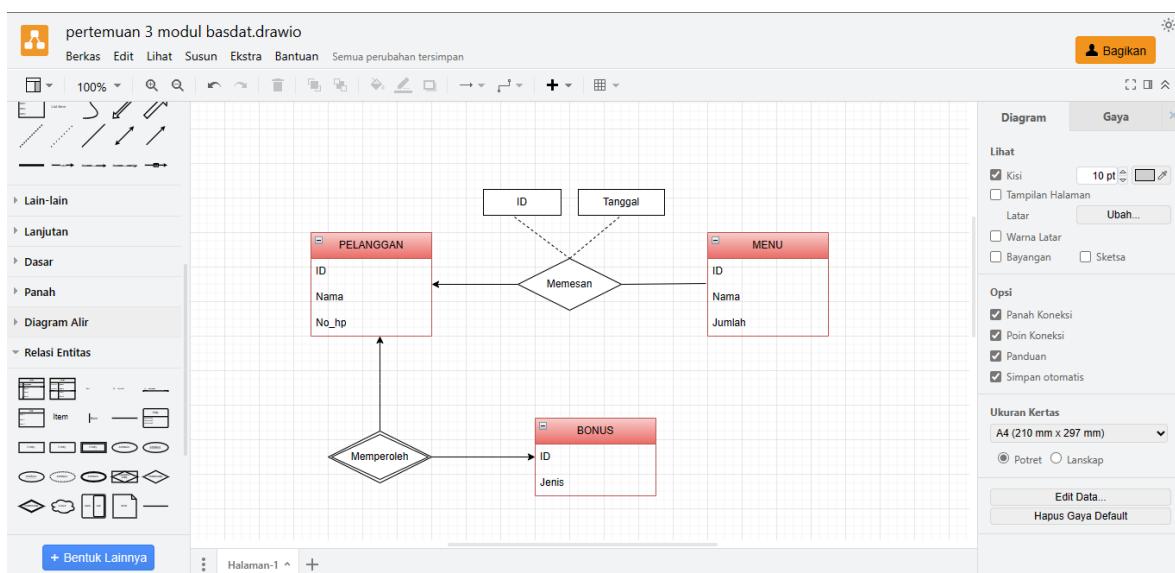
3. Kardinalitas = Berapa Banyak?

- ✓ Setiap PESANAN harus ditempatkan oleh satu dan hanya satu PELANGGAN
- ✓ setiap PELANGGAN harus ditempatkan satu atau lebih PESANAN

Dengan demikian dapat dibuat model Relasi Datanya seperti Gambar 3.1 dengan menggunakan aplikasi Drawing io dengan mengakses URL berikut ini:

<https://app.diagrams.net/#G1bK6HjhwK08e-OcqXB6en3mgWpAxsKRZ0> atau dapat menginstal Visio Drawing. Prinsip kerja kedua aplikasi ini sama.

Hasil Rancangan Relasi Model Data untuk Skenario Kasus A.



Gambar 3. 1 Implementasi Aplikasi dariwing io dalam perancangan Relasi Model Data

Dari Gambar 3.1 dapat dibuat analisa sebagai berikut:

- Terdapat entitas Kuat, yaitu; PELANGGAN dan MENU
- Terdapat Entitas Lemah, yaitu; BONUS karena tidak semua Pelanggan memperoleh BONUS, sehingga Relasi Lemah, yaitu; Memperoleh.
- Atribut Kunci Primer (Primary Key) pada PELANGGAN = ID dan pada MENU = ID sedangkan pada relasi memesan terdapat Atribut Kunci Primer = ID dengan atribut Kunci tamu (Foreign Key) ID PELANGGAN dan ID MENU
- Relasi yang terjadi antar entitas tersebut memiliki kardinalitas One to Many dengan nama relasi memesan.
- Setiap PELANGGAN dapat memesan satu atau lebih MENU.

Skenario Bisnis Kasus B

Dalam Suatu perusahaan XYZ, dilakukan pelacakan karyawan dan manajer mereka. Setiap karyawan memiliki satu manajer, termasuk Direktur Utama yang mengelola dirinya sendiri. Setiap manajer dapat mengelola beberapa karyawan. Seorang Manajer hanya memimpin satu Departemen, dan setiap karyawan menyelesaikan satu atau banyak jenis pekerjaan.

Langkah-Langkah Praktikum:

- Mahasiswa menjalankan aplikasi Visio Drawing
- Mahasiswa membaca Skenario Kasus A dan Skenario Kasus B
- Mahasiswa melakukan analisa terhadap Skenario Kasus A dan B sehingga dapat menidentifikasi relasi yang terjadi berserta mentukan kardinalitasnya.
- Mahasiswa membuat desain relasi model dengan menggunakan aplikasi Visio Drawing/edraw/Drawing io.
- Mahasiswa Membuat relasi table dengan visio drawing dengan memilih jenis menu Entity Relation dan kemudian buat hasil analisanya.

Untuk menyelesaikan permasalahan relasi dalam Skenario Kasus A dan Skenario Kasus B, jawablah pertanyaan berikut ini dengan menerapkan langkah Praktikum 1 sampai 5.

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04	CPMK-01	1. Tentukan nama relasi yang dapat terjadi pada scenario Kasus A!	Jawaban tertulis Jenis Relasi	15
2.	CPL-03 CPL-04 CPL-05	CPMK-02	2. Lakukan juga jenis kardinalitas yang dapat terjadi apakah bersifat Harus atau mungkin..? 3. Tentukan arsitektur data untuk setiap entitas, seperti atribut, primary key, dan lebar datanya dari kasus A tersebut!	Jawaban tertulis jenis kardinalitas Jawaban tertulis arsitektur data dari kasus A	15 15
3	CPL-05 CPL-07	CPMK-03	4. Desainlah model relasi yang terjadi antar entitas tersebut dengan menggunakan Visio Drawing	Hasil rancangan Relasi model data untuk kasus A	20
4	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-01 CPMK-02 CPMK-03	5. Jawablah pertanyan 1 sampai 4 untuk Skenario Kasus B.	Jawaban pertanyaan no 1 sampai 4	35
				Total Nilai	100

3.7 POST TEST

Pelajari Skenario Kasus C dan Skenario Kasus D, kemudian lakukan langkah praktikum 1 sampai 5 untuk menjawab pertanyaan yang diminta.

Skenario Kasus C

Sebuah Toko Sembako menjual barang-barang kebutuhan sehari-hari. Pelanggan yang datang dapat membeli kebutuhan lebih dari satu dan dapat bertransaksi lebih dari satu kali setiap harinya. Setiap barang ditempatkan pada rak yang berbeda. Dalam toko tersebut ada Karyawan dapat melayani satu atau lebih Pelanggan. Barang yang sudah dibeli tidak boleh dikembalikan.

Skenario Kasus D

Sebuah Sanggar Seni memiliki jenis pelatihan yang bermacam-macam, ada Pelatihan Tari, Pelatihan Music, dan Pelatihan Menggambar. Setiap siswa dapat mendaftar satu atau lebih jenis pelatihan.

Setiap jenis pelatihan memiliki satu atau lebih Instruktur. Setiap instruktur memiliki satu atau lebih jadwal, namun setiap instruktur hanya melatih satu jenis pelatihan.

Berdasarkan Skenario Kasus C dan Skenario Kasus D, jawablah pertanyaan-pertanyaan di bawah ini dimana setiap pertanyaan memiliki bobot nilai yang berbeda, dengan total nilai 100.

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04	CPMK-01	Untuk Skenario Kasus C 1. Tentukan nama relasi yang dapat terjadi !	Jawaban tertulis Jenis Relasi	15

2.	CPL-03 CPL-04 CPL-05	CPMK-02	2. Lakukan juga jenis kardinalitas yang dapat terjadi apakah bersifat Harus atau mungkin..? 3. Tentukan arsitektur data untuk setiap entitas, seperti atribut, primary key, dan lebar datanya dari kasus A tersebut!	Jawaban tertulis jenis kardinalitas Jawaban tertulis arsitektur data dari kasus	15
3	CPL-05 CPL-07	CPMK-03	4. Desainlah model relasi yang terjadi antar entitas tersebut dengan menggunakan Visio Drawing	Hasil rancangan Relasi model data untuk kasus	20
4	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-01 CPMK-02 CPMK-03	5. Jawablah pertanyan 1 sampai 4 untuk Skenario Kasus D.	Jawaban pertanyaan no 1 sampai 4	35
				Total Nilai	100

3.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03 CPL-04	CPMK-01 CPMK-02	20%		
2.	Praktik	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-01 CPMK-02 CPMK-03	30%		
3.	Post-Test	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-01 CPMK-02 CPMK-03	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

PRAKTIKUM 4: KONVERSI MODEL DATA KE TABEL

Pertemuan ke : 4

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-03	Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas
CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-01	Mahasiswa memahami konsep data dan informasi (ITE-IMA-02 ACM 2017)
CPMK-02	Mahasiswa memahami, menganalisis dan menyimpulkan arsitektur organisasi dan pemodelan data (ITE-IMA-03 ACM 2017, ITE-IMA-07 ACM 2017)
CPMK-03	Mahasiswa dapat mengimplementasikan pemodelan data

4.1 DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas dengan menggunakan asesmen Pre Test untuk materi terkait Konversi model data ke Tabel
2. Berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya melalui asesmen Pre test.
3. Mengkaji/menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah terkait Konversi Model data ke Tabel dengan asesmen praktik dan post test

4. Memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah terkait konversi model data ke Tabel dengan Asesmen Post Test dan Praktik.

4.2 INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-03 CPL-04	CPMK-01	1. Kemampuan mahasiswa meningkat dalam memahami konsep data dan informasi serta dapat berfikir logis yang diterapkan dengan cara mengkonversikan model data ke Tabel dengan menggunakan aplikasi atau tools desain model basis data yang tersedia.
CPL-03 CPL-04 CPL-05	CPMK-02	2. Kemampuan mahasiswa meningkat dalam memahami, menganalisis dan menyimpulkan arsitektur organisasi dan pemodelan data yang terkait konversi Model Data ke Tabel 3. Kemampuan mahasiswa meningkat dalam mengkaji/menganalisis permasalahan terkait konversi Model Data ke Tabel dengan mempertimbangkan berbagai aspek 1. Kemampuan mahasiswa meningkat dalam memilih, membuat dan menerapkannya dengan menggunakan perangkat desain basis data yang tersedia untuk penyelesaian masalah terkait Konversi Model Data ke Tabel
CPL-05 CPL-07	CPMK-03	2. Kemampuan mahasiswa meningkat dalam menganalisa, memilih dan mengimplementasikan masalah terkait konversi Model Data ke Tabel dengan menggunakan tool pemodelan yang tersedia

4.3 TEORI PENDUKUNG

Algoritma Pemetaan 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 ini melahirkan beberapa teori atau metode antara lain adalah pemetaan ER to tabel dan Normalisasi.

Uraian materi di bawah ini menjelaskan pemetaan 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 multivalue tersebut. Tambahkan primary key dari relasi yang memiliki multivalue 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 pemetaan sesuai dengan aturan mapping.

4.4 HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Buku Petunjuk Praktikum
3. Aplikasi pemodelan: visio drawing, edraw atau Draw io

4.5 PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03 CPL-04	CPMK-01	1. Untuk menentukan jenis relasi dalam suatu model data sangat bergantung dengan proses bisnis dalam suatu organisasi. Mengapa demikian..?	25
2.	CPL-03 CPL-04	CPMK-02	2. Jelaskan bagaimana peran relasi dalam menganalisis dan menyimpulkan arsitektur organisasi dan pemodelan data? 3. Jelaskan Aspek apa saja yang perlu dipertimbangkan dalam mengkaji/menganalisis permasalahan terkait Relasi? 4. Untuk memulai dalam menganalisa terkait masalah Relasi dalam pemodelan data bagaimana langkah-langkahnya..?	25 25 25
			Total Nilai	100

4.6 LANGKAH PRAKTIKUM

Bacalah Skenario kasus A dan Skenario Kasus B berikut ini, lalu selesaikanlah dengan langkah-langkah praktikum untuk penyelesaian masalah dalam Relasi model data, sehingga Anda bisa menjawab pertanyaan-pertanyaan terkait kasus A dan B.

Skenario Bisnis Kasus A

Dalam sebuah Perusahaan Arsitek yang baru berkembang, memiliki 5 DEPARTEMEN. Setiap DEPARTEMEN memiliki satu atau lebih KARYAWAN. Setiap KARYAWAN mengerjakan satu atau lebih PROYEK. Karyawan memperoleh BONUS jika dapat menyelesaikan sesuai JADWAL. Setiap Karyawan memiliki satu atau lebih JADWAL dalam PROYEK.

Dari Skenario Kasus A akan membuat Konversi dari Relasi model data menjadi Tabel, dengan cara sebagai berikut:

1. Mendefinisikan entitas Kuat dan Entitas Lemah

Pada Skenario Kasus A terdapat Entitas Kuat: DEPARTEMEN, KARYAWAN dan PROYEK, sedangkan Entitas lemah adalah BONUS karena tidak semua karyawan mendapatkan Bonus.

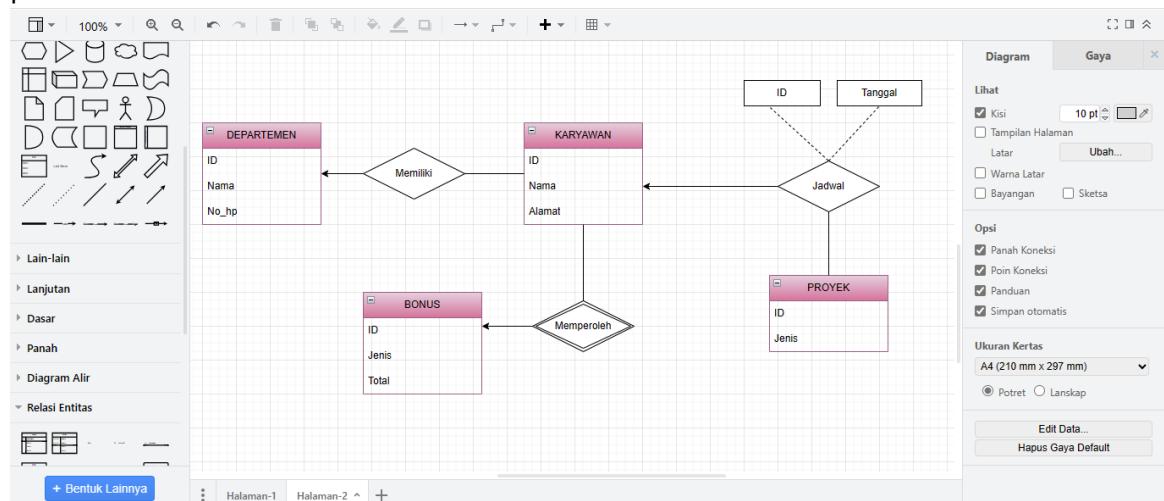
2. Menentukan Atribut, atribut kunci primer dan atribut kunci tamu.

Entitas	Atribut	Keterangan
DEPARTEMEN	ID, Nama	PK = ID
KARYAWAN	ID, Nama, Alamat, email, No-HP	PK = ID
PROYEK	ID, Nama-Proyek	PK = ID
JADWAL	Tanggal, ID Proyek, ID Karyawan	FK = ID Proyek, ID Karyawan
BONUS	ID, Jenis, Total	PK = ID

3. Mendefinisikan tipe relasi yang terjadi antar entitas beserta Kardinalitasnya

Relasi	Atribut	Kardinalitas	Keterangan
Memiliki		one to many	FK = ID Departemen, ID Karyawan
Mengerjakan		One to many	PK = ID Karyawan, ID Proyek
Memperoleh		One to many	FK = ID Proyek, ID Karyawan
Jadwal	Tanggal,	One to one	FK = ID Proyek, ID Karyawan

4. Membuat desain Model Data dengan menggunakan aplikasi Drawing io. Hasilnya ditunjukkan pada Gambar 4.1.



Gambar 4. 1 Relasi Model Data

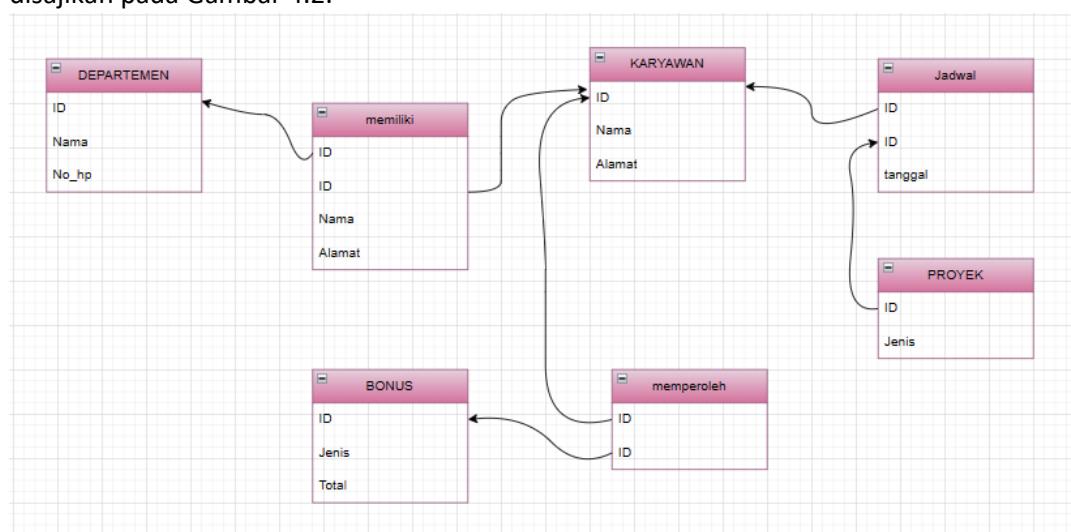
5. Membuat Struktur Fisik Tabel

Selanjutnya membuat table dengan mengacu pada Gambar 5.1 maka diidentifikasi ada 7 tabel, yaitu:

- a. Tabel Departemen
- b. Tabel proyek

- c. Tabel Karyawan
 - d. Tabel Bonus
 - e. Tabel Jadwal
 - f. Tabel memiliki
 - g. Tabel Memperoleh
6. Membuat Pemetaan Tabel/ Relasi Tabel.

Caranya dengan mengacu hasil Gambar 4.1 dapat dibuat Pemetaan Tabelnya seperti yang disajikan pada Gambar 4.2.



Gambar 4. 2 Hasil Konversi Relasi Model Data Tabel.Mapping Table

Skenario Bisnis Kasus B

Dalam Suatu perusahaan XYZ, dilakukan pelacakan karyawan dan manajer mereka. Setiap karyawan memiliki satu manajer, termasuk Direktur Utama yang mengelola dirinya sendiri. Setiap manajer dapat mengelola beberapa karyawan. Seorang Manajer hanya memimpin satu Departemen, dan setiap karyawan menyelesaikan satu atau banyak jenis pekerjaan.

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-01 CPMK-02 CPMK-03	1. Praktekkanlah langkah 1 sampai 6 pada pada Skenario Kasus A.	Jawaban hasil praktek Kasus A	40
2.	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-01 CPMK-02 CPMK-03	2. Praktekkanlah langkah 1 sampai 6 pada pada Skenario Kasus B	Jawaban hasil praktek Kasus B	60
				Total Nilai	100

4.7 POST TEST

Pelajari Skenario Kasus C dan Skenario Kasus D, kemudian lakukan langkah praktikum 1 sampai 5 untuk menjawab pertanyaan yang diminta.

Skenario Kasus C

Sebuah Apotek menjual berbagai jenis obat. Pelanggan yang datang dapat membeli obat lebih dari satu dan dapat bertransaksi lebih dari satu kali setiap harinya. Setiap Obat ditempatkan pada rak yang berbeda. Dalam Apotek tersebut Apoteker menerima resep satu atau lebih dari Pelanggan. Setiap Apoteker meramu satu atau lebih obat. Setiap resep memiliki satu atau lebih obat. Apoteker mencatat Obat setiap hari.

Skenario Kasus D

Pemerintah Daerah memiliki satu atau lebih unit kerja. Setiap Unit kerja berada pada satu gedung. Setiap unit kerja melayani satu jenis layanan. Setiap Unit Kerja memiliki satu atau lebih Sistem informasi. Setiap Unit Kerja memiliki satu atau beberapa Karyawan. Setiap Karyawan mengakses satu atau lebih Sistem Informasi. Setiap informasi memiliki satu akun.

Berdasarkan Skenario Kasus C dan Skenario Kasus D, jawablah pertanyaan-pertanyaan di bawah ini dimana setiap pertanyaan memiliki bobot nilai yang berbeda, dengan total nilai 100.

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-01 CPMK-02 CPMK-03	1. Praktekkanlah langkah 1 sampai 6 pada pada Skenario Kasus C.	Jawaban hasil praktek Kasus A	50
2.	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-01 CPMK-02 CPMK-03	2. Praktekkanlah langkah 1 sampai 6 pada pada Skenario Kasus BD	Jawaban hasil praktek Kasus B	50
				Total Nilai	100

4.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03 CPL-04	CPMK-01 CPMK-02	20%		
2.	Praktik	CPL-04 CPL-05 CPL-07	CPMK-02 CPMK-03	30%		
3.	Post-Test	CPL-04 CPL-05 CPL-07	CPMK-02 CPMK-03	50%		
Total Nilai						

Daftar Pustaka : https://repository.dinus.ac.id/docs/ajar/9.materi_ERD_.pdf

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

PRAKTIKUM 5: IMPLEMENTASI RELASI DAN KONVERSI TABEL

Pertemuan ke : 5

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-03	Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas
CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-01	Mahasiswa memahami konsep data dan informasi (ITE-IMA-02 ACM 2017)
CPMK-03	Mahasiswa dapat mengimplementasikan pemodelan data

5.1 DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas dengan asesmen Pre Test.
2. Mengkaji/menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah terkait Konversi Model data ke Tabel dengan asesmen praktik dan post test
3. Memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah terkait konversi model data ke Tabel dengan Asesmen Post Test dan Praktik.

5.2 INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-03 CPL-04	CPMK-01	1. Kemampuan mahasiswa meningkat dalam memahami konsep data dan informasi serta dapat berfikir logis yang diterapkan dengan membuat rancangan model data dan konversi tabel dengan menggunakan PHP My SQL.
CPL-05 CPL-07	CPMK-02	2. Kemampuan mahasiswa meningkat dalam mengkaji/menganalisis permasalahan membuat rancangan model data dan konversi table dengan menggunakan PHP My SQL. 3. Kemampuan mahasiswa meningkat dalam memilih, membuat dan menerapkannya dengan menggunakan perangkat desain basis data yang tersedia untuk penyelesaian masalah membuat rancangan model data dan konversi tabel dengan menggunakan PHP My SQL.

5.3 TEORI PENDUKUNG

PHPMyAdmin adalah perangkat lunak gratis yang ditulis dalam PHP, dimaksudkan untuk menangani administrasi MySQL melalui Web. PHPMyAdmin mendukung berbagai operasi di MySQL dan MariaDB. Operasi yang sering digunakan (mengelola database, tabel, kolom, relasi, indeks, pengguna, izin, dll) dapat dilakukan melalui antarmuka pengguna, sementara Anda masih memiliki kemampuan untuk mengeksekusi pernyataan SQL secara langsung.

PHPMyAdmin adalah proyek yang matang dengan basis kode yang stabil dan fleksibel; Anda dapat mengetahui lebih lanjut tentang proyek dan sejarahnya serta penghargaan yang diperolehnya. Ketika proyek berusia 15 tahun, kami menerbitkan halaman perayaan. Proyek PHPMyAdmin adalah anggota dari Software Freedom Conservancy. SFC adalah organisasi nirlaba yang membantu mempromosikan, meningkatkan, mengembangkan, dan mempertahankan proyek Perangkat Lunak Bebas, Gratis, dan Sumber Terbuka (FLOSS).

Fitur dalam PHPMyAdmin:

1. Antarmuka web yang intuitif
2. Dukungan untuk sebagian besar fitur MySQL: telusuri dan membuat database, tabel, tampilan, bidang, dan indeks salin, jatuhkan, ganti nama, dan ubah basis data, tabel, bidang, dan indeks
3. mengelola akun pengguna dan hak istimewa MySQL
4. mengelola prosedur dan pemicu yang tersimpan
5. Impor data dari CSV dan SQL
6. Eksport data ke berbagai format: CSV, SQL, XML, PDF, ISO/IEC 26300 - Teks dan Spreadsheet OpenDocument, Word, LATEX, dan lainnya.
7. Mengelola beberapa server
8. Membuat grafik tata letak basis data Anda dalam berbagai format
9. Membuat kueri kompleks menggunakan Query-by-example (QBE)
10. Mencari secara global dalam database atau subsetnya

Cara Instal XAMPP Di Windows:

1. Download aplikasi XAMPP terbaru, klik disini. Pilih salah satu (tips: jika Anda bingung, bisa pilih yang versi 8.0.28).

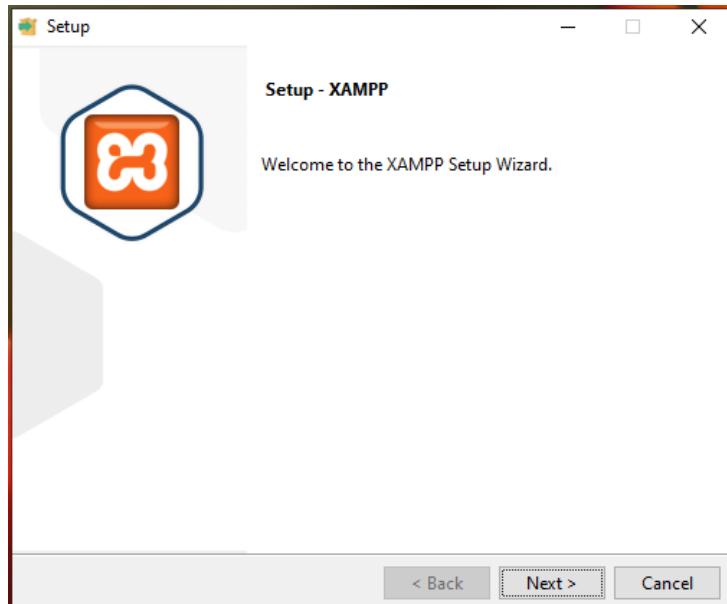
XAMPP for Windows 8.0.28, 8.1.17 & 8.2.4

Version	Checksum	Size
8.0.28 / PHP 8.0.28	What's Included? md5 sha1	Download (64 bit) 144 Mb
8.1.17 / PHP 8.1.17	What's Included? md5 sha1	Download (64 bit) 148 Mb
8.2.4 / PHP 8.2.4	What's Included? md5 sha1	Download (64 bit) 149 Mb

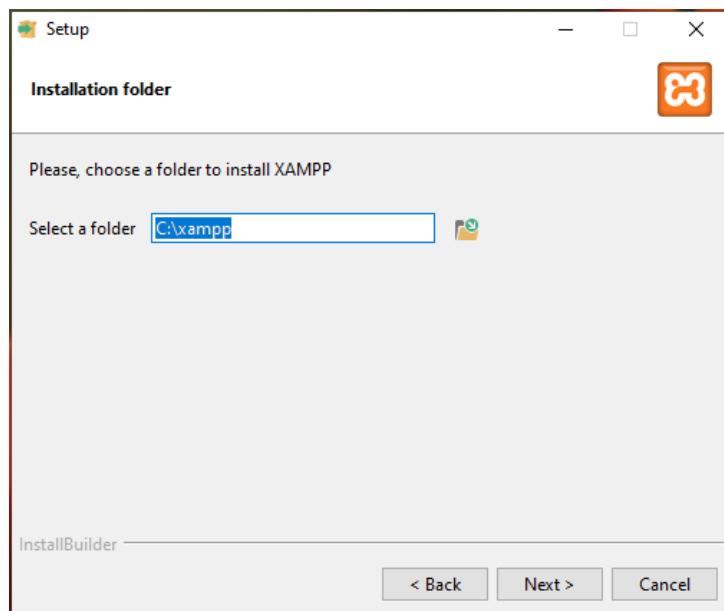
[Requirements](#) [More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

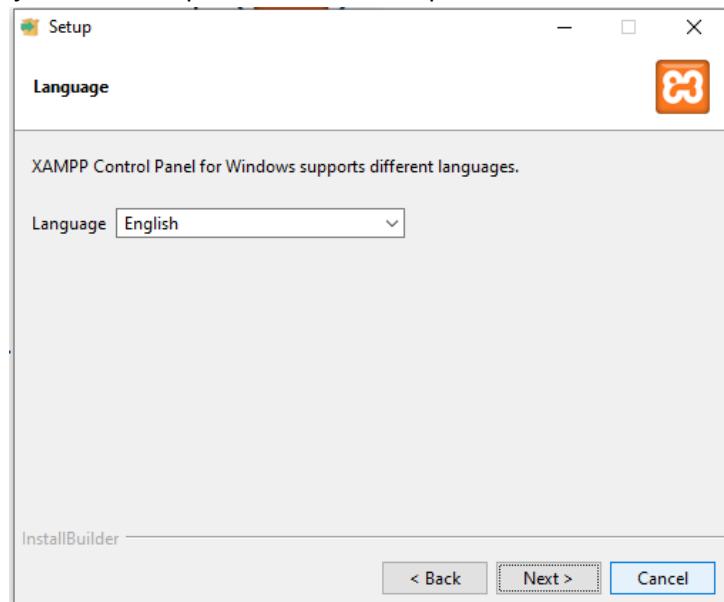
- Dobel klik file XAMPP yang baru saja Anda download, nanti selanjutnya akan muncul jendela “installer” seperti di bawah ini:



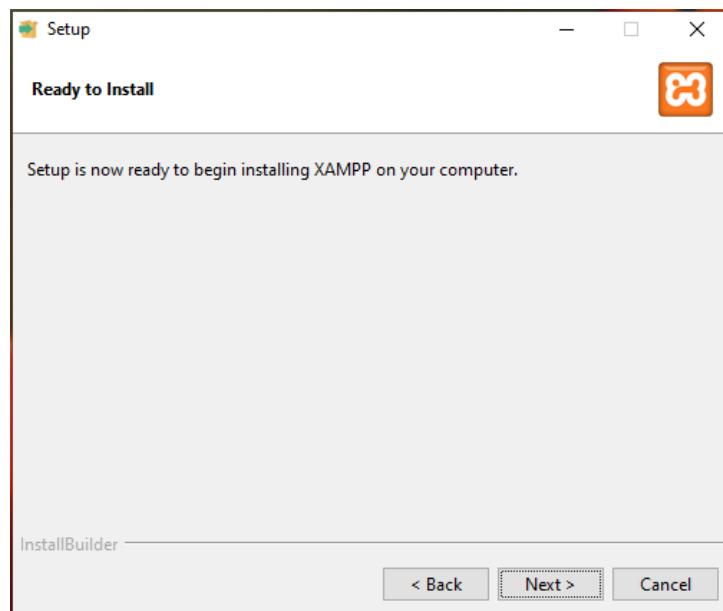
- Selanjutnya Klik tombol “Next”. Sampai Muncul Jendela untuk penyimpanan file instalasi, Kemudian Anda akan diminta untuk menentukan lokasi folder penyimpanan file-file dan folder XAMPP. Secara default akan diarahkan ke lokasi c:\xampp. Namun jika Anda ingin menyimpannya di folder lain bisa klik browse dan tentukan secara manual folder yang ingin digunakan. Tampilan seperti gambar dibawah ini:



4. Kadang pada proses muncul pesan error. Jika ada, abaikan saja dan lanjutkan dengan klik Next dan akan muncul jendela untuk pemilihan bahasa seperti dibawah ini:



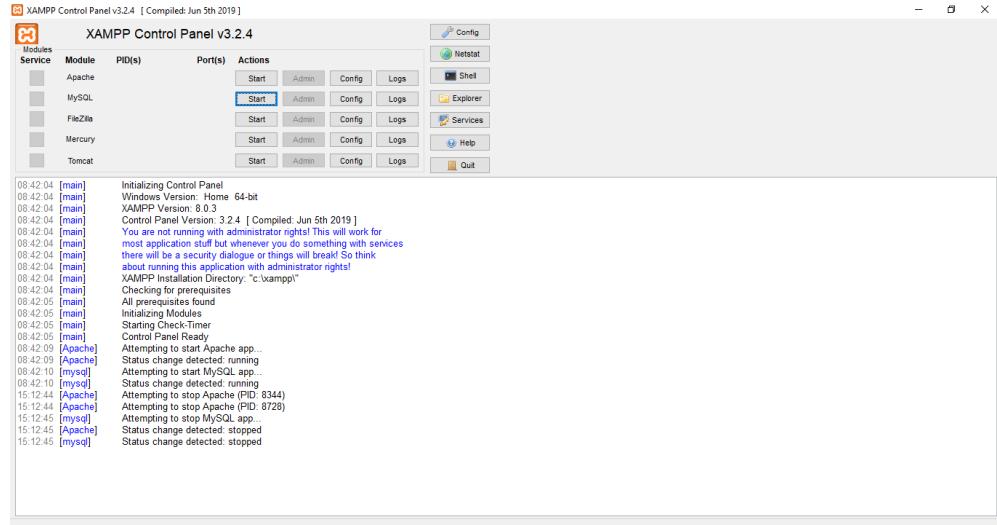
5. Kemudian lanjutkan proses instalasi dengan mengklik tombol "Next" hingga finish.
6. Di akhir proses intalasi, jika Anda sudah memiliki XAMPP sebelumnya maka akan muncul jendela pemberitahuan seperti dibawah ini:

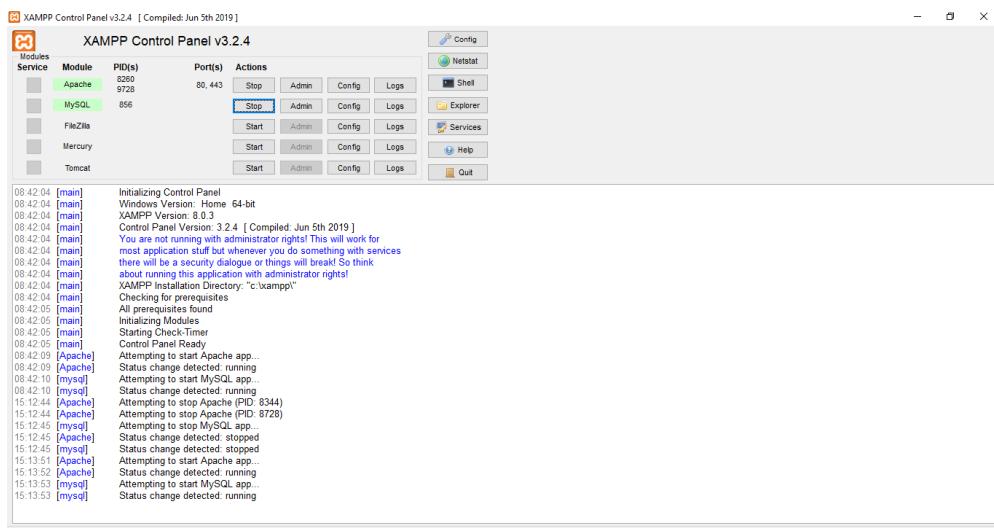


Jika muncul jendela diatas cukup batalkan proses instalasi dengan menekan tombol “Cancel” dan gunakan XAMPP yang telah ada di PC/Laptop.

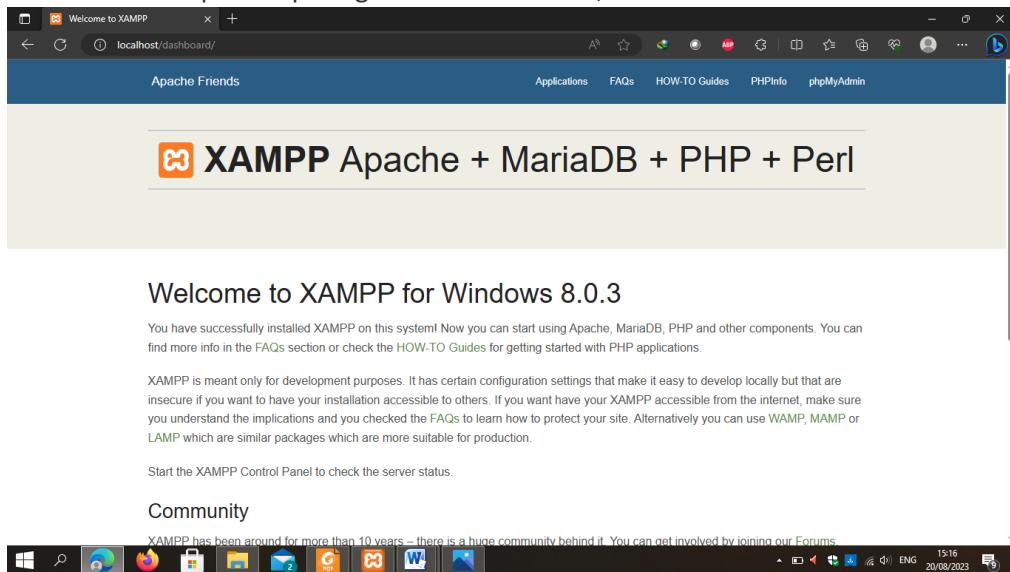
Cara Menjalankan Aplikasi XAMPP

1. Bukalah aplikasi XAMPP, bisa melalui Start Menu atau Desktop, dan klik icon XAMPP.
2. Setelah terbuka, silahkan klik tombol Start pada kolom Action sehingga tombol tersebut berubah menjadi Stop. Dengan mengklik tombol tersebut, artinya itulah aplikasi yang dijalankan. Biasanya jika saya menggunakan XAMPP, yang saya start hanyalah aplikasi Apache dan MySQL, karena saya tidak memerlukan aplikasi seperti Filezilla, dan lain-lain.





3. Sekarang bukalah browser kesukaan Anda, dan coba ketikkan <http://localhost/> di address bar. Jika muncul tampilan seperti gambar di bawah ini, instalasi telah berhasil.



5.4 HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Buku Petunjuk Praktikum
3. Aplikasi pemodelan: XAMPP.
4. Browser (firefox, chrome).

5.5 PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03 CPL-04	CPMK-01 CPMK-02	1. Jelaskan keunggulan PHP MySQL 2. Jelaskan cara kerja PHP My SQL	20 20
2.	CPL-03 CPL-04	CPMK-01 CPMK-02	3. Jelaskan bagaimana membuat desain basis data dengan PHP MySQL 4. Jelaskan Langkah-langkah membuat Struktur	20 20

		Tabel dengan menggunakan PHP MySQL 5. Jelaskan Langkah-langkah membuat relasi table dengan menggunakan PHP MySQL	20
Total Nilai			100

5.6 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_Matkul, Nama, sks. Kemudian isi data pada masing-masing tabel sebanyak 3 data seperti pada pertemuan sebelumnya.

Tabel Dosen :

				kode_dsn	nama_dsn	alamat_dsn
+ Opsi	Ubah	Salin	Hapus			
□	Ubah	Salin	Hapus	18001	Sastr Wardoyo	Yogyakarta
□	Ubah	Salin	Hapus	18002	Ayana Maladewi	Yogyakarta
□	Ubah	Salin	Hapus	18003	Mahardika	Aceh

Gambar 5. 1 Tabel Dosen dengan XAMPP

Tabel Mata_Kuliah:

				Kode_Matkul	Nama	sks
+ Opsi	Ubah	Salin	Hapus			
□	Ubah	Salin	Hapus	3	Basis Data	3
□	Ubah	Salin	Hapus	4	KDK	3
□	Ubah	Salin	Hapus	5	Data Mining	2

Gambar 5. 2 Membuat Tabel mata Kuliah dengan XAMPP

2. Tabel Relasi One to One

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

Nama	Jenis	Panjang/Nilai	Bawaan	Penyortiran	Atribut	Tak Termilai	Indeks	AJ
kode_kelas	VARCHAR	10	Tidak ada			<input type="checkbox"/>	PRIMARY	<input type="checkbox"/>
angkatan	VARCHAR	50	Tidak ada			<input type="checkbox"/>	---	<input type="checkbox"/>
nim_ketua	VARCHAR	15	Tidak ada			<input type="checkbox"/>	---	<input type="checkbox"/>
	INT		Tidak ada			<input type="checkbox"/>	---	<input type="checkbox"/>
Struktur								

Gambar 5. 3 Membuat Tabel Kelas untuk Membuat Relasi

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
				With selected:	

Gambar 5. 4 Tampilan desain table dengan XAMPP

Berdasarkan langkah-langkah praktikum, jawablah pertanyaan berikut ini.

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-01 CPMK-02 CPMK-03	1. Buatlah desain table dan relasinya dengan menggunakan aplikasi XSAM	Jawaban hasil praktek Kasus A	40
2.	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-01 CPMK-02 CPMK-03	2. Buatlah Desain Tabel dan relasi dengan XSAM untuk Skenario Kasus untuk Latihan 3. Bagaimana hasil analisa Anda dalam menggunakan XSAM	Jawaban hasil praktek Kasus B	60
				Total Nilai	100

Skenario Kasus Untuk Latihan:

Dalam sebuah Perusahaan Arsitek yang baru berkembang, memiliki 5 DEPARTEMEN. Setiap DEPARTEMEN memiliki satu atau lebih KARYAWAN. Setiap KARYAWAN mengerjakan satu atau lebih PROYEK. Bagi karyawan memperoleh BONUS jika dapat menyelesaikan sesuai JADWAL. Setiap Karyawan memiliki satu atau lebih JADWAL dalam PROYEK.

5.7 POST TEST

Pelajari Skenario Kasus C dan Skenario Kasus D, kemudian jawablah pertanyaan untuk Post Test ini.

Skenario Kasus C

Sebuah Apotek menjual berbagai jenis obat. Pelanggan yang datang dapat membeli obat lebih dari satu dan dapat bertransaksi lebih dari satu kali setiap harinya. Setiap Obat ditempatkan pada rak yang berbeda. Dalam Apotek tersebut Apoteker menerima resep satu atau lebih dari Pelanggan. Setiap Apoteker meramu satu atau lebih obat. Setiap resep memiliki satu atau lebih obat. Apoteker mencatat Obat setiap hari.

Skenario Kasus D

Pemerintah Daerah memiliki satu atau lebih unit kerja. Setiap Unit kerja berada pada satu gedung. Setiap unit kerja melayani satu jenis layanan. Setiap Unit Kerja memiliki satu atau lebih Sistem informasi. Setiap Unit Kerja memiliki satu atau beberapa Karyawan. Setiap Karyawan mengakses satu atau lebih Sistem Informasi. Setiap informasi memiliki satu akun.

Berdasarkan Skenario Kasus C dan Skenario Kasus D, jawablah pertanyaan-pertanyaan di bawah ini dimana setiap pertanyaan memiliki bobot nilai yang berbeda, dengan total nilai 100.

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04	CPMK-01 CPMK-02	1. Desain Relasi, Tabel dan model datanya dengan menggunakan XAMPP untuk kasus C	Jawaban hasil praktek Kasus C	50
2.	CPL-05 CPL-07	CPMK-03	2. Desain Relasi, Tabel dan model datanya dengan menggunakan XAMPP untuk kasus D	Jawaban hasil praktek Kasus D	50
			Total Nilai		100

5.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03 CPL-04	CPMK-01	20%		
2.	Praktik	CPL-04 CPL-05 CPL-07	CPMK-02 CPMK-03	30%		
3.	Post-Test	CPL-04 CPL-05 CPL-07	CPMK-02 CPMK-03	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

PRAKTIKUM 6: NORMALISASI

Pertemuan ke : 6

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-03	Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas
CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-03	Mahasiswa dapat memahami, menganalisa, dan mengimplementasikan Arsitektur Organisasi Data dengan konsep Normalisasi dan Bussines Rules

6.1 DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas dengan menggunakan asesmen praktik dan posttest.
2. Berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya melalui asesmen Pre test, praktik dan post test.
3. Mengkaji/menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu dengan asesmen Pre test, praktik dan post test
4. Memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah.

6.2 INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-03	CPMK-03	1. Kemampuan mahasiswa meningkat dalam memahami Normalisasi dan Bussines Rules
CPL-04	CPMK-03	2. Kemampuan mahasiswa berfikir logis yang diterapkan dengan cara memodelkan masalah terkait Relasi dengan menggunakan aplikasi atau tools desain model basis data yang tersedia.
CPL-05 CPL-07	CPMK-03	3. Kemampuan mahasiswa meningkat dalam menganalisa, memilih dan mengimplementasikan masalah terkait Normalisasi dan Bussines Rules dengan menggunakan tool pemodelan yang tersedia

6.3 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 redundansi 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 atribut yang berulang atau bernilai ganda, setiap atribut yang dimiliknya 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 bentuk 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.

6.4 HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Buku Petunjuk Praktikum
3. Aplikasi pemodelan: visio drawing, edraw atau Draw io

6.5 PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-03	Jelaskan apa yang dimaksud dengan Normalisasi Tabel	25
2.	CPL-04	CPMK-03	Sebutkan dan jelaskan tujuan Normalisasi	25
3.	CPL-05	CPMK-03	Sebutkan dan jelaskan tahapan Normalisasi yang umum dipakai	25
4.	CPL-07	CPMK-03	Carilah contoh bentuk normalisasi (diluar contoh yang ada dimodul, dan table tidak boleh copy paste, harus dibikin asli sampai bentuk 2 normalisasi)	25

6.6 LANGKAH PRAKTIKUM

- Bacalah kasus terkait entitas mahasiswa yang belum ternormalisasi pada Tabel 6.1. untuk proses Normalisasi dengan MySql menggunakan perinta: 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

- 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

- 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 (redundansi), untuk itu, data yang sama bisa dihilangkan.

Tabel 6. 7 Entitas 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 6.8 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		NIDN	Nama Dosen
1078523	Riyanti Anjani	Menjadi	1078523	Riyanti Anjani
1078523	Riyanti Anjani		1075047	Susan Savitri
1078523	Riyanti Anjani		1077021	Erwin Masadi
1078523	Riyanti Anjani			
1075047	Susan Savitri			
1075047	Susan Savitri			

Tabel 6. 10 Tabel Ambil MK Bentuk Normal 3

NIM	Makul	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

menjadi

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

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-03	Selesaikan Langkah praktikum 1	Hasil praktikum Langkah 1 (Screenshot)	25
2.	CPL-04	CPMK-03	Selesaikan Langkah praktikum 2	Hasil praktikum Langkah 2 (Screenshot)	25
3.	CPL-05	CPMK-03	Selesaikan Langkah praktikum 3-4	Hasil praktikum Langkah 3-4 (Screenshot)	25
4.	CPL-07	CPMK-03	Selesaikan Langkah praktikum 5-6	Hasil praktikum Langkah 5-6 (Screenshot)	25

6.7 POST TEST

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

Jawablah pertanyaan berikut (**Total Skor : 100**):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04	CPMK-03	Lakukan Normalisasi dari tabel diatas sampai dihasilkan tabel yang berkualitas baik!	Jawaban tertulis	50
2.	CPL-05 CPL-07	CPMK-03	Implementasikan tabel hasil Normalisasi pada PHPMYADMIN dan isikan datanya sesuai dengan kasus tersebut	Screenshot bukti bahwa tabel sudah saling berelasi	50

6.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-03	20%		
2.	Praktik	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-03	30%		
3.	Post-Test	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-03	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

PRAKTIKUM 7: DDL

Pertemuan ke : 7

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-03	Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas
CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPL-08	Mampu merancang dan mengimplementasikan algoritma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan perangkat lunak dan pemikiran komputasi
CPMK-04	Mahasiswa dapat memahami, menganalisa, dan mengimplementasikan Bahasa query basis data DDL dan DML

7.1 DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas dengan menggunakan asesmen Pre Test untuk materi terkait DDL.
2. Berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya melalui asesmen Pre test.

3. Mengkaji/menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah terkait DDL dengan asesmen praktik dan post test.
4. Memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah terkait DDL dengan Asesmen Post Test dan Praktik.
5. Merancang dan mengimplementasikan algoritma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan perangkat lunak dan pemikiran komputasi.

7.2 INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-03 CPL-04	CPMK-04	1. Kemampuan mahasiswa meningkat dalam memahami konsep data dan informasi serta dapat berfikir logis yang diterapkan dengan cara mempraktikan konsep DDL dan DML
CPL-05 CPL-07 CPL-08	CPMK-04	2. Kemampuan mahasiswa meningkat dalam menganalisa, memilih dan mengimplementasikan masalah terkait DDL dan DML dengan menggunakan tool pemodelan yang tersedia

7.3 TEORI PENDUKUNG

DDL (Data Definition Language) ialah bahasa yang digunakan untuk mendeskripsikan data dan hubungannya dalam suatu database. DDL digunakan untuk membuat dan memodifikasi struktur objek pada suatu database menggunakan perintah dan sintaks spesifik yang telah ditetapkan. Dalam pengertian umum, DDL juga digunakan untuk mengacu pada bahasa yang mendeskripsikan data.

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

7.4 HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Buku Petunjuk Praktikum
3. XAMPP
4. Browser (firefox, chrome)

7.5 PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-04	1. Jelaskan pengertian DDL ?	25
2.	CPL-04	CPMK-04	2. Jelaskan perbedaan DDL dan DML?	25
3.	CPL-05	CPMK-04	3. Jelaskan fungsi DDL dalam Database?	25
4.	CPL-07	CPMK-04	4. Berikan contoh minimal 3 dari perintah DDL	25
5	CPL-08	CPMK-04		

7.6 LANGKAH PRAKTIKUM

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

Sebelum Masuk ke perintah DDL, Buka terlebih dahulu Jendela Shell yang terdapat pada XAMPP, kemudian ketik perintah “mysql -u root” untuk mengakses mysql.

Perintah DDL:

- 1) Menampilkan database yang ada pada mysql dengan perintah SHOW databases;

```
XAMPP for Windows - mysql -u root
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| alfa_barbershop |
| basdat10 |
| crud |
| dahboard_smawonosobo |
| db_rental_mobil |
| db_rs |
| db_school |
| dbcrud22 |
| information_schema |
| kuliah |
| laprak10 |
| laprak6 |
| laprak6_2 |
| laprak7 |
| laprak7basdat |
| laprak9 |
+-----+
```

Gambar 7. 1 Menampilkan Database di MySQL

- 2) Membuat database

```
CREATE database <nama_database>;
CREATE database akademik;
```

```
MariaDB [(none)]> create database akademik;
Query OK, 1 row affected (0.034 sec)

MariaDB [(none)]>
```

Gambar 7. 2 Perintah untuk Membuat Database Akademik

- 3) Menggunakan database

```
USE <nama_database>;
USE akademik;
```

```
MariaDB [(none)]> use akademik;
Database changed
MariaDB [akademik]>
```

Gambar 7. 3 Perintah untuk Menggunakan Database

- 4) Menghapus database

```
DROP database <nama_database>;
DROP database akademik;
```

```
MariaDB [(none)]> use akademik;
Database changed
MariaDB [akademik]> show databases;
+-----+
| Database           |
+-----+
| akademik          |
| alfa_barbershop   |
| basdat10          |
+-----+
```

Gambar 7. 4 Melihat Database

```
MariaDB [(none)]> drop database akademik;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> ■
```

Gambar 7. 5 Hapus Database

```
MariaDB [(none)]> show databases;
+-----+
| Database           |
+-----+
| alfa_barbershop   |
| basdat10          |
| crud               |
+-----+
```

Gambar 7. 6 Tampilan Database Setelah Perintah DROP Database

- 5) Melihat tabel apa yang sudah ada dalam database yang aktif
SHOW tables;

```
MariaDB [(none)]> use akademik;
Database changed
MariaDB [akademik]> show tables;
Empty set (0.001 sec)

MariaDB [akademik]>
```

Gambar 7. 7 Show Table

- 6) Membuat tabel

```
CREATE TABLE <nama_table> (<nama_kolom><tipedata>);
CREATE TABLE mhs (NIM CHAR (8) NOT NULL PRIMARY KEY, nama_mhs VARCHAR (50), alamat
VARCHAR (50));
```

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.

```
MariaDB [akademik]> create table mhs (NIM char(8) NOT NULL PRIMARY KEY, nama_mhs varchar(50), alamat varchar(50));
Query OK, 0 rows affected (0.289 sec)

MariaDB [akademik]>
```

Gambar 7. 8 Membuat Tabel

- 7) Melihat struktur table

```
DESCRIBE <nama_tabel>;
DESC <nama_tabel>;
DESCRIBE mhs;
DESC mhs;
```

```
MariaDB [akademik]> desc mhs;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| NIM   | char(8) | NO   | PRI | NULL    |       |
| nama_mhs | varchar(50) | YES  |     | NULL    |       |
| alamat | varchar(50) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.097 sec)

MariaDB [akademik]>
```

Gambar 7. 9 Melihat Struktur Table

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;

```
XAMPP for Windows - mysql -u root

MariaDB [akademik]> ALTER TABLE mhs ADD no_telp INT AFTER alamat;
Query OK, 0 rows affected (0.077 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [akademik]> desc mhs;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| NIM   | char(8) | NO   | PRI  | NULL    |       |
| nama_mhs | varchar(50) | YES  |       | NULL    |       |
| alamat | varchar(50) | YES  |       | NULL    |       |
| no_telp | int(11)  | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.015 sec)

MariaDB [akademik]>
```

Gambar 7. 10 Tambah Atribut

b. Mengubah nama kolom

Parameter yang digunakan adalah CHANGE.

ALTER TABLE nama_tabel CHANGE kolom_lama kolom_baru tipe (panjang)

ALTER TABLE mhs CHANGE alamat alamat_mhs VARCHAR (35)

```
XAMPP for Windows - mysql -u root

MariaDB [akademik]> desc mhs
-> ;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| NIM   | char(8) | NO   | PRI  | NULL    |       |
| nama_mhs | varchar(50) | YES  |       | NULL    |       |
| alamat_mhs | varchar(50) | YES  |       | NULL    |       |
| no_telp | int(11)  | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.035 sec)

MariaDB [akademik]>
```

Gambar 7. 11 Ubah Nama Atribut

c. Mengubah nama tabel

Parameter yang digunakan adalah RENAME.

ALTER TABLE nama_lama RENAME [TO] nama_baru;

ALTER TABLE mhs RENAME TO mahasiswa;

```
MariaDB [akademik]> ALTER TABLE mhs RENAME TO mahasiswa;
Query OK, 0 rows affected (0.156 sec)

MariaDB [akademik]> show tables;
+-----+
| Tables_in_akademik |
+-----+
| mahasiswa          |
+-----+
1 row in set (0.001 sec)

MariaDB [akademik]>
```

Gambar 7. 12 Ubah Nama Table

- 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;

```
MariaDB [akademik]> ALTER TABLE mahasiswa DROP no_telp;
Query OK, 0 rows affected (0.082 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [akademik]> desc mahasiswa;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| NIM        | char(8)    | NO   | PRI | NULL    |       |
| nama_mhs   | varchar(50) | YES  |     | NULL    |       |
| alamat_mhs | varchar(50) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.073 sec)
```

```
MariaDB [akademik]> .
```

Gambar 7. 13 Fungsi DROP untuk Menghilangkan Komponen pada Tabel

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-04	1. Praktekkanlah langkah Langkah DDL 1-8	Jawaban hasil praktek (screenshot)	100

7.7 POST TEST

Jawablah pertanyaan-pertanyaan di bawah ini dimana setiap pertanyaan memiliki bobot nilai yang berbeda, dengan total nilai 100.

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-04	1. Buatlah sebuah database (bebas sesuai keinginan) - DDL 2. Database terdiri dari 2 buah tabel (nama dan jenis tabel bebas sesuai dengan database yang dirancang) 3. Pada masing-masing tabel , rancanglah 4 buah atribut dan tentukan primary keynya	Jawaban hasil praktek (Screenshot)	100
				Total Nilai	100

7.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-04	20%		
2.	Praktik	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-04	30%		
3.	Post-Test	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-04	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

PRAKTIKUM 8: DML DAN FUNGSI AGREGASI

Pertemuan ke : 8

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-03	Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas
CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPL-07	Mampu memilih, membuat dan menerapakan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPL-08	Mampu merancang dan mengimplementasikan algoritma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan perangkat lunak dan pemikiran komputasi
CPMK-04	Mahasiswa dapat memahami, menganalisa, dan mengimplementasikan Bahasa query basis data DDL dan DML
CPMK- 05	Mahasiswa dapat mengelola lingkungan basisdata dengan fungsi agregasi

8.1 DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas dengan menggunakan asesmen Pre Test untuk materi terkait DML dan Fungsi Agregasi.
2. Berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya melalui asesmen Pre test.

3. Mengkaji/menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah terkait DML dan Fungsi Agregasi dengan asesmen praktik dan post test.
4. Memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah terkait DML dan Fungsi Agregasi dengan Asesmen Post Test dan Praktik.
5. Merancang dan mengimplementasikan algoritma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan perangkat lunak dan pemikiran komputasi.

8.2 INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-03 CPL-04	CPMK-04	1. Kemampuan mahasiswa meningkat dalam memahami konsep data dan informasi serta dapat berpikir logis yang diterapkan dengan cara mempraktikkan konsep DDL dan DML
CPL-05 CPL-07 CPL-08	CPMK-04	2. Kemampuan mahasiswa meningkat dalam menganalisa, memilih dan mengimplementasikan masalah terkait DDL dan DML dengan menggunakan tool pemodelan yang tersedia
CPL-07 CPL-08	CPMK-05	3. Kemampuan mahasiswa meningkat dalam menerapkan teknik, penggunaan perangkat teknik dengan fungsi agregasi
		4. Kemampuan mahasiswa meningkat dalam merancang dan mengimplementasikan algoritma dengan menggunakan fungsi agregasi

8.3 TEORI PENDUKUNG

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

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

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 Klausus 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

8.4 HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer
2. Buku Petunjuk Praktikum
3. Aplikasi pemodelan: XAMPP
4. Browser (firefox, chrome)

8.5 PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-04	1. Jelaskan pengertian DML?	15
2.	CPL-04	CPMK-04	2. Jelaskan perbedaan DDL dan DML?	10
3.	CPL-05	CPMK-04	3. Jelaskan fungsi DML?	10
4.	CPL-07	CPMK-04	4. Berikan contoh minimal 3 dari perintah DML!	15
5.	CPL-08	CPMK-04		
6.	CPL-07 CPL-08	CPMK-05	5. Jelaskan maksud dari count, sum, min, max, dan avg dalam fungsi agregasi	20
7.	CPL-07 CPL-08	CPMK-05	6. Sebutkan dan jelaskan operator dalam MySQL	20
8.	CPL-07 CPL-08	CPMK-05	7. Apa maksud dari Query : SELECT SUM(nilai) AS Total FROM nilai;	10

8.6 LANGKAH PRAKTIKUM

1. 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 mahasiswa SET NIM = '18451', nama_mhs ='Nur Fatayah', alamat_mhs='Yogyakarta';
```

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 mahasiswa (NIM, nama_mhs, alamat_mhs) VALUES ('18451', 'Nur Fatayah', 'Yogyakarta');
```

Tanpa Menyebutkan Kolom

```
INSERT INTO nama_tabel VALUES (data_kolom_pertama, data_kolom_kedua,data_kolom_terakhir);
```

```
INSERT INTO mahasiswa VALUES ('18451', Nur Fatayah', 'Yogyakarta')
```

```
MariaDB [akademik]> insert into mahasiswa values ('18451', 'Nur Fatayah', 'Yogyakarta');  
Query OK, 1 row affected (0.340 sec)
```

```
MariaDB [akademik]> insert into mahasiswa values ('18324', 'Mahardika', 'Sleman');  
Query OK, 1 row affected (0.071 sec)
```

```
MariaDB [akademik]>
```

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 mahasiswa;
```

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 = '18451'

```
SELECT * from mahasiswa WHERE nim = '18451'
```

Melihat data mahasiswa yang bernama 'Nur Fatayah'

```
SELECT * from mahasiswa where nama_mhs = 'Nur Fatayah'
```

Melihat data mahasiswa yang beralamat di Sleman

```
SELECT * from mahasiswa 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.

```
MariaDB [akademik]> select * from mahasiswa;
+-----+-----+-----+
| NIM | nama_mhs | alamat_mhs |
+-----+-----+-----+
| 18324 | Mahardika | Sleman |
| 18451 | Nur Fatayah | Yogyakarta |
+-----+-----+-----+
2 rows in set (0.001 sec)

MariaDB [akademik]> select * from mahasiswa where nim = '18451';
+-----+-----+-----+
| NIM | nama_mhs | alamat_mhs |
+-----+-----+-----+
| 18451 | Nur Fatayah | Yogyakarta |
+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [akademik]> -
```

Gambar 8. 1 Perintah Tampil Data

3. Membatasi jumlah field yang dibaca

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

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

Jika urut menurun:

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

c) Menampilkan data berurutan

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

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

Jika urut menurun:

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

```
MariaDB [akademik]> select nim, nama_mhs from mahasiswa order by nim asc;
+-----+-----+
| nim | nama_mhs |
+-----+-----+
| 18090 | Sofia Ananta |
| 18102 | Rahma Nadia |
| 18115 | Dian Sastro |
| 18324 | Mahardika |
| 18451 | Nur Fatayah |
+-----+-----+
5 rows in set (0.049 sec)

MariaDB [akademik]> select nim, nama_mhs from mahasiswa order by nim desc;
+-----+-----+
| nim | nama_mhs |
+-----+-----+
| 18451 | Nur Fatayah |
| 18324 | Mahardika |
| 18115 | Dian Sastro |
| 18102 | Rahma Nadia |
| 18090 | Sofia Ananta |
+-----+-----+
5 rows in set (0.001 sec)

MariaDB [akademik]> -
```

Gambar 8. 2 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 ‘Nur Fatayah’ menjadi ‘N Fatayah’

```
UPDATE mahasiswa SET nama_mhs = " where nim = '18451';
```

```

MariaDB [akademik]> update mahasiswa set nama_mhs = 'N Fatayah' where nim='18451';
Query OK, 1 row affected (0.130 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [akademik]> select nim, nama_mhs from mahasiswa order by nim desc;
+-----+-----+
| nim | nama_mhs |
+-----+-----+
| 18451 | N Fatayah |
| 18324 | Mahardika |
| 18115 | Dian Sastro |
| 18102 | Rahma Nadia |
| 18090 | Sofia Ananta |
+-----+-----+
5 rows in set (0.001 sec)

MariaDB [akademik]>

```

Gambar 8. 3 Tampil Data Urut Abjad

- e) Menghapus isi tabel

```
DELETE FROM nama_tabel WHERE kondisi
```

Apabila tidak menggunakan klausa WHERE maka akan menyebabkan semua record dalam tabel terhapus. Misal akan dihapus data mahasiswa bernama ‘Sofia Ananta’

```
DELETE FROM mahasiswa WHERE nim = '18090'
```

```

MariaDB [akademik]> delete from mahasiswa where nim='18090';
Query OK, 1 row affected (0.139 sec)

MariaDB [akademik]> select * from mahasiswa;
+-----+-----+-----+
| NIM | nama_mhs | alamat_mhs |
+-----+-----+-----+
| 18102 | Rahma Nadia | Gunung Kidul |
| 18115 | Dian Sastro | Jakarta |
| 18324 | Mahardika | Sleman |
| 18451 | N Fatayah | Yogyakarta |
+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [akademik]>

```

Gambar 8. 4 Menghapus Isi Tabel

2. Fungsi Agregasi

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. 5 Tampil Seluruh Data

- Menampilkan data mata kuliah yang dilaksanakan di semester 1 SELECT * FROM mata_kuliah WHERE sem = 1;
- Menampilkan data mata kuliah yang dilaksanakan selain semester 1 SELECT * FROM mata_kuliah WHERE sem <>1;
- 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. 6 Tampil Data dengan Kondisi

- Menampilkan data mata kuliah yang mengandung judul informatika dan sksnya = 3
SELECT * FROM mat_kul WHERE nama_kul LIKE '%informatika%' AND sks = 3
- Menghitung jumlah data mata kuliah SELECT COUNT(*) from mat_kul;
- Menghitung sks yang paling sedikit, sks yang paling banyak dan rata-rata sks SELECT MIN (sks), MAX (sks), AVG (sks) from mat_kul;
- 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 | 2.833333333333333 |
+-----+-----+-----+
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. 7 Tampil Data dengan Kondisi SUM

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
----	-----	------	------------	-------------------	------

1.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-04	1. Praktekanlah langkah Langkah DML 1-5	Jawaban hasil praktek (Screenshot)	50
2.	CPL-07 CPL-08	CPMK-05	2. Praktekanlah langkah Fungsi Agregasi 1-7	Jawaban hasil praktek (Screenshot)	50

8.7 POST TEST

Jawablah pertanyaan-pertanyaan di bawah ini dimana setiap pertanyaan memiliki bobot nilai yang berbeda, dengan total nilai 100

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07	CPMK-01 CPMK-02 CPMK-03	1. Pada database yang telah dirancang, inputkan minimal 5 buah data pada masing-masing tabel . - DML 2. Tampilkan semua data yang telah diinputkan 3. Tampilkan data yang telah diinputkan sesuai dengan kondisi tertentu (gunakan perintah (WHERE)) 4. Perbarui salah satu data pada setiap tabel	Jawaban hasil praktek (Screenshot)	50
2.	CPL-07 CPL-08	CPMK-05	5. Jelaskan pengertian dari insert data , update data dan delete data? 6. Tuliskan syntax dari insert, update dan delete! 7. Tuliskan syntax untuk memasukan data sebagai berikut: (Nur Fatayah, 2000018451, nur18451@webmail.uad.ac.id)	Jawaban hasil	50
				Total Nilai	100

8.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	20%		

2.	Praktik	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	30%		
3.	Post-Test	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

PRAKTIKUM 9: RELASI TABEL DENGAN JOIN

Pertemuan ke : 9

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-03	Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas
CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPL-08	Mampu merancang dan mengimplementasikan algoritma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan perangkat lunak dan pemikiran komputasi
CPMK-05	Mahasiswa memahami, menganalisa dan menyimpulkan arsitektur organisasi dan pemodelan data (ITE-IMA-03 ACM 2017, ITE-IMA-07 ACM 2017)

9.1 DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas dengan menggunakan asesmen Pre Test untuk materi terkait JOIN Tabel

2. Berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya melalui asesmen Pre test.
3. Mengkaji/menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah terkait JOIN Tabel dengan asesmen praktik dan post test.
4. Memilih, membuat dan menerapakan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalahterkait JOIN Tabel dengan Asesmen Post Test dan Praktik.
5. Merancang dan mengimplementasikan algoritma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan perangkat lunak dan pemikiran komputasi.

9.2 INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-03 CPL-04	CPMK-05	1. Kemampuan mahasiswa meningkat dalam memahami konsep data dan informasi serta dapat berfikir logis yang diterapkan dengan cara mempraktikan konsep JOIN Tabel
CPL-05 CPL-07 CPL-08	CPMK-05	2. Kemampuan mahasiswa meningkat dalam menganalisa, memilih dan mengimplementasikan masalah terkait JOIN Tabel dengan menggunakan tool pemodelan yang tersedia

9.3 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

9.4 HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

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

9.5 PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-05	1. Jelaskan apa yang dimaksud dengan relasi database!	25
2.	CPL-04	CPMK-05	2. Jelaskan fungsi JOIN pada database!	25
3.	CPL-05	CPMK-05	3. Sebut dan jelaskan macam macam JOIN!	25
4.	CPL-07	CPMK-05	4. Berikan satu contoh relasi table untuk Inner JOIN,	25

5	CPL-08	CPMK-05	Full Outer JOIN, Left Outer JOIN, Right Outer JOIN, Union ?	
---	--------	---------	---	--

9.6 LANGKAH PRAKTIKUM

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

- Buatlah tabel mahasiswa, nilai, dan mata_kuliah seperti pada Gambar 9.1.

```
MariaDB [akademik]> select * from mahasiswa;
+-----+-----+-----+
| NIM  | nama_mhs    | alamat_mhs |
+-----+-----+-----+
| 18102 | Rahma Nadia | Gunung Kidul |
| 18115 | Dian Sastro | Jakarta      |
| 18210 | Aulia        | Kulon Progo  |
| 18321 | Fidia Fajri  | Sleman       |
| 18324 | Mahardika    | Sleman       |
| 18451 | N Fatayah    | Yogyakarta   |
+-----+-----+-----+
6 rows in set (0.065 sec)
```

```
MariaDB [akademik]> select * from nilai;
+-----+-----+-----+
| NIM  | kode_kul | nilai |
+-----+-----+-----+
| 18102 | IF0105   | 85   |
| 18102 | IF0104   | 90   |
| 18451 | IF0105   | 90   |
| 18451 | IF0206   | 85   |
| 18321 | IF0206   | 90   |
| 18321 | IF0104   | 80   |
+-----+-----+-----+
6 rows in set (0.001 sec)
```

```
MariaDB [akademik]> select * from mata_kuliah;
+-----+-----+-----+-----+
| kode_kul | nama_matkul          | sks  | sem  |
+-----+-----+-----+-----+
| IF0104   | Basis Data            | 4    | 3    |
| IF0105   | Pemrograman Web Dinamis | 2    | 5    |
| IF0206   | Statistika Informatika | 4    | 3    |
+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

Gambar 9. 1 Tabel Mahasiswa, Nilai, Mata kuliah

- Lakukan Inner Join

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

Select mahasiswa.NIM, mahasiswa.nama_mhs, nilai.kode_kul, nilai.nilai from mahasiswa, nilai Where mahasiswa.nim = nilai.nim;

```
MariaDB [akademik]> select mahasiswa.NIM, mahasiswa.nama_mhs, nilai.kode_kul, nilai.nilai from mahasiswa, nilai where mahasiswa.NIM = nilai.NIM;
+-----+-----+-----+-----+
| NIM | nama_mhs | kode_kul | nilai |
+-----+-----+-----+-----+
| 18102 | Rahma Nadia | IF0105 | 85 |
| 18102 | Rahma Nadia | IF0104 | 90 |
| 18451 | N Fatayah | IF0105 | 90 |
| 18451 | N Fatayah | IF0206 | 85 |
| 18321 | Fidia Fajri | IF0206 | 90 |
| 18321 | Fidia Fajri | IF0104 | 80 |
+-----+-----+-----+-----+
6 rows in set (0.042 sec)

MariaDB [akademik]> -
```

Gambar 9. 2 Perintah Inner Join 1

Dengan perintah inner join

```
SELECT mahasiswa.NIM, mahasiswa.nama_mhs, nilai.kode_kul, nilai.nilai FROM mahasiswa
INNER JOIN nilai ON mahasiswa.NIM = nilai.NIM;
```

```
MariaDB [akademik]> select mahasiswa.NIM, mahasiswa.nama_mhs, nilai.kode_kul, nilai.nilai
    -> from mahasiswa inner join nilai
    -> on mahasiswa.NIM = nilai.NIM;
+-----+-----+-----+-----+
| NIM | nama_mhs | kode_kul | nilai |
+-----+-----+-----+-----+
| 18102 | Rahma Nadia | IF0105 | 85 |
| 18102 | Rahma Nadia | IF0104 | 90 |
| 18451 | N Fatayah | IF0105 | 90 |
| 18451 | N Fatayah | IF0206 | 85 |
| 18321 | Fidia Fajri | IF0206 | 90 |
| 18321 | Fidia Fajri | IF0104 | 80 |
+-----+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [akademik]> -
```

Gambar 9. 3 Perintah Inner Join 2

Dua gambar di atas menghasilkan hasil yang sama. Terlihat bahwa pada tabel mahasiswa dan mata_kuliah, nim adalah atribut yang menghubungkan ke dua tabel tersebut. Pada perintah di atas hanya record yang berpasangan yang dimunculkan. Pada tabel mahasiswa terdapat record dengan nim ‘18115’ dan ‘18210’ yang tidak terdapat pada tabel nilai, sehingga tidak dimunculkan.

3. Lakukan Left Join

Menampilkan semua data mahasiswa yang ada di tabel mahasiswa berikut data nilai mahasiswa
SELECT mahasiswa.NIM, mahasiswa.nama_mhs, nilai.kode_kul, nilai.nilai
FROM mahasiswa LEFT JOIN nilai on mahasiswa.NIM = nilai.NIM;

```
MariaDB [akademik]> select mahasiswa.NIM, mahasiswa.nama_mhs, nilai.kode_kul, nilai.nilai
    -> from mahasiswa left join nilai on mahasiswa.NIM = nilai.NIM;
+-----+-----+-----+-----+
| NIM | nama_mhs | kode_kul | nilai |
+-----+-----+-----+-----+
| 18102 | Rahma Nadia | IF0105 | 85 |
| 18102 | Rahma Nadia | IF0104 | 90 |
| 18451 | N Fatayah | IF0105 | 90 |
| 18451 | N Fatayah | IF0206 | 85 |
| 18321 | Fidia Fajri | IF0206 | 90 |
| 18321 | Fidia Fajri | IF0104 | 80 |
| 18115 | Dian Sastro | NULL | NULL |
| 18210 | Aulia | NULL | NULL |
| 18324 | Mahardika | NULL | NULL |
+-----+-----+-----+-----+
9 rows in set (0.041 sec)

MariaDB [akademik]>
```

Gambar 9. 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.

4. Lakukan Right Join

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

```
SELECT mahasiswa.NIM, mahasiswa.nama_mhs, nilai.kode_kul, nilai.nilai
FROM mahasiswa RIGHT JOIN nilai ON mahasiswa.NIM = nilai.NIM;
```

```
MariaDB [akademik]> select mahasiswa.NIM, mahasiswa.nama_mhs, nilai.kode_kul, nilai.nilai
-> from mahasiswa right join nilai on mahasiswa.NIM = nilai.NIM;
+-----+-----+-----+
| NIM | nama_mhs | kode_kul | nilai |
+-----+-----+-----+
| 18102 | Rahma Nadia | IF0105 | 85 |
| 18102 | Rahma Nadia | IF0104 | 90 |
| 18451 | N Fatayah | IF0105 | 90 |
| 18451 | N Fatayah | IF0206 | 85 |
| 18321 | Fidia Fajri | IF0206 | 90 |
| 18321 | Fidia Fajri | IF0104 | 80 |
+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [akademik]> -
```

Gambar 9. 5 Perintah Right Join

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

5. Lakukan Union

```
SELECT * FROM mahasiswa UNION SELECT * FROM nilai;
```

Perintah tersebut akan menghasilkan table yang berisi penggabungan antara table mahasiswa dan nilai seperti pada gambar 9.6.

```
MariaDB [akademik]> select * from mahasiswa union select * from nilai;
+-----+-----+-----+
| NIM | nama_mhs | alamat_mhs |
+-----+-----+-----+
| 18102 | Rahma Nadia | Gunung Kidul |
| 18115 | Dian Sastro | Jakarta |
| 18210 | Aulia | Kulon Progo |
| 18321 | Fidia Fajri | Sleman |
| 18324 | Mahardika | Sleman |
| 18451 | N Fatayah | Yogyakarta |
| 18102 | IF0105 | 85 |
| 18102 | IF0104 | 90 |
| 18451 | IF0105 | 90 |
| 18451 | IF0206 | 85 |
| 18321 | IF0206 | 90 |
| 18321 | IF0104 | 80 |
+-----+-----+-----+
12 rows in set (0.001 sec)

MariaDB [akademik]> -
```

Gambar 9. 6 Perintah Union

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
----	-----	------	------------	-------------------	------

1.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	Praktekkan Langkah 1 membuat tabel	Jawaban hasil praktek (Screenshot)	30
2.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	Praktekkan Langkah 2-5 melakukan inner join, Right join, left join, Union	Jawaban hasil praktek (Screenshot)	70

9.7 POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	1. Buatlah sebuah database (bebas sesuai keinginan) 2. Database terdiri dari 2 buah tabel (nama dan jenis tabel bebas sesuai dengan database yang dirancang) 3. Isilah data pada masing masing table minimal 5 data 4. Gunakan innerjoin, rightjoin, leftjoin, sesuai dengan table yang telah dibuat tersebut!	100

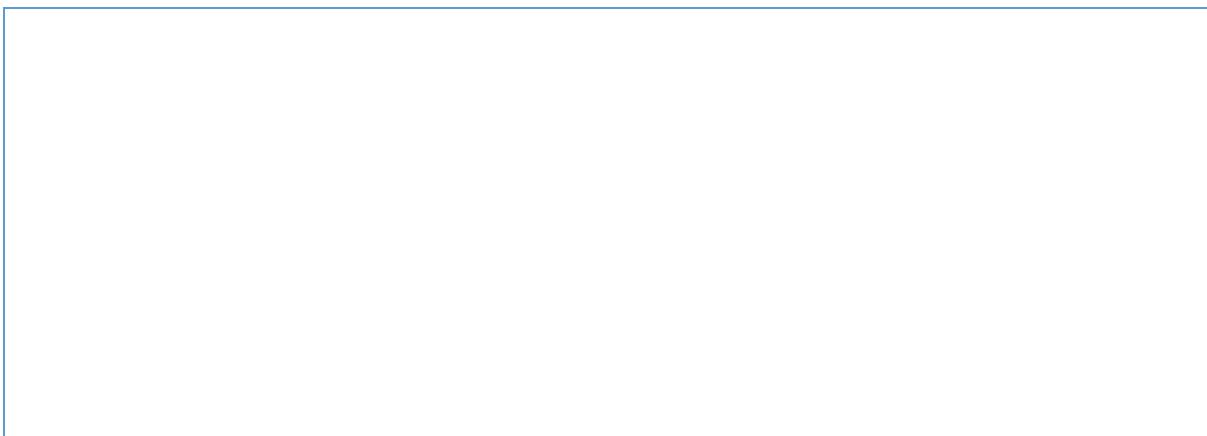
9.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03 CPL-05	CPMK-05	20%		
		CPL-08				
2.	Praktik	CPL-03	CPMK-05	30%		
3.	Post-Test	CPL-07	CPMK-05	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-----------------	----------------------------	--------------------



PRAKTIKUM 10: SUB QUERY, FUNCTION, STORE PROCEDURE

Pertemuan ke : 10

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-03	Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas
CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPL-07	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPL-08	Mampu merancang dan mengimplementasikan algoritma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan perangkat lunak dan pemikiran komputasi
CPMK-05	Mahasiswa memahami, menganalisis dan menyimpulkan arsitektur organisasi dan pemodelan data (ITE-IMA-03 ACM 2017, ITE-IMA-07 ACM 2017)

10.1 DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu menerapkan:

1. Menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas dengan menggunakan asesmen Pre Test untuk materi terkait SUBQUERY (IN, NOT IN, EXIST, Operator Perbandingan).
2. Berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya melalui asesmen Pre test.
3. Mengkaji/menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah terkait SUBQUERY (IN, NOT IN, EXIST, Operator Perbandingan) dengan asesmen praktik dan post test.
4. Memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah terkait SUBQUERY (IN, NOT IN, EXIST, Operator Perbandingan) dengan Asesmen Post Test dan Praktik.
5. Merancang dan mengimplementasikan algoritma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan perangkat lunak dan pemikiran komputasi.

10.2 INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-03 CPL-04	CPMK-05	1. Kemampuan mahasiswa meningkat dalam memahami konsep data dan informasi serta dapat berpikir logis yang diterapkan dengan cara mempraktekkan konsep SUBQUERY (IN, NOT IN, EXIST, Operator Perbandingan)
CPL-05 CPL-07 CPL-08	CPMK-05	2. Kemampuan mahasiswa meningkat dalam menganalisa, memilih dan mengimplementasikan masalah SUBQUERY (IN, NOT IN, EXIST, Operator Perbandingan) dengan menggunakan tool pemodelan yang tersedia

10.3 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.

10.4 HARDWARE DAN SOFTWARE

Alat dan bahan yang digunakan dalam praktikum ini yaitu:

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

10.5 PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-05	1. Jelaskan apa yang dimaksud dengan SUBQUERY!	25
2.	CPL-04	CPMK-05	2. Jelaskan fungsi SUBQUERY pada database!	25
3.	CPL-05	CPMK-05	3. Sebutkan dan jelaskan macam macam perintah SUBQUERY!	25
4.	CPL-07	CPMK-05	4. Berikan dua contoh penerapan SUBQUERY dalam basis data?	25
5	CPL-08	CPMK-05		

10.6 LANGKAH PRAKTIKUM

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

1. Buatlah dan Isikan tabel nilai seperti Gambar 10.1:

```
CREATE TABLE <nama_table> (<nama_kolom><tipedata>);
```

NIM	kode_kul	nilai
18102	IF0105	85
18102	IF0104	90
18451	IF0105	90
18451	IF0206	85
18321	IF0206	90
18321	IF0104	80

6 rows in set (0.001 sec)

Gambar 10. 1 Nilai Mahasiswa

2. Menampilkan data nilai mahasiswa yang nilainya melebihi rata-rata nilai mata kuliah secara keseluruhan (AVG)

Langkahnya:

- 1) 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.
- 2) Dari Sub Query tersebut akan dimasukkan dalam Query utama (Main Query). Main Query ini untuk mencari nilai yang memiliki nilai di atas rata rata.
- 3) Lihat gambar 10. 2 berikut ini:

Sub Query: SELECT avg(nilai) from nilai;

```
MariaDB [akademik]> select avg(nilai) from nilai;
+-----+
| avg(nilai) |
+-----+
| 86.6667 |
+-----+
1 row in set (0.001 sec)

MariaDB [akademik]> -
```

Gambar 10. 2 Sub Query

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

```
MariaDB [akademik]> select NIM, nilai from nilai where nilai > (select avg(nilai) from nilai);
+-----+-----+
| NIM | nilai |
+-----+-----+
| 18102 | 90 |
| 18451 | 90 |
| 18321 | 90 |
+-----+-----+
3 rows in set (0.005 sec)

MariaDB [akademik]>
```

Gambar 10. 3 Main Query

3. Carilah data nilai yang nilainya sama dengan nilai terbesar (MAX)

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.

Langkahnya:

- Tampilkan Tabel Nilai

```
SELECT * FROM nilai;
```

```
MariaDB [akademik]> select * from nilai;
+-----+-----+-----+
| NIM | kode_kul | nilai |
+-----+-----+-----+
| 18102 | IF0105 | 85 |
| 18102 | IF0104 | 90 |
| 18451 | IF0105 | 90 |
| 18451 | IF0206 | 85 |
| 18321 | IF0206 | 90 |
| 18321 | IF0104 | 80 |
+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [akademik]> -
```

Gambar 10. 4 Tabel Nilai

- Tampilkan nilai maksimal dengan menggunakan Perintah MAX

```
SELECT NIM, nilai, kode_kul
```

```
FROM nilai WHERE nilai=(SELECT MAX(nilai) FROM nilai)
```

```
MariaDB [akademik]> select NIM, nilai, kode_kul
    -> from nilai where nilai=(select max(nilai) from nilai);
+-----+-----+-----+
| NIM   | nilai  | kode_kul |
+-----+-----+-----+
| 18102 |    90   | IF0104  |
| 18451 |    90   | IF0105  |
| 18321 |    90   | IF0206  |
+-----+-----+-----+
3 rows in set (0.001 sec)

MariaDB [akademik]>
```

Gambar 10. 5 Tabel hasil Query tanpa Limit

c. Gunakan Perintah ORDER BY dan LIMIT

Query di atas menampilkan data nilai lebih dari 1 baris ketika baris yang nilainya sama dengan nilai MAX(nilai) lebih dari 1 baris. Namun pada Query menggunakan ORDER BY dan LIMIT akan menampilkan 1 baris saja karena ada penggunaan LIMIT yang dapat dilihat pada gambar 10.6.

Kekurangan dari SQL ini adalah ketika ada data nilai yang sama-sama memiliki nilai sama dengan MAX(nilai) lebih dari 1 baris namun tidak ditampilkan.

SELECT nim, nilai, kode_kul

FROM nilai ORDER BY nilai ASC LIMIT 1

```
MariaDB [akademik]> select NIM, nilai, kode_kul
    -> from nilai order by nilai asc limit 1;
+-----+-----+-----+
| NIM   | nilai  | kode_kul |
+-----+-----+-----+
| 18321 |    80   | IF0104  |
+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [akademik]> -
```

Gambar 10. 6 Tabel hasil Query dengan Limit

4. Menampilkan data mahasiswa yang mengambil mata kuliah ‘IF0104’ dengan IN

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

SELECT NIM, nama_mhs FROM mahasiswa

WHERE nim IN (SELECT nim FROM kuliah WHERE kode_kul = ‘IF0104’)

```
MariaDB [akademik]> select NIM, nama_mhs
    -> from mahasiswa
    -> where NIM in (select nim from nilai where kode_kul='IF0104');
+-----+-----+
| NIM   | nama_mhs  |
+-----+-----+
| 18102 | Rahma Nadia |
| 18321 | Fidia Fajri |
+-----+-----+
2 rows in set (0.001 sec)

MariaDB [akademik]> -
```

Gambar 10. 7 Tabel hasil Query perbandingan dua table dengan IN

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

5. Menampilkan data mahasiswa yang tidak mengambil mata kuliah basis data dengan NOT IN

SELECT NIM, nama_mhs from mahasiswa

WHERE NIM NOT IN (SELECT NIM FROM nilai WHERE kode_kul = 'IF0104'

```
MariaDB [akademik]> select NIM, nama_mhs from mahasiswa
    -> where NIM not in (select NIM from nilai where kode_kul = 'IF0104');
+-----+-----+
| NIM   | nama_mhs |
+-----+-----+
| 18115 | Dian Sastro |
| 18210 | Aulia        |
| 18324 | Mahardika    |
| 18451 | N Fatayah    |
+-----+-----+
4 rows in set (0.087 sec)

MariaDB [akademik]>
```

Gambar 10. 8 Tabel hasil Query Perbandingan dua Tabel dengan NOT IN

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	Praktekkan Langkah 1 membuat tabel	Jawaban hasil praktek (Screenshot)	30
2.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	Praktekkan Langkah 2-5 melakukan AVG, MAX, IN, NOT IN	Jawaban hasil praktek (Screenshot)	70

10.7 POST TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	<ol style="list-style-type: none"> Buatlah sebuah database (bebas sesuai keinginan) Database terdiri dari 2 buah tabel (nama dan jenis tabel bebas sesuai dengan database yang dirancang) Isilah data pada masing masing table minimal 5 data Gunakan perintah SUBQUERY MIN, <=, >=, AVG sesuai dengan table yang telah dibuat tersebut! 	Jawaban hasil praktek disamping (Screenshot)	100

10.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	20%		
2.	Praktik	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	30%		
3.	Post-Test	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

PRAKTIKUM 11: TRIGGER DAN KEAMANAN DATA

Pertemuan ke : 11

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-03	Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas
CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskiripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPL-07	Mampu memilih, membuat dan meneíapkan teknik, sumbeí daya, penggunaan peíangkat teknik modeín dan implementasi teknologi infoímasi untuk memecahkan masalah
CPL-08	Mampu meíancang dan mengimplementasikan algoítma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan peíangkat lunak dan pemikiían komputasi
CPMK-05	Mahasiswa memahami, menganalisis dan menyimpulkan arsitektur organisasi dan pemodelan data (ITE-IMA-03 ACM 2017, ITE-IMA-07 ACM 2017)

11.1 DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu memahami:

1. Menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas dengan menggunakan asesmen Pre Test untuk materi TRIGGER.
2. Berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya melalui asesmen Pre test.

3. Mengkaji/menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah terkait TRIGGER dengan asesmen praktik dan post test.
4. Memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah terkait TRIGGER dengan Asesmen Post Test dan Praktik.
5. Merancang dan mengimplementasikan algoritma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan perangkat lunak dan pemikiran komputasi.

11.2 INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-03 CPL-04	CPMK-05	1. Kemampuan mahasiswa meningkat dalam memahami konsep data dan informasi serta dapat berfikir logis yang diterapkan dengan cara mempraktikkan konsep TRIGGER
CPL-05 CPL-07 CPL-08	CPMK-05	2. Kemampuan mahasiswa meningkat dalam menganalisa, memilih dan mengimplementasikan masalah terkait TRIGGER dengan menggunakan tool pemodelan yang tersedia

11.3 TEORI PENDUKUNG

1. Trigger

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 statement (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 pendefinisan 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 dimasukan ke dalam table.

AFTER INSERT – dijalankan setelah data masuk ke dalam table.

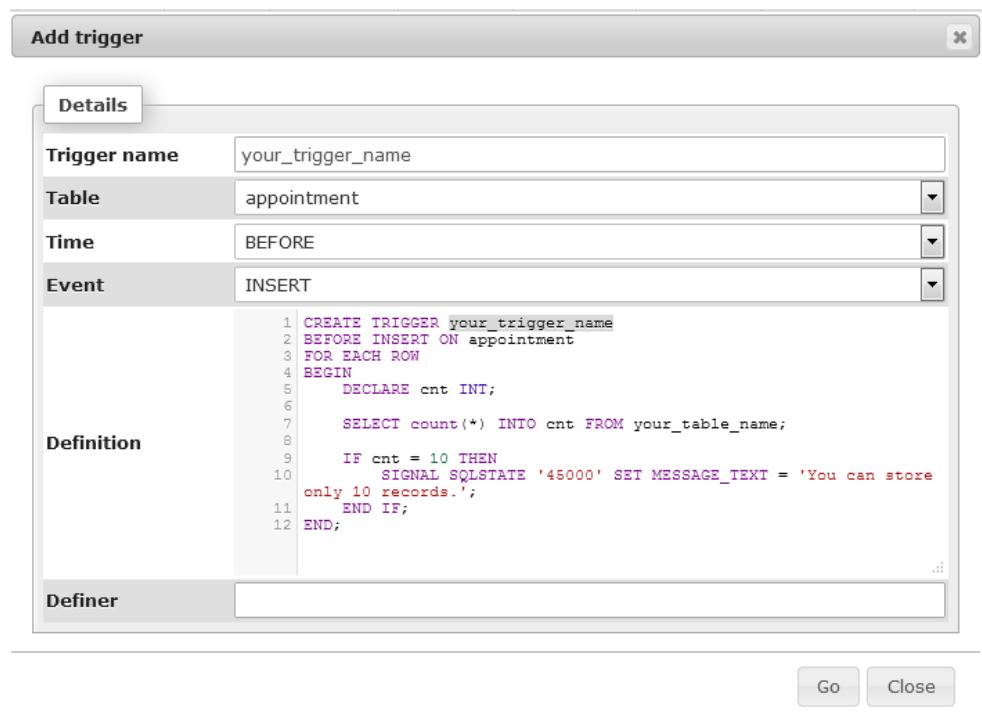
BEFORE UPDATE – dijalankan sebelum proses update data.

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 pendefinisan trigger diperlihatkan sebagai berikut:



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.

2. Keamanan Database

Database merupakan sesuatu yang sangat penting dan harus dijaga keberadaannya. Pada bagian ini akan membahas bagaimana agar keamanan database tetap terjaga untuk mengantisipasi hal-hal yang tidak diinginkan.

Pentingnya Keamanan

Sebagai seorang DBA, setidaknya Anda sudah memahami seperti apa database itu, dan sejauh mana keamanan yang diperlukan. Juga yang perlu Anda ketahui, kebanyakan software database yang ada di pasaran belum memiliki security secara khusus. Anda dapat mencarinya di internet, baik itu yang komersil maupun yang didistribusikan secara gratis.

Mengamankan Database dari Crack

Salah satu cara terbaik untuk mengamankan database dari gangguan crack yaitu menanggulangi adanya penyerang atau cracker dengan mengamankan database sistem mysql. Diantaranya yaitu:

a. Backup Database Sistem

Data terpenting pada MySQL terletak pada database sistem bernama mysql. Untuk menjaga dari segala kemungkinan, baik itu kerusakan atau gangguan dari cracker, Anda perlu memiliki file backup-nya. Sehingga jika sewaktu-waktu diperlukan, Anda tinggal merestore ke database aslinya.

b. Proteksi Database Sistem

Cara mengamankan database sistem, yaitu dengan memberikan perintah GRANT. Dengan demikian, client hanya dapat mengakses saja tanpa bisa melakukan modifikasi. Hal ini perlu di terapkan pada database server. Selain itu, jika perlu Anda dapat men-disable perintah SHOW DATABASES sehingga client tidak dapat menampilkan database yang terdapat pada MySQL dengan mengaktifkan perintah “skip_show_database”.

Untuk itu lakukan perintah seperti berikut:

1. Buka Command Prompt as Administrator dan Matikan MySQL Server dengan perintah: NET STOP MySQL.
2. Kemudian aktifkan skip_show_database, dengan menggunakan perintah berikut: mysqld –skip-show-database
3. Untuk memastikannya, masuk pada prompt mysql kemudian berikan perintah: SHOW VARIABLES;
4. Pada tampilan yang ada carilah bagian yang berisi:

skip_show_database	ON
--------------------	----

5. Setelah mendapatkan skip_show_database sudah aktif, lalu keluar dari prompt mysql.
6. Jalankan kembali MySQL server seperti biasanya.
7. Selanjutnya untuk melihat hasilnya, lakukan perintah “SHOW DATABASES;” dengan menggunakan user biasa dan akan keluar hasil bahwa show databases tidak diperbolehkan pada user tamu.

Keamanan Transaksi Data

Transaksi data merupakan proses pertukaran data antara client dengan server. Normalnya MySQL tidak menggunakan enkripsi ketika melakukan koneksi, karena hal ini dapat menyebabkan proses koneksi cline/server menjadi lambat. Hal ini juga yang menyebabkan mengapa MySQL begitu cepat dalam proses query. Meski demikian, dengan lemahnya transaksi data segala kemungkinan bisa terjadi. Misalnya saja, penyerangan yang dilakukan oleh cracker memungkinkan dapat membuka password di antara lalu lintas hubungan client ke server.

Untuk itu, jika komunikasi yang terjadi antara client dan server tersebut menggunakan jaringan yang kurang bagus, sebaiknya menggunakan komunikasi data dalam bentuk terenkripsi.

OpenSSL

OpenSSL (Open Secure Socket Layer) merupakan suatu proyek pengembangan protokol SSL (Secure Socket Layer) yang didasarkan pada library SSLeay.

SSL merupakan protokol yang digunakan untuk melakukan algoritma enkripsi yang menjamin keamanan data. Protokol ini menggunakan verifikasi X509, yakni suatu standar identitas yang sering digunakan pada internet.

Sejak versi 4.0 ini, MySQL memiliki program internal yang dapat digunakan untuk mengecek status OpenSSL. Sehingga Anda dapat mengetahui apakah protokol tersebut sudah aktif atau belum. Untuk itu lakukan perintah seperti berikut:

```
[cmd] XAMPP for Windows - mysql -u root
MariaDB [(none)]> show variables like "have_openssl";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_openssl | NO   |
+-----+-----+
1 row in set (0.115 sec)

MariaDB [(none)]> -
```

Pada gambar diatas terlihat bahwa OpenSSL belum diaktifkan. Untuk itu perlu untuk mencari program tersebut dan bisa diakses melalui alamat <https://www.openssl.org/>. Jika Anda menggunakan koneksi tersebut, maka semua koneksi client server dengan TCP/IP akan dilakukan dalam bentuk terenkripsi dan keamanan transaksi data akan terjamin.

Privilege System

Dalam pemberian izin akses privilege system, hendaknya Anda sedikit saja melakukan perlu berhati-hati. Sekali Keteledoran, segala kemungkinan yang dapat merugikan database Anda bisa terjadi.

Berikut ini beberapa tips yang berhubungan dengan privilege system:

1. Gunakan selalu option IDENTIFIED BY PASSWORD ketika Anda menambahkan user baru dengan perintah GRANT.
2. Sebaiknya jangan berikan izin akses PROCESS privilege kepada semua user, karena tampilan perintah seperti mysqladmin processlist akan menampilkan proses aktif eksekusi query. Hal ini memungkinkan user lain yang dapat menjalankan perintah tersebut untuk melihat user lain ketika melakukan perintah penting seperti UPDATE password.
3. Jangan berikan izin akses FILE privilege kepada semua user, hal ini dapat dimanfaatkan user untuk memasukkan file ke dalam database sistem. Di mana file tersebut dapat di-setting readable sehingga dapat dibaca setiap user, dan Anda tidak dapat melakukan overwrite pada file tersebut.

11.4 HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

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

11.5 PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-05	1. Jelaskan Konsep TRIGGER dalam basisdata?	25
2.	CPL-04	CPMK-05	2. Sebutkan dan jelaskan 3 contoh Event dalam TRIGGER?	25
3.	CPL-05	CPMK-05	3. Jelaskan fungsi TRIGGER dalam basisdata?	25
4.	CPL-07	CPMK-05	4. Berikan contoh minimal 2 penggunaan TRIGGER dalam basisdata?	25
5.	CPL-08	CPMK-05		

11.6 LANGKAH PRAKTIKUM

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

LANGKAH-LANGKAH

- Buatlah dan Isikan tabel produk dan table log_harga seperti tabel 11.1 dan 11.2:

Tabel 11. 1 Tabel Produk

```
MariaDB [penjualan]> desc produk;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| kode_produk | varchar(6) | NO | PRI | NULL |
| nama_produk | varchar(100) | YES | | NULL |
| harga | int(11) | YES | | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.094 sec)

MariaDB [penjualan]>
```

Tabel 11. 2 Tabel Log Harga

```
MariaDB [penjualan]> desc log_harga;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| log_id | int(11) | NO | PRI | NULL |
| kode_produk | varchar(6) | YES | | NULL |
| harga_lama | int(11) | YES | | NULL |
| harga_baru | int(11) | YES | | NULL |
| waktu_perubahan | datetime | YES | | NULL |
+-----+-----+-----+-----+-----+
5 rows in set (0.017 sec)
```

- 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
    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 ('PR001', 'TIGA BULAN JAGO LARAVEL', 200000);
```

```
INSERT INTO `produk` VALUES ('PR002', 'SEMINGGU JAGO PHP MYSQL', 180000);
```

Setelah itu, cek isi table dengan perintah:

```
SELECT * FROM produk;
```

```
MariaDB [penjualan]> select * from produk;
+-----+-----+-----+
| kode_produk | nama_produk | harga |
+-----+-----+-----+
| PR001       | TIGA BULAN JAGO LARAVEL | 200000 |
| PR002       | SEMINGGU JAGO PHP MYSQL | 180000 |
+-----+-----+-----+
2 rows in set (0.001 sec)
```

```
MariaDB [penjualan]>
```

4. Kemudian lakukan UPDATE Data produk dengan tujuan mengganti harga PR001 dari 20000 menjadi 250000 dengan perintah.

```
UPDATE produk SET harga=250000 WHERE kode_produk='PR001'
```

Lakukan pengecekan terhadap perintah di atas dengan memberi perintah:

```
SELECT * FROM produk;
```

```
MariaDB [penjualan]> select * from produk;
+-----+-----+-----+
| kode_produk | nama_produk | harga |
+-----+-----+-----+
| PR001       | TIGA BULAN JAGO LARAVEL | 250000 |
| PR002       | SEMINGGU JAGO PHP MYSQL | 180000 |
+-----+-----+-----+
2 rows in set (0.001 sec)
```

```
MariaDB [penjualan]>
```

5. Kemudian lakukan pengecekan pada table log_harga dengan perintah:

```
SELECT * FROM log_harga.
```

Waktu terpasang setelah dilakukan perubahan kembali, yaitu dari harga 250000 menjadi 200000, kemudian perubahan akan tercatat di dalam tabel log_harga.

```
MariaDB [penjualan]> select * from log_harga;
+-----+-----+-----+-----+
| log_id | kode_produk | harga_lama | harga_baru | waktu_perubahan |
+-----+-----+-----+-----+
|      0 | PR001       |    250000 |    200000 | 2023-08-21 11:43:24 |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

```
MariaDB [penjualan]>
```

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;
```

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	Praktekanlah langkah Langkah 1-7	Jawaban hasil praktek (Screenshot)	100

11.7 POST TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	Silahkan lakukan percobaan dengan menggunakan table yang sudah ada dengan perintah Trigger: BEFORE DELETE – dijalankan sebelum proses delete data. AFTER DELETE – dijalankan setelah proses delete data	Jawaban hasil praktek disamping (Screenshot)	100

11.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	20%		
2.	Praktik	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	30%		

3.	Post-Test	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	50%		
Total Nilai						

LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

PRAKTIKUM 12: TREND DBMS

Pertemuan ke : 12

Total Alokasi Waktu : 180 menit

- Materi : 30 menit
- Pre-Test : 30 menit
- Praktikum : 60 menit
- Post-Test : 60 menit

Total Bobot Penilaian : 100%

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

Pemenuhan CPL dan CPMK:

CPL-03	Mampu menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas
CPL-04	Mampu berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya
CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskiripsi saintifik hasil kajian untuk pemecahan masalah dengan mempeitimbangkan multidisiplin ilmu
CPL-07	Mampu memilih, membuat dan meneíapkan teknik, sumbeí daya, penggunaan peíangkat teknik modeín dan implementasi teknologi infoímasi untuk memecahkan masalah
CPL-08	Mampu meíancang dan mengimplementasikan algoítma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan peíangkat lunak dan pemikián komputasi
CPMK-05	Mahasiswa memahami, menganalisis dan menyimpulkan arsitektur organisasi dan pemodelan data (ITE-IMA-03 ACM 2017, ITE-IMA-07 ACM 2017)

12.1 DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu memahami:

1. Menerapkan konsep teoritis bidang area Informatika terkait matematika dasar dan ilmu komputer untuk memodelkan masalah dan meningkatkan produktivitas dengan menggunakan asesmen Pre Test untuk materi TRIGGER.
2. Berpikir logis, kritis, sistematis dan inovatif, dan mampu mengambil keputusan secara tepat di bidang keahliannya melalui asesmen Pre test.
3. Mengkaji/menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah terkait TRIGGER dengan asesmen praktik dan post test.
4. Memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah terkait TRIGGER dengan Asesmen Post Test dan Praktik.
5. Merancang dan mengimplementasikan algoritma/metode dalam mengidentifikasi dan memecahkan masalah yang melibatkan perangkat lunak dan pemikiran komputasi

12.2 INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator Ketercapaian diukur dengan:

CPL-03CPL-04	CPMK-01	1. Kemampuan mahasiswa meningkat dalam memahami konsep data dan informasi serta dapat berpikir logis yang diterapkan dengan cara menggunakan dbms NoSQL
CPL-05 CPL-07 CPL-08	CPMK-05	2. Kemampuan mahasiswa meningkat dalam menganalisa, memilih dan mengimplementasikan masalah terkait membangun database dengan menggunakan tool pemodelan yang tersedia

12.3 TEORI PENDUKUNG

MongoDB adalah salah satu jenis sistem manajemen basis data (DBMS) yang mendukung basis data NoSQL. MongoDB memiliki beberapa karakteristik unik yang membuatnya populer di kalangan pengembang perangkat lunak. Di bawah ini adalah beberapa informasi penting tentang MongoDB:

1. NoSQL Database: MongoDB adalah basis data NoSQL, yang berarti ia tidak mengikuti model relasional tradisional seperti yang digunakan oleh basis data SQL. Sebaliknya, ia menggunakan model data dokumen yang fleksibel, yang sering kali menggunakan format JSON.
2. Skema Fleksibel: Salah satu fitur utama MongoDB adalah kemampuannya untuk bekerja dengan skema dokumen yang fleksibel. Ini berarti Anda tidak harus mendefinisikan struktur tabel terlebih dahulu. Setiap dokumen dalam basis data dapat memiliki struktur yang berbeda, yang memungkinkan Anda mengubah skema aplikasi Anda dengan mudah.
3. Dukungan untuk Data Semi-Struktur: MongoDB sangat cocok untuk data semi-struktur atau tidak terstruktur, seperti data JSON yang dapat bervariasi dalam atribut yang ada. Ini membuatnya berguna untuk aplikasi seperti media sosial, e-commerce, dan sensor IoT.
4. Indeks: MongoDB mendukung pembuatan indeks yang memungkinkan pencarian dan kueri data dengan cepat. Anda dapat membuat indeks pada salah satu atau beberapa bidang dalam dokumen.
5. Skalabilitas Horizontal: MongoDB dirancang untuk mendukung skalabilitas horizontal yang mudah. Ini berarti Anda dapat menambahkan server tambahan untuk menangani beban yang lebih besar dengan mudah, membuatnya cocok untuk aplikasi yang memerlukan pertumbuhan cepat.
6. Replikasi dan Failover: MongoDB mendukung replikasi data untuk meningkatkan ketersediaan. Anda dapat mengkonfigurasi replikasi untuk melindungi data Anda dari kegagalan server. MongoDB juga mendukung otomatisasi failover, sehingga jika satu server gagal, yang lain akan mengambil alih.

7. Aggregation Framework: MongoDB memiliki kerangka kerja agregasi yang kuat yang memungkinkan Anda melakukan analisis dan pengolahan data yang kompleks di dalam basis data.

8. Dukungan untuk Geospatial Data: MongoDB memiliki dukungan khusus untuk data geospasial, yang memungkinkan Anda untuk menyimpan, mengindeks, dan mengquery data yang terkait dengan lokasi geografis.

9. Komunitas Besar dan Ekosistem: MongoDB memiliki komunitas pengguna yang besar dan ekosistem alat dan layanan terkait yang kuat. Ini membuatnya mudah untuk menemukan dukungan dan sumber daya untuk mengembangkan aplikasi dengan MongoDB.

MongoDB adalah salah satu dari banyak pilihan basis data NoSQL yang tersedia, dan pemilihan basis data harus didasarkan pada kebutuhan spesifik proyek Anda. Ini sangat cocok untuk aplikasi yang membutuhkan fleksibilitas skema dan skala horizontal yang cepat.

MongoDB merupakan database open source berbasis dokumen (Document-Oriented Database) yang awalnya dibuat dengan bahasa C++. MongoDB sendiri sudah dikembangkan oleh 10gen sejak Oktober 2007, namun baru dipublikasikan pada Februari 2009. Selain karena performanya 4 kali lebih cepat dibandingkan MySQL serta mudah diaplikasikan, karena telah tergabung juga sebagai modul PHP.

Dalam konsep MongoDB tidak ada yang namanya tabel, kolom ataupun baris yang ada hanyalah collection (ibaratnya tabel), document (ibaratnya record). Data modelnya sendiri disebut BSON dengan struktur mirip dengan JSON. Strukturnya cukup mudah dibaca, contohnya seperti ini.

```
{
  "nama" : "budi",
  "kontak" : {
    "alamat" : "Jl. Mawar No. 15",
    "kota" : "Jakarta",
    "kodepos" : "12345",
    "telp" : "081234567"
  }
}
```

Dengan konsep key-value yang ada pada MongoDB, setiap document otomatis memiliki index id yang unik. Hal ini membantu mempercepat proses pencarian data secara global.

Ada beberapa istilah di MongoDB yang mirip dengan database relasional seperti MySQL antara lain

Database SQL (MySQL)	Database MongoDB
Database	Database
Table	Collection
Field / Column	Field

Row	Document
-----	----------

12.4 HARDWARE DAN SOFTWARE

1. Komputer.
2. Sistem operasi windows 10 64 bit
3. MongoDB versi 2.6.4

12.5 PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01	1. Jelaskan Konsep Monggodb dalam basis data?	25
2.	CPL-04	CPMK-01	2. Sebutkan dan jelaskan 5 karakteristik Monggodb	25
3.	CPL-05	CPMK-05	3. Apa yang dimaksud dengan Non SQL database?	25
4.	CPL-07	CPMK-05	4. Jelaskan yang dimaksud dengan konsep key value pada mongodb.	25
5.	CPL-08	CPMK-05		

12.6 LANGKAH PRAKTIKUM

1. Membuat & Menghapus Database

Pertama kita akan buat database terlebih dahulu..

Contoh kasusnya yaitu:

```
Nama database : kampus
Nama collection : mahasiswa
Nama field : nim, nama, alamat
```

Saat membuat database gunakan perintah

```
use nama_database
```

Membuat database kampus maka gunakan perintah

```
1 > use kampus
2 switched to db kampus
```

Database kampus akan terbuat tetapi tidak akan muncul jika ditampilkan semua list database karena di dalam database kampus tidak terdapat *document* apapun

Gunakan perintah db.mahasiswa.insert untuk menambahkan satu document dengan nama koleksi mahasiswa di database kampus

```
1 > db.mahasiswa.insert({ "nim": "21400200", "nama": "faqih",
"alamat": "bandung" })
2 WriteResult({ "nInserted" : 1 })
```

Dan list semua database dengan perintah

```
1 > show dbs
2 admin    0.000GB
3 config   0.000GB
4 kampus   0.000GB
5 local    0.000GB
```

Oke sekarang sudah ada database kampus di list database
Untuk menghapus database pastikan anda sudah masuk di dalam database yang ingin dihapus
Gunakan perintah `db.dropDatabase()` untuk menghapus database

```
> use kampus
switched to db kampus
> db.dropDatabase()
{ "dropped" : "kampus", "ok" : 1 }
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
```

2. Manajemen Collection

collection atau koleksi merupakan istilah table di MongoDB. Di pembahasan membuat database telah disinggung sekilas dalam membuat collection.

Untuk membuat koleksi kita dapat langsung lakukan operasi *insert* data yang didalamnya terdapat nama koleksi seperti saat menambah data mahasiswa
Atau kita bisainisialisasikan diawal dalam pembuatan *collection* dengan perintah

```
db.createCollection("nama koleksi")
```

Misal kita ingin membuat *collection* **mahasiswa** di dalam database **kampus** maka

```
1 > use kampus
2 switched to db kampus
3 { "ok" : 1 }
4 > show collections
5 > db.createCollection("mahasiswa")
6 mahasiswa
```

Sedangkan untuk menghapus collection gunakan perintah
`db.collection_name.drop()`

```
1 > db.mahasiswa.drop()
2 true
3
```

3. Insert, query, Update dan Delete

Seperti halnya MySQL yang menyediakan fungsi data manipulation language (DML), MongoDB juga bisa melakukan proses *insert*, *query*, *update* dan *delete*

#1 Insert

Untuk menambahkan document / record gunakan perintah

```
db.collection_name.insert(document)
```

Bentuk penulisan document adalah *key value* seperti penulisan **dictionary di Python**

Misal kita insert *nim*, *nama* dan *alamat* maka bentuk *key value*-nya adalah seperti ini

```
1 {
2     "nim": "21400200",
3     "nama": "faqih",
4     "alamat": "bandung"
5 }
```

Sehingga perintah insert document MongoDB menjadi seperti ini

```
1 > db.mahasiswa.insert(
2     {
3         "nim": "21400200",
4         "nama": "faqih",
5         "alamat": "bandung"
6     })
```

Atau jika ditulis dalam satu baris menjadi

```
1 > db.mahasiswa.insert({"nim": "21400200", "nama": "faqih",
2 "alamat": "bandung"})
```

Jika ingin insert lebih dari satu document gunakan array seperti ini

```
1 > db.mahasiswa.insert([
2     {
3         "nim": "21400200",
4         "nama": "faqih",
5         "alamat": "bandung"
6     },
7     {
8         "nim": "21400201",
9         "nama": "zaki",
10        "alamat": "padang",
11        "hobi": "coding"
12    }])
```

```
11      }
12  ])
```

Kita bisa menambahkan *field* untuk *document* yang berbeda karena fleksibilitas skema database NoSQL tidak perlu mendefinisikan nama *field* di awal seperti MySQL

#2 Query

Untuk melakukan menampilkan data gunakan perintah query seperti ini

```
db.collection_name.find()
```

Sedangkan jika ingin menampilkan secara cantik gunakan tambahan perintah pretty()

```
db.collection_name.find().pretty()
```

```
1 > db.mahasiswa.find()
2 { "_id" : ObjectId("5e27dc0736652fbbc379af33"), "nim" : "21400200", "nama" :
3 { "_id" : ObjectId("5e27dc0736652fbbc379af34"), "nim" : "21400201", "nama" :
4
5 > db.mahasiswa.find().pretty()
6 {
7     "_id" : ObjectId("5e27dc0736652fbbc379af33"),
8     "nim" : "21400200",
9     "nama" : "faqih",
10    "alamat" : "bandung"
11 }
12 {
13     "_id" : ObjectId("5e27dc0736652fbbc379af34"),
14     "nim" : "21400201",
15     "nama" : "zaki",
16     "alamat" : "padang",
17     "hobi" : "coding"
18 }
19
```

Jika ingin mencari *document* tertentu berdasarkan *field*-nya cukup masukkan *value* yang ingin dicari di dalam fungsi *find()*

```
1 > db.mahasiswa.find({"nama":"faqih"})
2 { "_id" : ObjectId("5e27dc0736652fbbc379af33"), "nim" :
"21400200", "nama" : "faqih", "alamat" : "bandung" }
```

#3 Update

Untuk melakukan perubahan document gunakan perintah

```
db.collection_name.update({nilai document lama}, {$set:{nilai
document baru}})
```

Nilai *document* lama = kondisi document yang ingin diupdate nilainya

Nilai *document* baru = Nilai baru untuk document yang ingin diupdate

Misal kita ingin merubah alamat "padang" menjadi "surabaya" maka perintah akan seperti ini

```

1 > db.mahasiswa.update({ "alamat": "padang"}, { $set: { "alamat": "surabaya" } })
2 WriteResult({ "nMatched": 1, "nUpserted": 0, "nModified": 1 })
3
4 > db.mahasiswa.find()
5 { "_id": ObjectId("5e27dc0736652fbbc379af33"), "nim": "21400200", "nama": ""
6 { "_id": ObjectId("5e27dc0736652fbbc379af34"), "nim": "21400201", "nama": ""
7

```

#4 Delete

Untuk delete document gunakan perintah

```
db.collection_name.remove(nilai dokumen)
```

nilai dokumen = kondisi nilai dokumen yang ingin dihapus

Misal kita akan menghapus document mahasiswa yang berasal dari bandung, maka

```
1 > db.mahasiswa.remove({ "alamat": "bandung" })
```

selamat anda telah sukses membuat database, collection dan melakukan manipulasi *document*

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	Praktekkanlah langkah Langkah 1-3	Jawaban hasil praktek (Screenshot)	100

12.7 POST TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	Silahkan lakukan percobaan dengan: 1. Melakukan proses tambah 5 data mahasiswa 2. melakukan proses tambah 1 kolom alamat pada tabel mahasiswa	Jawaban hasil praktek disamping (Screenshot)	100

12.8 HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-01	20%		
2.	Praktik	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	30%		
3.	Post-Test	CPL-03 CPL-04 CPL-05 CPL-07 CPL-08	CPMK-05	50%		
Total Nilai						

DAFTAR PUSTAKA

1. <https://www.mysql.com/why-mysql/> (diakses 10 September 2023)
2. Oracle, 2019, Oracle academy Modul, Database Foundations
3. https://repository.dinus.ac.id/docs/ajar/9.materi_ERD_.pdf (diakses 5 September 2023)
4. <https://ngodingdata.com/tutorial-mongodb-cara-membuat-database-dan-collection/> (diakses 3 September 2023)
5. <https://elektro.um.ac.id/wp-content/uploads/2016/04/Basis-Data-Modul-6-Subquery.pdf> (diakses 7 September 2023)
6. <https://elektro.um.ac.id/wp-content/uploads/2016/04/Basis-Data-Modul-7-Stored-Procedure.pdf> (Diakses 7 September 2023)
7. https://bookdown.org/moh_rosidi2610/panduan_access/relasi.html (diakses 7 September 2023)
8. <https://kuliah.brigidaarie.com/wp-content/uploads/2018/02/Basis-Data-8.pdf> (Diakses 7 September 2023)
9. Silberschatz, A., H.F. Korth, and S. Sudarshan. 2019. Database System Concepts (7th Edition). McGraw-Hill.
10. Elmasri, R., and S. Navathe. 2016. Fundamentals of database systems 7th Edition. PEARSON



**LABORATORIUM
S1 INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN**



2023