LAPORAN PRAKTIKUM
BASIS DATA
TAHUN AKADEMIK 2025/2026

Disusun Oleh :

Nama : Mochammad Ferdyansyah
NIM : 251080200060
Kelompok : 6

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO

LEMBAR PERSETUJUAN

Telah Diperiksa Dan Disetujui
Isi Laporan Ini

LAPORAN PRAKTIKUM
BASIS DATA

Disusun Oleh:

Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 6

Mengetahui,
Laboran Informatika

(**Melina Atikawati, S.Kom.**)

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 - 2026

KATA PENGANTAR

Alhamdulillah segala puji syukur atas kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penyusun dapat menyelesaikan Laporan Praktikum Basis Data ini tanpa halangan yang berarti.

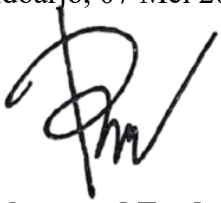
Keberhasilan penyusun dalam menyusun Laporan Praktikum Basis Data ini tidak lepas dari bantuan berbagai pihak. Untuk itu saya selaku penyusun menyampaikan terimakasih yang sebesar-besarnya kepada :

1. **Ir. Iswanto, ST., M.MT.,** selaku Dekan Fakultas Sains dan Teknologi Universitas Muhammadiyah Sidoarjo.
2. **Ade Eviyanti, S.Kom., M.Kom.,** selaku Kepala Program Studi Informatika Universitas Muhammadiyah Sidoarjo.
3. **Ika Ratna Indra Astutik, S.Kom., MT.,** selaku Dosen Praktikum Basis Data Universitas Muhammadiyah Sidoarjo.
4. **Melina Atikawati, S.Kom.,** selaku Laboran Informatika Universitas Muhammadiyah Sidoarjo.
5. **Zianisa Azzahra,** selaku Asisten praktikum yang telah membantu terlaksananya Praktikum Basis Data.
6. Keluarga dan teman-teman yang telah memberi bantuan baik materi maupun spiritual dalam pembuatan Laporan Basis Data.

Saya selaku penyusun menyadari bahwa laporan ini masih jauh dari kesempurnaan, untuk itu sangat mengharapkan kritik dan saran dari berbagai pihak yang sifatnya membangun.

Akhir kata, semoga laporan ini dapat menjadi referensi untuk menambah wawasan para pembaca dan tentunya dapat memberikan manfaat sebagaimana yang diharapkan.

Sidoarjo, 07 Mei 2026



(**Mochammad Ferdyansyah**)

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

DAFTAR ISI

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

BAB I

PENDAHULUAN

A. PROFIL LABORATORIUM

Laboratorium Terpadu Fakultas Sains dan Teknologi digunakan sebagai pusat pembelajaran secara praktek dan eksperimental. Mahasiswa diharapkan akan dapat menerapkan materi kuliah secara langsung pada alat yang telah disediakan, mempelajari alat secara langsung, melakukan pengambilan data, penelitian, dan konsultasi. Dalam laboratorium ini terdapat dua 7 Program Studi yaitu : Informatika, Teknik Mesin, Teknik Industri, Teknik Elektro, Teknik Sipil, Agroteknologi, Teknologi Pangan.

1. Visi Laboratorium :

Menjadi laboratorium terpadu yang unggul dan inovatif dalam pengembangan sains dan teknologi berdasarkan nilai-nilai Islam untuk kesejahteraan masyarakat tingkat ASEAN pada tahun 2038.

2. Misi Laboratorium :

1) Melaksanakan kegiatan praktikum sebagai penerapan teori yang didapat selama perkuliahan.
2) Menyediakan sarana dan prasarana untuk kegiatan penelitian dalam bidang rekayasa IPTEKS yang mendukung proses pembelajaran untuk kesejahteraan masyarakat.
3) Menyediakan sarana dan prasarana untuk kegiatan pengabdian kepada masyarakat dalam rekayasa IPTEKS berbasis potensi lokal untuk kesejahteraan masyarakat.
4) Memberikan layanan kerjasama dengan instansi pemerintah/swasta/ baik dalam maupun luar negeri yang berkelanjutan untuk menguatkan Catur Dharma Perguruan Tinggi Muhammadiyah (pendidikan pengajaran, penelitian, pengabdian dan Al Islam Kemuhammadiyahan).
5) Memberikan layanan pengujian, analisis, dan sertifikasi yang dapat dipertanggung jawabkan keakuratannya dan keabsahannya

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

B. MANAJEMEN LABORATORIUM

1. Definisi

[1] Praktikum Basis Data adalah kegiatan akademis yang wajib dilakukan oleh setiap Mahasiswa Program Studi Informatika yang telah menempuh matakuliah prasarat Basis Data. Praktikum ini berbentuk praktek coding yang menekankan aspek kognitif atau berbentuk magang kerja dilaboratorium yang menekankan aspek psikomotorik diikuti aspek kognitif.

[2] Ka.Prodi adalah penanggung jawab secara administratif segala aktifitas yang ada di lingkungan Program Studi Informatika.

[3] Ka.Lab adalah penanggung jawab secara administratif segala aktifitas yang ada di laboratorium terpadu fakultas sains dan teknologi.

[4] Laboran adalah wakil Ka.Lab dalam melaksanakan aktifitas sehari-hari yang ada dilingkungan masing-masing laboratorium program studi.

[5] Dosen pembimbing praktikum tim staf pengajar yang dibentuk oleh Program Studi yang bertugas membuat modul praktikum, menguji mahasiswa dan membimbing mahasiswa dalam penulisan laporan praktikum secara menyeluruh.

[6] Monitoring merupakan pemantauan yang dilakukan oleh Dosen pembimbing praktikum dan Ka.Lab untuk mengecek pelaksanaan praktikum oleh mahasiswa. Monitoring dilakukan secara acak dan minimum sekali dilakukan dalam satu periode pelaksanaan praktikum.

[7] *Logsheet* adalah lembar kerja yang berisi aktivitas kerja mahasiswa yang diketahui dan ditandatangani oleh asisten praktikum gambar teknik.

[8] Ujian pretest dan posttest praktikum adalah ujian yang dilaksanakan sebelum dan sesudah pelaksanaan praktikum yang dilakukan oleh asisten laboratorium dan dosen pembimbing untuk memperoleh nilai praktikum.

[9] Laporan praktikum adalah naskah laporan praktikum yang ditulis sesuai dengan panduan penulisan praktikum dilaboratorium yang sudah direvisi setelah ujian dan melaporkan aktivitas pelaksanaan praktikum

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

di laboratorium. Laporan praktikum dibuat dalam bentuk *hardcopy* dan *softcopy.*

2. Waktu Dan Tempat Pelaksanaan

Waktu praktikum sesuai dengan jadwal yang telah ditentukan. Tempat pelaksanaan praktikum di Laboratorium yang sudah tercantum pada jadwal.

[1] Jumlah tatap muka

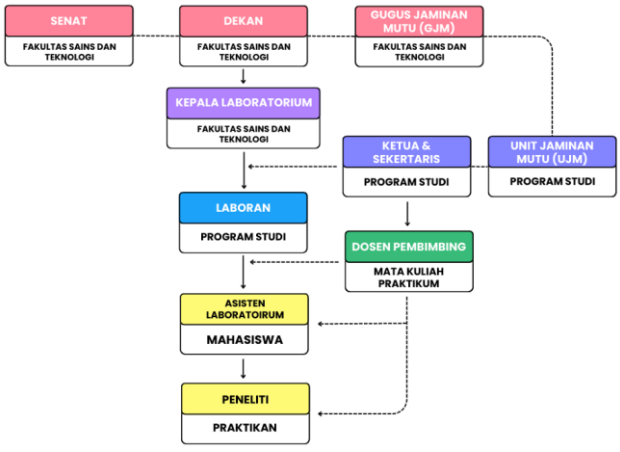
Jumlah tatap muka ditentukan dengan jumlah kelompok mahasiswa praktikum dengan pertimbangan memaksimalkan penggunaan peralatan praktikum, kapasitas laboratorium mampu menampung maksimal 40 mahasiswa praktikum.

[2] Lama praktikum setiap tatap muka

Lama praktikum setiap kelompok adalah 16 jam tatap muka ini dibagi dalam dua sesi, sesi hari pertama 8 jam tatap muka istirahat 1 jam kemudian sesi hari kedua 8 jam tatap muka istirahat 1 jam.

3. Struktur Organisasi

Laboratorium Terpadu berada di bawah struktur Fakultas Sains dan Teknologi Universitas Muhammadiyah Sidoarjo, seperti pada Gambar 1.1 berikut :



Gambar 1.1 Bagan Laboratorium Terpadu Fakultas Sains dan Teknologi

Keterangan :

_____ : Garis Komando

---------- : Garis Koordinasi

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

C. SISTEM PENGELOLAAN

1. Tugas pihak terkait :

[1] Praktikan

- Mendaftar sebagai calon praktikan
- Mengikuti praktikum sesuai jadwal
- Konsultasi kepada asisten dan dosen pembimbing
- Membuat laporan praktikum
- Mengikuti ujian akhir praktikum

[2] Asisten Praktikum

- Menyiapkan segala perlengkapan praktikum
- Menjelaskan modul praktikum
- Membimbing perbaikan laporan praktikum
- Membimbing praktikan membuat laporan
- Memberikan nilai praktikum
- Perawatan laboratorium

[3] Laboran

- Membuat daftar peserta praktikum
- Menyusun jadwal praktikum
- Melakukan pengecekan dan perawatan segala perlengkapan di laboratorium
- Mengevaluasi kegiatan praktikum

[4] Dosen Pembimbing

- Mendampingi kegiatan praktikum
- Membimbing perbaikan laporan praktikum
- Memberkan penilaian hasil praktikum

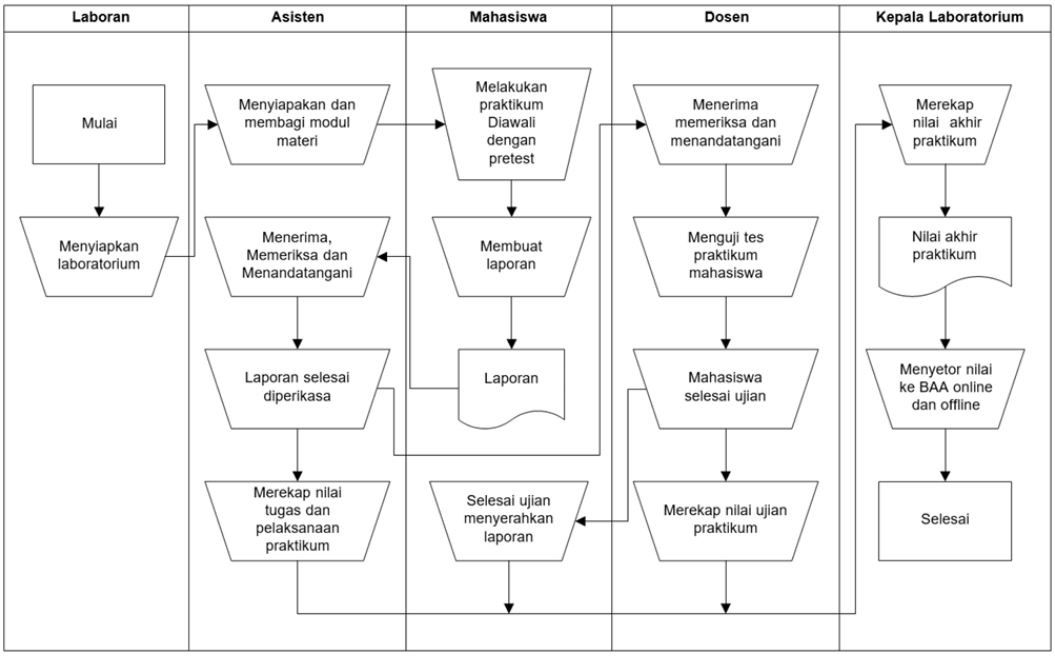
[5] Kepala Laboratorium

- Menkoordinasikan laboran dan asisten praktikum
- Mengajukan penambahan alat praktikum
- Membuat program kerja tahunan
- Menjalin kerjasama antar laboratorium internal maupun external

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

D. PROSEDUR DAN MEKANISME

Langkah-langkah yang dilakukan oleh Laboran, Asisten, Mahasiswa, Dosen dan Ka.lab disajikan dalam diagram alir pada Gambar 1.2.



Gambar 1.2 Diagram alir Langkah-langkah yang Dilakukan oleh Kepala Laboratorium, Mahasiswa, Asisten, Laboran dan Dosen Praktikum pada pelaksanaan praktikum.

Penjelasan :

1. Laboran menyiapkan laboratorium dan perlengkapannya, peralatan alat-alat kelengkapan praktikum yang dipimjam praktikan ditulis dalam form pemimjaman alat yang ditandatangani asisten dan perwakilan praktikan sebagai penanggung jawab peralatan.
2. Asisten menyiapkan materi praktikum berdasarkan modul praktikum
3. Mahasiswa melaksanakan pretest terlebih dahulu kemudian dilanjutkan dengan praktikum didampingi asisten, di monitoring dosen pembimbing dan Ka.Lab secara acak.
4. Mahasiswa membuat *Logsheet* dan diserahkan kepada asisten pada pertemuan berikutnya.
5. Mahasiswa wajib absen untuk setiap sesi tatap muka praktikum, yang dilakukan oleh asisten praktikum.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

6. Asisten memeriksa dan menandatangai asistensi laporan praktikum mahasiswa, yang point koreksinya sesuai dengan petunjuk penulisan laporan praktikum kemudian yang menjadi point utama dalam koreksi diantaranya:
   - Unsur originallitas.
   - Kesesuain materi laporan dengan job sheet yang dikerjakan.
   - Kedalaman analisis data pengukuran produk/quality control.
   - Sistematika laporan sesuai dengan petunjuk penulisan laporan praktikum.
7. Laporan yang sudah dikoreksi/diperiksa asisten dikembalikan lagi kepada mahasiswa, untuk menjadi pertimbangan penulisan laporan yang lebih baik sebelum maju asistensi ke dosen pembimbing.
8. Pada akhir praktikum, Dosen memberikan tes/ujian yang harus diikuti oleh semua mahasiswa praktikum.
9. Dosen menyerahkan nilai hasil tes/ujian praktikum yang dilaksanakan mahasiswa ke Kepala Laboratorium.
10. Asisten merekap nilai praktikum [10%P (tugas Pralab) + 10%K (Kehadiran) + 10%AS (Asistensi) + 10%LS (Laporan Sementara) + 30%LP (Laporan Praktikum) + 30%SH (Seminar Hasil / Ujian Praktikum)], kemudian Asisten menyerahkan nilai ke Laboran.
11. Mata Kuliah Praktikum

    Mata kuliah praktikum merupakan mata kuliah untuk mencapai capaian pembelajaran keterampilan khusus sekaligus pengetahuan. Penilaian untuk mata kuliah Praktikum meliputi aspek :

    a) Pralab (P)
    b) Kehadiran (K)
    c) Asisten (As)
    d) Laporan Sementara (LS)
    e) Laporan Praktikum (LP)
    f) Seminar Hasil (SH)

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Sehingga dapat dirumuskan penilaian KKN sebagai berikut:

$$NA = \frac{1 \times P + 1 \times K + 1 \times As + 1 \times LS + 3 \times LP + 3 \times SH}{10}$$

12. Laboran merekap nilai praktikum dan menyerakhan nilai ke dosen pengampu yang nantinya nilai praktikum dimaukkan nilau tugas dari kompilasi [5%K (Kehadiran) + 5%P (Aktifitas Partisipasi Aktif) + 20%Pro (Hasisl Proyek) + 10%Q (Quiz) + 20%T (Tugas) + 20%UTS (Ujian Tengah Smester + 20%UAS (Ujian Akhir Semester)].

$$NA = \frac{0.5 \times K + 0.5 \times P + 2 \times Pro + 1 \times Q + 2 \times T + 2 \times UTS + 2 \times UAS}{10}$$

Keterangan :

- Kehadiran (K)
- Aktivitas Partisipasi Aktif (P)
- Hasil Proyek (Pro)
- Quiz (Q)
- Tugas (T)
- Ujian Tengah Semester (UTS)
- Ujian Akhir Semester (UAS)

13. Mahasiswa dinyatakan lulus praktikum dengan nilai minimal adalah C range nilai tertera seperti pada table 1.1 berikut :

Tabel 1.3 Range nilai huruf dan angka

Huruf	Angka	Range Nilai
A	4,00	$85 \leq NA \leq 100$
A-	3,67	$80 \leq NA < 85$
B+	3,33	$75 \leq NA < 80$
B	3,00	$70 \leq NA < 75$
B-	2,67	$65 \leq NA < 70$
C+	2,33	$60 \leq NA < 65$
C	2,00	$55 \leq NA < 60$
D	1,00	$40 \leq NA < 55$
E	0,00	$0 \leq NA < 40$

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

E. TATA TERTIB

Tata Tertib yang harus di patuhi baik oleh peneliti maupun praktikan pada saat menggunakan laboratorium adalah sebagai berikut :

1. Praktikum dilaksanakan tepat waktu sesuai dengan jadwal yang ditetapkan.
2. Mahasiswa yang terlambat datang atau absen harus memberikan surat/bukti yang dapat dipercaya.
3. Mahasiswa diperkenankan pindah kelompok/jam/hari praktikum dengan syarat mengkonfirmasi H-3 sebelum pelaksanakan praktikum melalui Asisten Laboratorium.
4. Mahasiswa yang tidak hadir pada saat jadwal yang telah ditentukan diperkenankan mengikuti praktikum berikutnya dengan syarat penilaian sebelumnya akan dinilai kosong, jika ingin melakukan penambahan jam harus koordinasi dengan asisten laboratorium.
5. Mahasiswa harus meminjam alat praktikum dengan cara mengisi lembaran form pinjam alat yang tersedia.
6. Praktikum dianggap selesai jika mahasiswa telah menyerahkan laporan sementara dan alat yang dipinjam dalam keadaan baik, bersih, dan rapi.
7. Kerusakan alat yang dipinjam oleh mahasiswa menjadi tanggung jawab penuh kelompok mahasiswa yang bersangkutan.
8. Selama praktikum berlangsung, mahasiswa dilarang merokok, makan, bergurau, bermain alat, menghidupkan HandPhone, atau pun keluar masuk ruangan tanpa seijin dosen pembimbing/asisten pendamping.
9. Mahasiswa yang dinyatakan tidak lulus praktikum harus mengulang dijadwal praktikum berikutnya.
10. Laporan praktikum diserahkan ke laboratorium paling lambat 2 minggu dari pelaksanaan praktikum, jika lebih dari itu maka akan dikenakan sanksi sesuai arahan dari asisten laboratorium/laboran.

F. PENGGUNAAN LABORATORIUM

Penggunaan Praktikum harus sesuai aturan yang berlaku yang sudah di jelaskan diatas dan lebih penekanan dan kesimpulannya sebagai berikut :

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

1. Pengguna laboratorium khususnya mahasiswa praktikum, adalah mereka yang terdaftar sebagai praktikan yang dikeluarkan oleh DA.
2. Penggunaan alat-alat praktikum sesuai dengan petunjuk kerja yang ada pada modul praktikum dengan sepengetahuan asisten laboratorium.
3. Pemimjaman kelengkapan alat-alat praktikum dengan mengisi form pemimjaman yang telah disediakan sesuai dengan modul praktikum.
4. Pelanggaran atas aturan ini dikenakan sanksi tidak dapat mengikuti praktikum berikutnya.
5. Asisten harus melaporkan terjadinya pelanggaran ke laboran dan mencatat pelanggaran yang terjadi.
6. Kerusakan karena kelalaian praktikan menjadi tanggung jawab praktikan yang bersangkutan.

G. KESEHATAN DAN KESELAMATAN KERJA

Adapun keselamatan kerja yang harus di patuhi oleh seluruh mahasiswa, asisten lab ataupun dosen sebagai berikut :

1. Pada saat akan memasuki ruang laboratorium, peneliti/mahasiswa/dosen melakukan pengisian pada barcode yang terdapat di depan tiap pintu laboratorium atau melalui link : bit.ly/SafetyLabTerpaduSAINTEK
2. Laboran/Asisten Laboratorium menyiapkan alat dan bahan sebelum digunakan.
3. Pengguna laboratorium diwajibkan menaati semua petunjuk keselamatan kerja dan memakai alat-alat perlindungan diri yang diwajibkan.
4. Pengguna laboratorium mengisi logbook yang terdapat pada tiap ruang laboratorium
5. Apabila terdapat pelanggaran penggunaan Alat Pelindung Diri (APD) maupun peraturan lain yang ditetapkan, laboran/asisten laboratorium berhak memberikan sanksi.
6. Setelah selesai menggunakan laboratorium, pengguna laboratorium harus merapikan kembali dan menjaga kebersihan laboratorium.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

BAB II
KURIKULUM

A. PENDAHULUAN MATA KULIAH

Mata kuliah Basis Data merupakan salah satu matakuliah utama yang ada di Jurusan Informatika Universitas Muhammadiyah Sidoarjo dengan kode matakuliah INF23208, 4 SKS. Kompetensi pada matakuliah ini yakni: (1) Pengertian Data dan Informasi, Basis Data, sistem basis data; (2) DMBS dan Lingkungan Basis Data; (3) Desain basis data (Model Konseptual, Model Logis, Model Fisik); (4) Entity Relationship Model; (5) Membuat model RDBMS Definisi Normalisasi -Aturan-aturan normalisasi -Bentuk bentuk normal ke-1, normal ke2, normal ke-3, BCNF, normal ke-4 dan normal ke-5; (6) DBMS MySQL, DLL Prompt; (7) Data Manipulasi Langaunge (Insert, update, Hapus); (8) Join, Query dan Klausa In, Not In, Exist dan not Exist; (9) Create and Replace View; (10) Grant dan Revoke di MySQL; (11) Ketepatan merancang basis data. -Ketepatan mengimplementasikan dari fitur database yang dibuat.. Tujuan dari matakuliah ini yaitu Menguasai konsep teoritis bidang pengetahuan Ilmu Komputer/Informatika dalam mendesain dan mensimulasikan aplikasi teknologi multi-platform yang relevan dengan kebutuhan industri dan masyarakat, memberikan pemahaman dan penguasaan konsep basis data dan mampu menerapkannya ke dalam desain basis data sesuai dengan kebutuhan sistem serta mahasiswa mampu menggunakan Structure Query Language (SQL) pada basis data server dan kasus nyata.

B. TUJUAN PEMBELAJARAN

Tujuan pembelajaran mata kuliah praktikum adalah Memahami Struktur Basis Data bagaimana data disusun, diorganisasikan, dan dikelola dalam bentuk tabel, relasi, dan skema, Mengenal Sistem Manajemen Basis Data (DBMS) dan mempelajari cara kerja serta fungsi dari DBMS, seperti MySQL, PostgreSQL, atau Oracle, dalam mengelola dan mengakses data, Menguasai bahasa pemrograman SQL (Structured Query Language) untuk melakukan berbagai operasi pada basis data, seperti SELECT, INSERT, UPDATE, DELETE, dan lain-lain. Dengan mempelajari praktikum basis data,

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

diharapkan mahasiswa dapat menjadi lebih siap untuk menangani proyek-proyek yang melibatkan pengelolaan data di dunia industri maupun dalam penelitian akademik.

C. PRASYARAT MATAKULIAH

Syarat teknis untuk melaksanakan praktikum Basis Data adalah Mahasiswa telah menempuh mata kuliah Basis Data (INF23208)

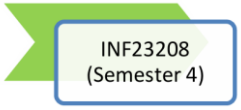
D. SILABUS PERKULIAHAN

Silabus modul praktikum mengacu pada kurikulum program studi Informatika Fakultas Sains dan Teknologi Universitas Muhammadiyah Sidoarjo seperti berikut ini :

1. Basis Data, Model Data, Diagram E-R
2. Struktur Query Language (SQL)
3. Data Definition Language (DDL)
4. Data Manipulation Language (DML)
5. Query dan View
6. Data Control Language (DCL) atau Hak Akses User

E. DIAGRAM PENCAPAIAN KOMPETENSI

Modul praktikum “Basis Data” merupakan modul yang terdapat dalam matakuliah Basis Data dengan kode INF23208, sehingga sebagai berikut ini :



Keterangan :

INF23208: Basis Data

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

F. RENCANA PELAKSANAAN PEMBELAJARAN

Skenario pembelajaran yang digunakan saat praktikum Basis Data sebagai berikut :

RENCANA PELAKSANAAN PEMBELAJARAN (RPP)

TAHUN AJARAN 2025/2026

Satuan Pendidikan	**: Universitas Muhammadiyah Sidoarjo**
Mata Kuliah	**: Basis Data**
Program Studi	**: S1 Informatika**
Semester	**: 2**
Materi Pokok	**: Praktikum Basis Data**
Alokasi waktu	**: 180 Menit**

1. Kompetensi Dasar

Matakuliah ini memberikan pemahaman dan penguasaan konsep basis data dan mampu menerapkannya ke dalam desain basis data sesuai dengan kebutuhan sistem serta mahasiswa mampu menggunakan Structure Query Language (SQL) pada basis data server dan kasus nyata.

2. Tujuan Pembelajaran

- Mahasiswa Mampu Menjelaskan Konsep Dasar Basis Data.
- Mahasiswa mampu Menjelaskan Sistem Basis Data.
- Mahasiswa mampu menerapkan Bahasa SQL (DDL,DML,DCL).
- Mahasiswa Mampu Membuat Database dengan Penerapan SQL pada DBMS.
- Mahasiswa mengetahui bagaimana mengimplementasi suatu algorithma sederhana dengan menggunakan bahasa pemrogaman python untuk Algoritma Searching.
- Mahasiswa dapat melakukan kolaborasi dengan program studi lain dalam rangka Project based Learning.

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 – 2026

Lembar Asistensi

Praktikum Basis Data

Pokok Bahasan 1

Judul	: Basis Data, Model Data, Diagram E-R
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 6
Dilaksanakan	: 07 Mei 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
13/05 26	12/05 2026
(Ika Ratna Indra Astutik, S.Kom., MT.)	**(Zianisa Azzahra)**

BAB III
MATERI MODUL
POKOK BAHASAN 1

Basis Data, Model Data, Diagram E-R

A. PENDAHULUAN

Pada pokok bahasan ini berisi penjelasan disertai contoh mengenai konsep basis data, pemodelan data dan pembuatan diagram E-R yang menjadi pemahaman dasar bagi mahasiswa sebelum mempelajari sistem basis data dan Structure Query Language (SQL).

B. TUJUAN

Setelah mempelajari bab ini diharapkan mahasiswa dapat :

1. Memahami sistem basis data dan komponennya.
2. Membuat desain basis data menggunakan ER_Diagram.
3. Memahami dan mengimplementasikan fitur-fitur yang ada pada ER_Diagram.

C. PENYAJIAN (TUTORIAL)

1. Konsep Sistem Basis Data

Basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi serta dapat diakses dengan mudah dan tepat menggunakan perangkat lunak (program aplikasi) untuk menghasilkan sebuah informasi. Sistem basis data merupakan ruang lingkup yang lebih luas dari basis data. Sistem basis data memuat sekumpulan basis data dalam suatu sistem yang mungkin tidak ada hubungan antara satu dengan yang lain, tetapi secara keseluruhan mempunyai hubungan sebagai sebuah sistem yang didukung oleh komponen lainnya. Komponen

Sistem Basis Data: Perangkat Keras (Hardware), Sistem Operasi (Operating System), basis data (Database), DBMS (Database Management System), Pemakai (User). DBMS (Database Management

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

System) merupakan basis data dan set perangkat lunak (software) untuk pengelolaan basis data.

2. Konsep Model Data

Model data merupakan suatu cara untuk menjelaskan tentang data-data yang tersimpan dalam basis data dan bagaimana hubungan antar data tersebut untuk para pengguna (user) secara logika. Secara garis besar model data dapat dikelompokkan menjadi 3 macam yaitu

1. Model Data Berbasis Objek (Object based data model)

   Merupakan himpunan data dan relasi yang menjelaskan hubungan logik antar data dalam suatu basis data berdasarkan pada obyek datanya. Salah satunya adalah Entity Relationship Model.

   Model *Entity Relationship Diagram* (ERD) atau *Conceptual Data Model* (CDM) Merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan suatu persepsi bahwa dunia nyata terdiri dari obyek-obyek dasar (entitas) yang mempunyai hubungan atau relasi antar obyek-obyek dasar (entitas) tersebut yang dilukiskan dengan menggunakan simbol-simbol grafik tertentu.

2. Model Data berbasis Record (*Record Based Data Model*)

   Model ini berdasarkan pada record/rekaman untuk menjelaskan kepada para pemakai tentang logik antar data dalam basis data. Salah satunya adalah Relational model. Model Rasional merupakan model data yang menjelaskan pada pengguna tentang hubungan logik antar data dalam basis data dengan mempresentasikannya ke dalam betuk tabel-tabel yang terdiri atas sejumlah baris yang menunjukkan record dan kolom yang menunjukkan *atribut* tertentu.

3. Physical Based Data Model

   Model ini berdasarkan pada teknis penyimpanan record dalam basis data. Model ini jarang digunakan untuk memodelkan data kepada pemakai karena kerumitan dan kompleksitasnya yang tinggi.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

3. Bahasa Basis Data

Bahasa yang digunakan untuk mendefinisikan, mengelolah dan memanipulasi basis data dikelompokkan 3 macam yaitu :

1. DDL (*Data Definition Language*) digunakan untuk mendefinisikan struktur dan kerangka dari basis data yang meliputi :
   a. Membentuk basis data, tabel, indeks.
   b. Mengubah struktur table.
   c. Menghapus basis data, tabel atau indeks.
2. DML (*Data Manipulation Language*) digunakan untuk menjabarkan pemrosesan data pada basis data yang meliputi :
   a. Menambahkan atau menyisipkan data baru ke basis data.
   b. Mengelolah data yang tersimpan dalam basis data (query).
   c. Mengubah dan menghapus data dalam basis data.
3. DCL (*Data Control Language*) digunakan untuk pengaturan hak akses pengguna pada basis data yang meliputi :
   a. Menugaskan hak akses terhadap basis data kepada pengguna atau grup pengguna.
   b. Membatalkan hak akses pengguna terhadap basis data.

4. Entity Relationship Diagram (ER-D)

Merupakan model data yang dikembangkan berdasarkan obyek atau entitas. ER_D berguna membantu perancang atau analis sistem pada saat melakukan analisis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan direlasikan antar data di dalamnya.

1. Komponen ER_Diagram

   Sebuah diagram ER tersusun atas tiga komponen, yaitu entitas yang merupakan obyek dasar yang terlibat dalam sistem, atribut yang berperan sebagai penjelas entitas, kerelasian antar entitas menunjukkan hubungan yang terjadi diantara dua entitas.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

a. Entitas (Entity)

Entitas menunjukkan obyek-obyek dasar yang terkait di dalam sistem. Obyek dasar dapat berupa orang, benda atau hal yang keterangannya perlu disimpan di dalam basis data. Macam-macam Entitas :

- Entitas Reguler

  Entitas ini disebut juga entitas dominan (strong entity). Keberadaan entitas ini tidak tergantung pada entitas yang lain.

  Contoh : Mahasiswa, Matakuliah.

- Entitas dependen

  Entitas ini disebut juga entitas tidak bebas/independen atau entitas lemah (weak entity) atau entitas subordinat. Entitas ini dapat muncul jika ada entitas lain sebagai acuannya (entitas reguler).

  Contoh : Matakuliah_konsentrasi, bergantung pada entitas Matakuliah.

- Entitas super type dan sub type

  Entitas super type merupakan entitas yang memiliki tingkatan yang lebih tinggi yaitu membawahi atau mempunyai entitas bagian yang lebih rendah.

  Contoh : Entitas Karyawan.

  Entitas sub type merupakan entitas yang lebih rendah yaitu entitas yang menjadi entitas bagian dari entitas lain.

  Contoh : Entitas karyawan_tetap dan karyawan_tidak_tetap

b. Atribut (Attribute)

Merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan ke dalam database. Atribut berfungsi sebagai penjelas pada sebuah entitas.

Contoh : mahasiswa mempunyai atribut nim, nama, jurusan, kelamin, tempat_lahir, tanggal_lahir, dsb.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

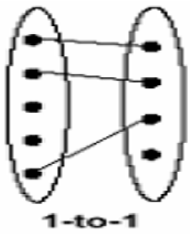
Atribut pada sebuah entitas dibagi menjadi 2 yaitu :

- Atribut sederhana (simple attribute), yaitu jika atribut berisi sebuah komponen/nilai/elementer.
  Contoh : pada entitas mahasiswa adalah tahun masuk = 2013
- Atribut komposit (composite attribute), yaitu jika atribut berisi lebih dari sebuah komponen nilai.
  Contoh : pada entitas mahasiswa adalah tanggal lahir yang terdiri atas komponen nilai tanggal, bulan, tahun.

c. Kerelasian antar entitas (*Entity Relationship*)

Mendefiniskan hubungan antara 2 buah entitas. Jenis kerelasian antar entitas dibagi mejadi 3 sebagai berikut :

1. Kerelasian jenis satu ke satu (one to one), kerelasian terjadi jika kejadian atau transaksi di antara dua entitas yang berhubungan hanya memungkinkan terjadi sebuah kejadian atau transaksi pada kedua entitas.



Gambar 1.1 Relasi satu ke satu

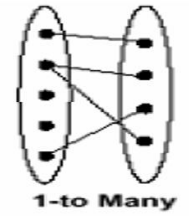
Dimana setiap tupel (baris) pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B, dan begitu juga sebaliknya setiap tupel pada entitas B berhubungan dengan paling banyak satu tupel pada entitas A.

2. Kerelasian banyak ke satu (many to one) atau satu ke banyak (one to many), kerelasian ini terjadi jika kejadian atau transaksi di antara dua entitas yang berhubungan hanya memungkinkan terjadi satu kali dalam entitas pertama dan dapat terjadi lebih dari satu kali kejadian atau transaksi pada entitas kedua.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

- Satu ke banyak (one to many)

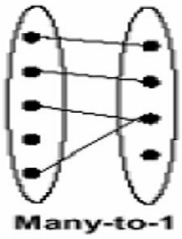
  Dimana satu tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, tetapi tidak sebaliknya, dimana setiap tupel pada entitas B berhubungan dengan paling banyak satu tupel pada entitas A.



Gambar 1.2 Relasi satu ke banyak

- Banyak ke satu (many to one)

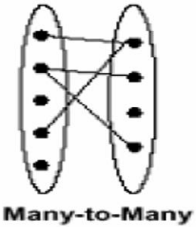
  Dimana setiap tupel pada entitas A dapat berhubungan dengan paling banyak satu tupel pada entitas B, tetapi tidak sebaliknya, dimana setiap tupel pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B.



Gambar 1.3 Relasi banyak ke satu

3. Kerelasian jenis banyak ke banyak (many to many)

   Kerelasian jenis ini terjadi jika kejadian atau transaksi di antara dua entitas yang berhubungan memungkinkan terjadi lebih dari satu kali dalam entitas pertama dan kedua.



Gambar 1.4 Relasi banyak ke banyak

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Dimana setiap tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, dan demikian juga sebaliknya, dimana setiap tupel pada entitas B dapat berhubungan dengan banyak tupel pada entitas A.

2. Langkah-langkah Membuat ER_Diagram

   Untuk membuat ER_Diagram secara lengkap dapat dilakukan dengan mengikuti langkah langkah sebagai berikut :

   - Identifikasikan setiap entitas yang terlibat.
   - Lengkapi masing-masing entitas dengan atribut yang sesuai.
   - Tentukan primari key dari masing-masing entitas.
   - Identifikasikan setiap kerelasian berikut jenisnya yag terjadi di antara entitas dengan membuat tabel daftar kerelasian antar entitas.
   - Gambarkan simbol-simbol entitas, atribut, dan kerelasian antar entitas secara jelas dan tidak bertabrakan.

   Cek ER_Diagram yang terbenuk, dalam hal : kelengkapan entitas, kelengkapan atribut, kelengkapan kerelasian antar entitas dan jenis kerelasian antar entitas.

D. LEMBAR KERJA DAN TUGAS

A. LEMBAR KERJA

Bacalah dengan seksama studi kasus perancangan basis data di bawah ini: Perpustakaan di Universitas Muhammadiyah Sidoarjo memutuskan untuk menyimpan semua informasi mengenai koleksi bukunya dalam sebuah basis data (database). Basis data yang akan di rancang mempuyai deskripsi sebagai berikut :

1. Setiap mahasiswa yang ingin meminjam buku harus mempuyai kartu anggota yang terdiri dari kode anggota, nama, alamat, email dan telpon.
2. Tiap buku yang di pinjamkan mempunyai kode buku, judul buku, harga, tahun terbit, pegarang dan penerbit.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

3. Tiap pengarang yang bukunya di beli perpustakaan mempunyai kode pengarang dan nama pengarang.
4. Tiap penerbit yang bukunya telah di beli oleh perpustakaan mempunyai kode penerbit,nama penerbit, nama perusahaan, alamat, kabupaten dan telpon.
5. Tiap buku bisa di pinjam oleh beberapa anggota dan seorang anggota bisa meminjam beberapa buku.
6. Tiap pengarang mungkin mengarang beberapa buku dan setiap buku dapat di pinjam oleh beberapa anggota.
7. Tiap penerbit bisa menerbitkan beberapa buku. seorang pengarang bisa menghasilkan beberapa buku.

Langkah-langkah merancang basis data perpustakaan :

1. Identifikasikan setiap entitas yang terlibat dengan membuat tabel daftar entitas.

   Tabel 1.1 Daftar Entitas

No	Nama Entitas
1.	Anggota
2.	Buku
3.	Pengarang
4.	Penerbit

2. Lengkapi masing-masing entitas dengan atribut yang sesuai.

   Tabel 1.2 Atribut masing-masing entitas

Entitas	Atribut yang diperlukan
Anggota	Kode_anggota, nama, alamat, email, telpon
Buku	Kode_buku, judul_buku, harga, tahun_terbit, pengarang, penerbit
Pengarang	Kode_pengarang, nama_pengarang
Penerbit	Kode_penerbit, nama_penerbit, nama_perusahaan, alamat, kabupaten, telpon

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

3. Tentukan primari key dari masing-masing entitas

Tabel 1.3 Primary key

Entitas	Primary Key
Anggota	Kode_anggota
Buku	Kode_buku
Pengarang	Kode_pengarang
Penerbit	Kode_penerbit

4. Identifikasikan setiap kerelasian berikut jenisnya yag terjadi di antara entitas dengan membuat tabel daftar kerelasian antar entitas.

Tabel 1.4 Kerelasian antar entitas

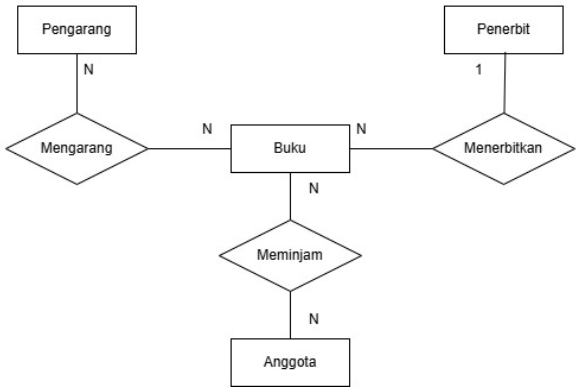
Entitas yang berhubungan		Nama Kerelasian	Jenis Kerelasian	Representasi
Entitas I	Entitas II			
Anggota	Buku	Meminjam	n-ke-n	File Peminjaman : Id_peminjaman, kode_anggota, kode_buku, tanggal_pinjam, tanggal_kembali
Buku	Pengarang	Mengarang	n-ke-n	Penghubung : Kode_pengarang dalam entitas buku
Buku	Penerbit	Menerbitkan	n-ke-1	Penghubung : Kode_penerbit dalam entitas buku

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

5. Gambarkan simbol-simbol entitas, atribut, dan kerelasian antar entitas secara jelas dan tidak bertabrakan.

   Hasil ER_Diangram untuk basis data pengolahan data perpustakaan tanpa melibatkan atribut untuk setiap entitas sebagai berikut:

   Ferdy_060

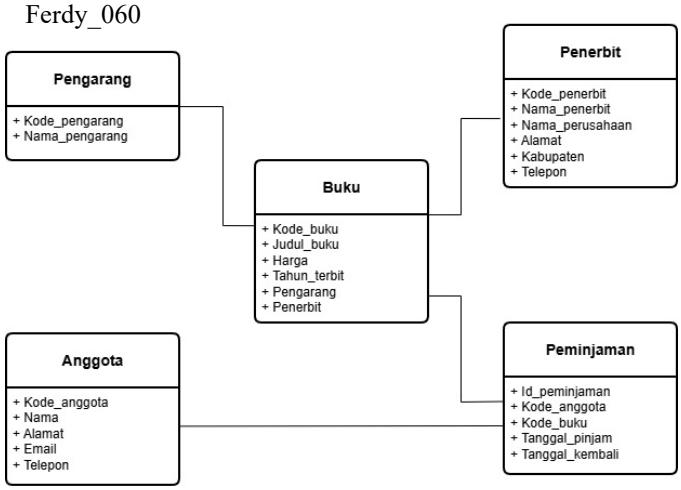


Gambar 1.5 E-R Diagram Pepustakaan

6. Gambarkan model konseptual dari ER_diagram diatas

   Model konseptual digunakan untuk emperjelas relasi antara entitas satu dengan entitas lain menggunakan Primary key dan Foreign Key.

   Ferdy_060



Gambar 1.6 Model Konseptual Perpustakaan

7. Membuat Kamus Data (Struktur Entitas)

   Kamus data digunakan untuk menjabarkan struktur dari tabel atau entitas dan basis data

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Tabel 1.5 Anggota

No	Field	Type	Size	Keterangan
1.	Kode_anggota	Varchar	5	Primary Key
2.	Nama	Varchar	25	
3.	Alamat	Varchar	50	
4.	Email	Varchar	20	
5.	Telpon	Varchar	15	

Tabel 1.6 Buku

No.	Field	Type	Size	Keterangan
1.	Kode_buku	Varchar	5	Primary Key
2.	Judul_buku	Varchar	35	
3.	Harga	Integer	5	Default 0
4.	Tahun_terbit	Varchar	5	
5.	Kode_pengarang	Varchar	5	Foreign Key
6.	Kode_penerbit	Varchar	5	Foreign Key

Tabel 1.7 Pengarang

No.	Field	Type	Size	Keterangan
1.	Kode_pengarang	Varchar	5	Primary Key
2.	Nama_pengarang	Varchar	35	

Tabel 1.8 Penerbit

No.	Field	Type	Size	Keterangan
1.	Kode_penerbit	Varchar	5	Primary Key
2.	Nama_penerbit	Varchar	25	
3.	Nama_perusahaan	Varchar	50	
4.	Alamat	Varchar	50	
5.	Kabupaten	Varchar	25	
6.	Telepon	Varchar	15	

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Tabel 1.9 Peminjaman

No.	Field	Type	Size	Keterangan
1.	Peminjam	Member	5	Auto Increment
2.	Kode_anggota	Varchar	5	Foreign Key
3.	Kode_buku	Varchar	5	Foreign Key
4.	Tanggal_pinjam	Date		Default
5.	Tanggal_kembali	Date		Default

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

B. TUGAS

Studi Kasus

Universitas Harapan Jaya yang beralamatkan di Jl. Diponegoro 11B Indonesia akan membuat sebuah basis data untuk menyimpan dan mengelolah data pegawai. Basis data yang akan dibuat diberi nama basis data Kepegawaian. Basis data ini harus dapat mengelolah data pegawai dan data absensi pegawai.

Pada Universitas ini pegawai dibedakan menjadi pegawai tetap dan pegawai kontrak. Selain itu pegawai juga mempunyai jabatan dan golongan yang berbeda. Dimana Jabatan terbagi atas : Rektor, Wakil Rektor, Kepala Biro, Kepala UPT, Dekan, Kepala Program Studi, Kepala Laboratorium, Staff, Keamanan, Kebersihan. Sedangkan Golongan ditentukan oleh tingkat pendidikan yang di tempuh, dimulai dari golongan IA sampai golongan IVD. Jam Kerja di Universitas dibedakan menjadi dua yaitu hari Senin s/d Jum'at pukul 08.00-14.00 WIB sedangkan hari Sabtu pukul 08.00-12.00 WIB.

Dari studi kasus di atas :

1. Bacalah dengan seksama studi kasus di atas kemudian analisa-lah basis data pengolahan data kepegawaian seperti langkah-langkah diatas.
2. Kumpulkan laporan (hasil analisa basis data) sebelum Anda masuk ke pokok bahasan 2

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Hasil analisa

1. Mengidentifikasikan setiap entitas yang terlibat.

   Tabel Daftar Entitas

No	Nama Entitas
1.	Pegawai
2.	Absensi
3.	Jabatan
4.	Golongan
5.	Status pegawai
6.	Jam kerja

2. Melengkapi masing-masing entitas dengan atribut yang sesuai.

   Tabel Atribut masing-masing entitas

Entitas	Atribut yang diperlukan
Pegawai	Kode_pegawai, nama_pegawai, status_pegawai, kode_jabatan, kode_golongan, alamat, telepon
Absensi	Kode_absensi, kode pegawai, tanggal_absen, jam_masuk, jam_keluar, keterangan
Jabatan	Kode_jabatan, nama_jabatan
Golongan	Kode_golongan, nama_golongan
Status pegawai	Kode_status, nama_status
Jam kerja	Kode_jamkerja, hari, jam_masuk, jam_pulang

3. Menentukan primary key dari masing-masing entitas

   Tabel Primary key

Entitas	Primary Key
Pegawai	Kode_pegawai
Absensi	Kode_absensi
Jabatan	Kode_jabatan
Golongan	Kode_golongan

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Status pegawai	Kode_status
Jam kerja	Kode_jamkerja

4. Mengidentifikasikan setiap kerelasian berikut jenisnya yag terjadi di antara entitas.

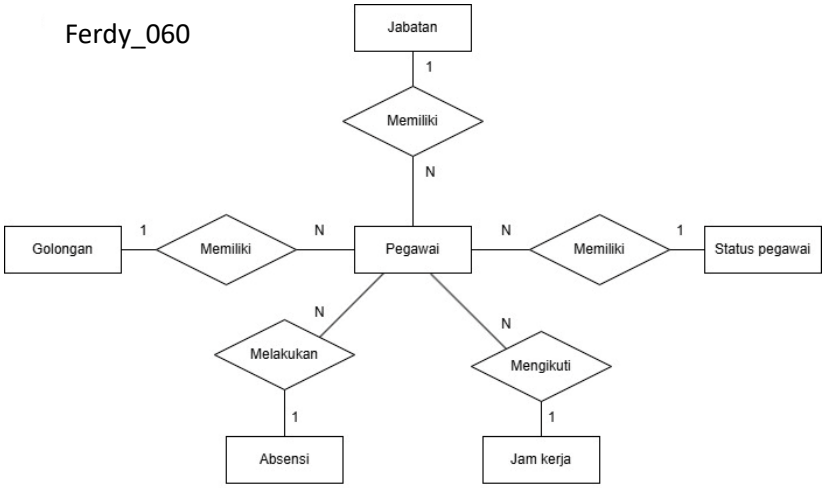
   Tabel Kerelasian antar entitas

Entitas yang berhubungan		Nama Kerelasian	Jenis Kerelasian	Representasi
Entitas I	Entitas II			
Pegawai	Absensi	Melakukan	1-ke-n	Penghubung: kode_pegawai pada entitas absensi
Pegawai	Jabatan	Memiliki	n-ke-1	Penghubung: kode_jabatan pada entitas pegawai
Pegawai	Golongan	Memiliki	n-ke-1	Penghubung: kode_golongan pada entitas pegawai
Pegawai	Status pegawai	Memiliki	n-ke-1	Penghubung: kode_status pada entitas pegawai
Pegawai	Jam kerja	Mengikuti	n-ke-1	Penghubung: kode_jamkerja pada entitas pegawai

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

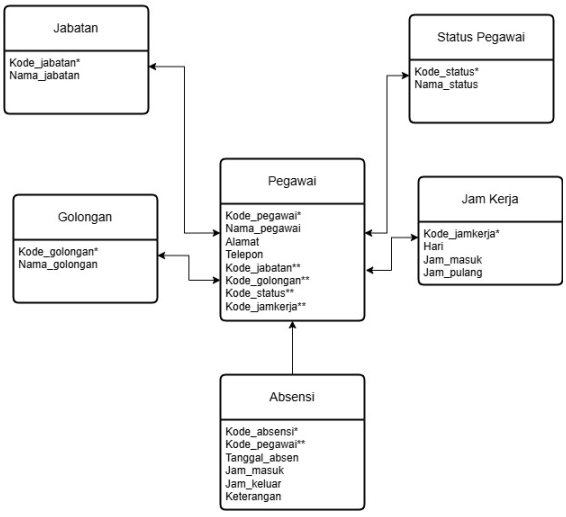
5. Gambar simbol-simbol entitas, atribut, dan kerelasian antar entitas. Hasil ER_Diangram untuk basis data pengolahan data perpustakaan tanpa melibatkan atribut untuk setiap entitas sebagai berikut :

Ferdy_060



6. Gambar model konseptual dari ER_diagram diatas.

Ferdy_060



fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

7. Kamus Data (Struktur Entitas)

Tabel Pegawai

No.	Field	Type	Size	Keterangan
1.	Kode_pegawai	Varchar	5	Primary Key
2.	Nama_pegawai	Varchar	30	
3.	Alamat	Varchar	50	
4.	Telepon	Varchar	15	
5.	Kode_jabatan	Varchar	5	Foreign Key
6.	kode_golongan	Varchar	5	Foreign Key
7.	kode_status	Varchar	5	Foreign Key
8.	kode_jamkerja	Varchar	5	Foreign Key

Tabel Jabatan

No.	Field	Type	Size	Keterangan
1.	Kode_jabatan	Varchar	5	Primary Key
2.	Nama_jabatan	Varchar	30	

Tabel Golongan

No.	Field	Type	Size	Keterangan
1.	Kode_golongan	Varchar	5	Primary Key
2.	Nama_golongan	Varchar	10	

Tabel Status Pegawai

No.	Field	Type	Size	Keterangan
1.	Kode_status	Varchar	5	Primary Key
2.	Nama_status	Varchar	15	

Tabel Jam Kerja

No.	Field	Type	Size	Keterangan
1.	Kode_jamkerja	Varchar	5	Primary Key
2.	Hari	Varchar	15	

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

3.	Jam_masuk	Time		
4.	Jam_pulang	Time		

Tabel Absensi

No.	Field	Type	Size	Keterangan
1.	Kode_absensi	Varchar	5	Primary Key
2.	Kode_pegawai	Varchar	5	Foreign Key
3.	Tanggal_absen	Date		
4.	Jam_masuk	Time		
5.	Jam_pulang	Time		
6.	Keterangan	Varchar	20	

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 – 2026

Lembar Asistensi

Praktikum Basis Data

Pokok Bahasan 2

Judul	: Struktur Query Language (SQL)
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 6
Dilaksanakan	: 07 Mei 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
13/05 26	12/05 2026
(Ika Ratna Indra Astutik, S.Kom., MT.)	**(Zianisa Azzahra)**

POKOK BAHASAN 2

Struktur Query Language (SQL)

A. PENDAHULUAN

Pada pokok bahasan ini akan dibahas mengenai structured query language (SQL) yang pembahasannya meliputi definisi dan representasi sql, elemen-elemen pada sql serta penerapan operasi elemen sql pada MySQL

B. TUJUAN

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Menjelaskan definisi dan representasi SQL.
2. Mengetahui elemen-elemen pada SQL.
3. Memahami fungsi-fungsi dan operator dalam SQL.
4. Mengetahui cara menginstall MySQL pada sistem operasi Windows.

C. PENYAJIAN (TUTORIAL)

A. SQL (Structured Query Language)

SQL merupakan suatu bahasa (language) standar menurut ANSI (American National Standards Institute) yang digunakan untuk mengakses basis data. SQL pertama kali diterapkan pada sistem R (sebuah proyek riset pada laboratorium riset San Jose, IBM). Kini SQL juga dijumpai pada berbagai platform, dari mikrokomputer hingga mainframe. SQL dapat digunakan baik secara berdiri sendiri maupun dilekatkan pada bahasa-bahasa lain seperti C dan Delphi. SQL juga telah menjadi bagian dari sejumlah DBMS, seperti Oracle, Sybase, MySQL dan Informix.

B. Elemen SQL

Elemen dasar SQL mencakup pernyataan, nama, tipe data, konstanta, ekspresi, operator relasi, operator logika dan fungsi bawaan.

a. Pernyataan

Merupakan perintah SQL yang meminta sesuatu tindakan kepada DBMS (Database Management System). SQL memiliki kira-kira 30

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

pernyataan. Beberapa pernyataan dasar SQL dapat dilihat pada tabel berikut :

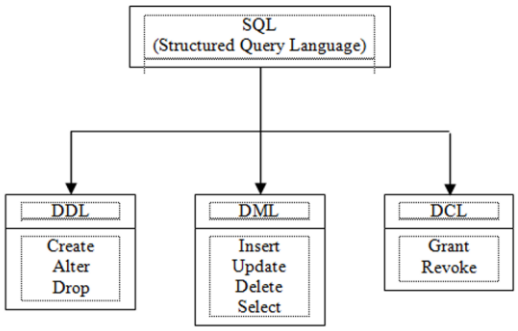
Tabel 2.1 Pernyataan SQL

Pernyataan	Keterangan
CREATE	Menciptakan basis data, tabel atau indeks
ALTER	Mengubah struktur tabel
DROP	Menghapus basis data, tabel atau indeks
COMMIT	Mengakhiri sebuah eksekusi transaksi data
ROLLBACK	Mengembalikan ke keadaan semula sekiranya suatu transaksi gagal dilaksanakan
INSERT	Menambahkan sebuah baris pada tabel
UPDATE	Mengubah nilai pada sebuah baris
SELECT	Memilih baris dan kolom pada tabel
DELETE	Menghapus baris pada tabel
GRANT	Menugaskan hak terhadap basis data kepada pengguna atau grup pengguna
REVOKE	Membatalkan hak terhadap basis data

Yang semuanya dikelompokkan berdasarkan fungsinya masing-masing yaitu :

a. Data Definition Language (DDL) : Digunakan untuk mendefinisikan data dengan menggunakan perintah : CREATE, DROP, ALTER.
b. Data Manipulation Language (DML) : Digunakan untuk memanipulasi data dengan menggunakan perintah : INSERT, SELECT, UPDATE, DELETE.
c. Data Control Language (DCL) : Digunakan untuk mengontrol hak para pemakai data dengan perintah : GRANT, REVOKE.

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271



Gambar 2.1 Komponen SQL

b. Nama

Nama digunakan sebagai identitas bagi objek-objek pada DBMS (Database Management System). Contoh objek pada DBMS adalah tabel, kolom dan pengguna.

c. Tipe Data

Setiap data memiliki tipe data. Berikut ini adalah tipe data dalam MySQL :

Tabel 2.2 Tipe data untuk numerik

Tipe	Keterangan	Range Nilai
TINYINT	Nilai integer yang sangat kecil	Signed : -128 s.d. 127 Unsigned : 0 s.d. 255
SMALLINT	Nilai integer yang kecil	Signed : -32768 s.d. 32767 Unsigned : 0 s.d. 65535
MEDIUMINT	Integer dengan nilai medium	Signed : -8388608 s.d. 8388607 Unsigned : 0 s.d. 16777215
INT	Integer dengan nilai standar	Signed : -2147483648 s.d. 2147483647 Unsigned : 0 s.d. 4294967295
BIGINT	Integer dengan nilai besar	Signed : -9223372036854775808 s.d. 9223372036854775807 Unsigned : 0 s.d. 18446744073709551615
FLOAT	Bilangan desimal dengan single-precission	minimum ± 1.175494351e-38 maksimum ± 3.402823466e+38

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

DOUBLE	Bilangan desimal dengan double-precission	minimum ± 2.2205738585072014e-308 maksimum ± 1.7976931348623457e+308
DECIMAL(M, D)	Bilangan float (desimal) yang dinyatakan sebagai string. M adalah jumlah digit yang disimpan dalam suatu kolom, N adalah jumlah digit dibelakang koma	Tergantung pada nilai M dan D

Keterangan :

Signed dan Unsigned adalah atribut untuk tipe data numerik

- Signed : Data yang disimpan dalam suatu kolom dapat berupa data negatif dan positif.
- Unsigned : Digunakan agar data yang dimasukkan bukan data negatif (>=0). Tipe data float tidak Dapat dinyatakan dengan unsigned.

Tabel 2.3 Tipe data string atau karakter

Tipe	Keterangan	Ukuran Maksimum
CHAR(n)	String karakter dengan panjang yang tetap, yaitu n	1 M byte
VARCHAR(n)	String karakter dengan panjang yang tidak tetap, maksimum n.	1 M byte
TINYBLOB	BLOB (Binary Large Object) yang sangat kecil	2^8-1 byte
BLOB	BLOB berukuran kecil	2^{16}-1 byte
MEDIUMBLOB	BLOB berukuran sedang	2^{24}-1 byte
LONGBLOB	BLOB berukuran besar	2^{32}-1 byte
TINYTEXT	String teks yang sangat kecil	2^8-1 byte

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

TEXT	String teks berukuran kecil	2^{16}-1 byte
MEDIUMTEXT	String teks berukuran medium(sedang)	2^{24}-1 byte
LONGTEXT	String teks berukuran besar	2^{32}-1 byte
ENUM	Enumerasi, kolom dapat diisi dengan satu member enumerasi	65535 anggota
SET	Himpunan, kolom dapat diisi dengan beberapa nilai anggota himpunan	64 anggota himpunan

Tabel 2.4 Tipe data tanggal dan jam

Tipe	Range	Format
DATE	"1000-01-01" s.d. "9999-12-31"	"0000-00-00"
TIME	"-832:59:59" s.d. "838:59:59"	"00:00:00"
DATETIME	"1000-01-01 00:00:00" s.d. "9999-12-31 23:59:59"	"0000-00-00 00:00:00"

d. Konstanta

Konstanta menyatakan nilai yang tetap atau tidak berubah. Konstanta sering di pakai pada perintah SELECT. Konstanta di bagi menjadi 2 :

1. Konstanta bertipe numerik : 200, -3, 1500, 3.25
2. Konstanta bertipe karakter : 'Teknik Informatika'

Keterangan :

Konstanta bertipe karakter atau String diapit oleh tanda petik tunggal. Dan konstanta dengan nilai pecahan desimal adalah berupa tanda titik.

e. Operator Aritmatika

Operator Aritmatika adalah ekspresi untuk memperoleh suatu nilai dari hasil perhitungan.

Contoh : harga*jumlah+2

Simbol-simbol yang dapat digunakan pada ekspresi aritmatika :

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Tabel 2.5 Simbol Ekspresi Aritmatika

Simbol	Keterangan
*	Perkalian
/	Pembagian
+	Penjumlahan
-	Pengurangan
%	Sisa pembagian

f. Operator Relasi

Merupakan operator yang digunakan untuk membandingkan suatu nilai dengan nilai yang lain. Biasanya operator relasi digunakan bersamaan dengan operator logika dalam membantu untuk menampilkan informasi dengan kriteria tertentu. Simbol-simbol yang dapat digunakan pada operator relasi :

Tabel 2.6 Simbol Operator Relasi

Simbol	Keterangan
=	Sama dengan
>	Lebih besar
<	Lebih kecil
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan
<>	Tidak sama dengan

g. Operator Logika

Operator logika ada 3 yaitu OR, AND dan NOT

Tabel 2.7 Operator Logika

Simbol	Keterangan
NOT atau !	Sebagai negasi atau pembalik nilai
OR atau ||	Atau
AND atau &&	Dan

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

h. Operator Pembanding

Tabel 2.8 Operator Pembanding

Simbol	Keterangan
IS NOT NULL	Apakah sebuah nilai adalah tidak kosong (not null)
IS NULL	Apakah sebuah nilai adalah kosong (null)
BETWEEN	Apakah suatu nilai di antara dua batasan nilai
IN	Apakah suatu nilai berada di dalam pilihan yang ada
NOT IN	Apakah suatu nilai tidak berada dalam pilihan yang ada
LIKE	Apakah suatu nilai sesuai dengan kriteria tertentu
NOT LIKE	Apakah suatu nilai tidak sesuai dengan kriteria tertentu

i. Aggregate Functions (Fungsi Agregat)

Fungsi adalah sebuah subprogram yang menghasilkan suatu nilai jika dipanggil. Fungsi agregat adalah fungsi standar di dalam SQL, suatu fungsi yang digunakan untuk melakukan summary, fungsi statistik standar yang dikenakan pada suatu tabel atau query.

1. SUM(ekspresi)

   Fungsi ini digunakan untuk mendapatkan nilai total dari suatu kolom pada suatu tabel

2. AVG(ekspresi)

   Fungsi ini digunakan untuk mencari rata-rata nilai dalam suatu kolom dari suatu tabel atau ekspresi. Ekspresi dalam fungsi AVG umumnya adalah nama kolom. Kolom yang dicari nilai rata-ratanya adalah kolom dengan tipe data numerik.

3. COUNT(x)

   Fungsi ini digunakan untuk menghitung jumlah record (baris) dari suatu kolom dari suatu tabel. X adalah nama kolom yang ingin dicari jumlah barisnya.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

4. MAX(ekspresi)

   Fungsi ini digunakan untuk mencari nilai terbesar dari suatu kolom dari suatu tabel. Kolom yang dicari nilai terbesarnya memiliki tipe data numerik.

5. MIN(ekspresi)

   Fungsi ini digunakan untuk mencari nilai terkecil dari suatu kolom dari suatu tabel. Kolom yang dicari nilai terkecilnya memiliki tipe data numerik.

C. XAMPP dan MySQL (MariaDB)

XAMPP adalah perangkat lunak open-source yang berfungsi sebagai server web mandiri (localhost). Nama XAMPP merupakan singkatan dari X (lintas platform/berjalan di berbagai OS), Apache (web server), MySQL/MariaDB (sistem manajemen basis data), PHP, dan Perl (bahasa pemrograman).

Dalam praktikum basis data ini, kita menggunakan XAMPP karena di dalamnya sudah terintegrasi perangkat lunak MySQL. MySQL sendiri merupakan Relational Database Management System (RDBMS) yang menggunakan bahasa standar SQL (Structured Query Language).

Kelebihan menggunakan XAMPP (dan MySQL di dalamnya) dalam pengelolaan data adalah:

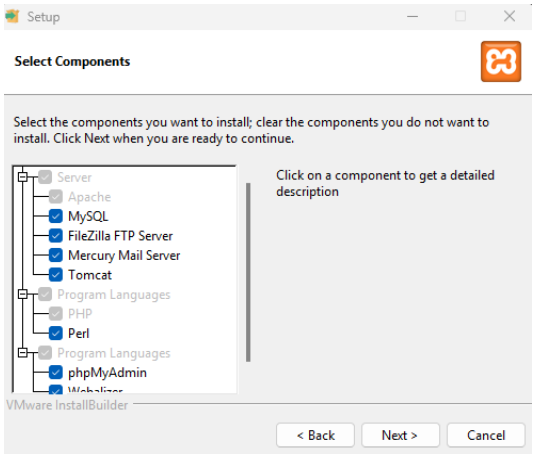
- Praktis dan Terintegrasi: Cukup dengan satu kali instalasi, sistem sudah mencakup web server, database engine (MySQL), dan antarmuka manajemen visual (phpMyAdmin).
- Mudah Digunakan: Dilengkapi dengan XAMPP Control Panel yang sangat memudahkan praktikan untuk menghidupkan (start) atau mematikan (stop) layanan database.
- Open Source dan Gratis: Sepenuhnya gratis tanpa biaya lisensi di bawah naungan General Public License (GPL).
- Kecepatan dan Kapabilitas: MySQL di dalam XAMPP sangat ringan, cepat, dan mampu memproses data dalam jumlah yang sangat besar (jutaan record).

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

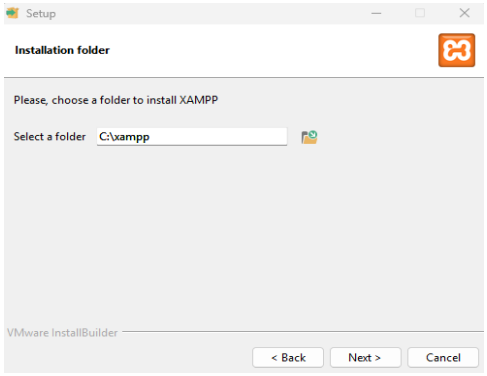
- Lintas Platform: XAMPP dapat dijalankan dengan stabil di berbagai sistem operasi, baik Windows, Linux, maupun macOS.

a. Instalasi XAMPP:

1. Unduh Installer: Kunjungi situs resmi Apache Friends (https://www.apachefriends.org/) dan unduh installer XAMPP terbaru sesuai dengan sistem operasi yang digunakan (misalnya Windows).
2. Mulai Instalasi: Klik ganda pada file installer yang telah diunduh. Jika muncul peringatan User Account Control (UAC) atau peringatan instalasi di folder C:\Program Files, klik OK untuk melanjutkan.
3. Pilih Komponen (Select Components): Pada layar pemilihan komponen, pastikan layanan inti untuk praktikum basis data yaitu Apache, MySQL, dan phpMyAdmin dalam keadaan tercentang. Klik Next.

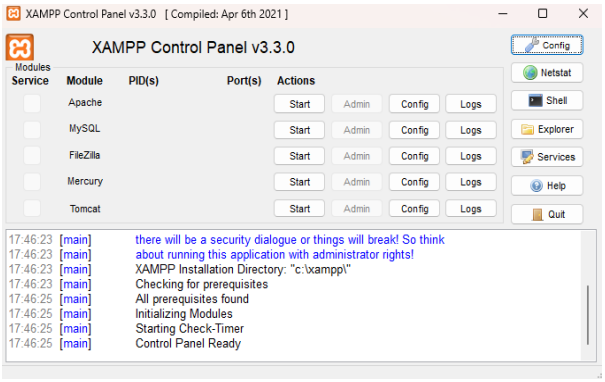


4. Pilih Lokasi Instalasi: Tentukan direktori tempat XAMPP akan diinstal. Sangat disarankan untuk membiarkannya pada pengaturan default, yaitu di C:\xampp. Klik Next.

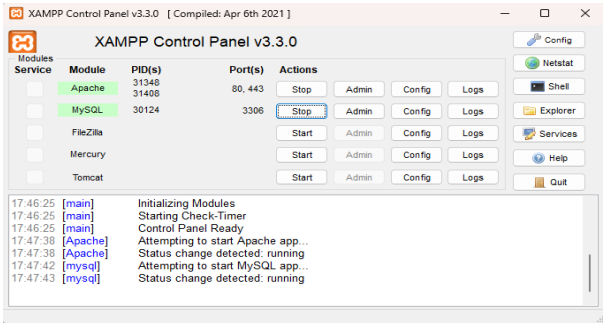


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

5. Proses Instalasi: Klik Next untuk memulai proses instalasi. Tunggu beberapa saat hingga proses penyalinan (extracting) seluruh file selesai.
6. Selesaikan Instalasi: Setelah proses selesai, pastikan opsi "Do you want to start the Control Panel now?" dicentang, kemudian klik Finish.
7. Jalankan Layanan (Start Service): Jendela XAMPP Control Panel akan terbuka. Untuk mulai menggunakan database, klik tombol Start pada baris Apache dan baris MySQL.



8. Verifikasi: Jika latar belakang tulisan Apache dan MySQL pada Control Panel sudah berubah warna menjadi hijau, berarti server web dan server database lokal telah berhasil berjalan dan siap digunakan.



b. Melakukan Koneksi ke MySQL :

Cara 1 (Melalui Command Prompt / CMD Windows) :

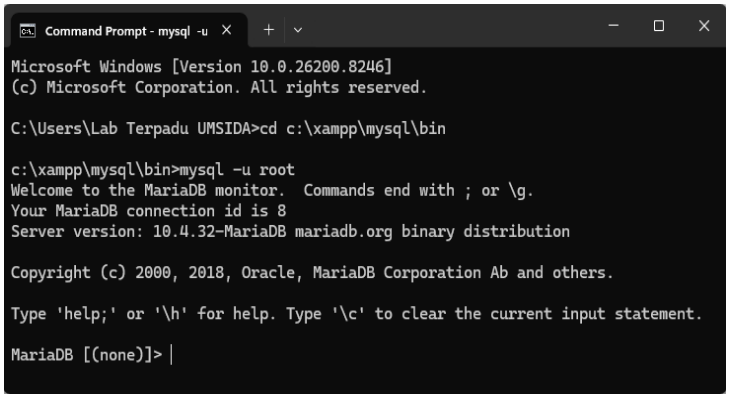
1. Buka aplikasi Command Prompt (CMD), kemudian masuk ke direktori utama MySQL bawaan XAMPP dengan cara sebagai berikut **cd c:\xampp\mysql\bin**
2. Setelah itu ketikkan perintah berikut untuk login ke database :

   C:*xampp\mysql\bin*> **mysql -u root**

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

(Catatan: Secara bawaan/default, instalasi XAMPP tidak menggunakan password untuk user root. Jika Anda menggunakan perintah mysql -u root -p, Anda cukup langsung menekan Enter saat dimintai password).

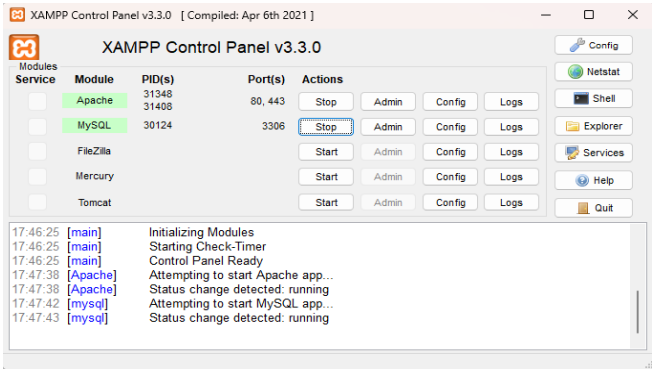
3. Selanjutnya akan ada respons dari server (biasanya bertuliskan Welcome to the MariaDB monitor...) seperti gambar berikut :



Tampilan tersebut di atas menandakan bahwa telah berhasil melakukan koneksi ke server.

Cara 2 (Melalui Shell XAMPP Control Panel) :

1. Buka XAMPP Control Panel dan pastikan module MySQL sudah dalam keadaan aktif (klik Start hingga latar belakang tulisan MySQL menjadi hijau).



2. Klik tombol Shell yang berada di menu deretan sebelah kanan pada jendela XAMPP Control Panel. Maka akan muncul jendela hitam command line.
3. Pada jendela Shell tersebut, ketikkan perintah berikut kemudian tekan Enter. **mysql -u root**

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

c. Merubah Prompt MySQL :

Rubahlah nama prompt mysql dengan nama dan nim masing-masing mahasiswa. Sintax :

```
Mysql> prompt prakDB/nama( 3 nim terakhir ) > (spasi)
(enter)
```

```
MariaDB [(none)]> prompt prakDB/Ferdy(060) >
PROMPT set to 'prakDB/Ferdy(060) > '
prakDB/Ferdy(060) >
```

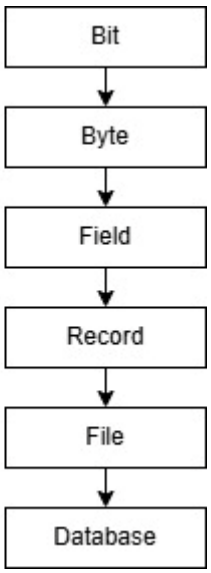

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

LEMBAR KERJA DAN TUGAS

1. Berdasarkan tingkat kompleksitas nilai data, tingkatan data dalam basis data dapat disusun dalam sebuah hierarki, mulai dari yang paling sederhana sampai yang paling kompleks. Gambarkan hierarki data hingga tersusun suatu basis data.

   Jawab :

   Ferdy_060



2. Jelaskan perbedaan antara tipe data CHAR dan VARCHAR serta berikan contoh penerapannya (contoh tidak boleh sama).

   Jawab : Tipe data CHAR memiliki panjang tetap sesuai ukuran yang telah ditentukan. Jika jumlah karakter yang dimasukkan kurang dari kapasitasnya, maka sistem akan menambahkan spasi secara otomatis hingga memenuhi ukuran tersebut. Oleh karena itu, CHAR cocok digunakan untuk data yang panjangnya selalu tetap, seperti jenis kelamin, plat nomor, golongan darah. Sedangkan VARCHAR merupakan tipe data karakter dengan panjang yang tidak tetap. Data yang disimpan akan menyesuaikan jumlah karakter yang dimasukkan tanpa menambahkan spasi tambahan. Karena penyimpanannya lebih fleksibel, VARCHAR lebih hemat ruang penyimpanan dan cocok digunakan untuk data yang panjangnya berbeda-beda, seperti judul film, nama hewan, nama perusahaan.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Contoh CHAR :

Data	Tipe	Contoh
Jenis kelamin	CHAR(1)	L, P
Plat Nomor	CHAR(8)	W 4444 HH
Golongan Darah	CHAR(2)	‘A’,’B’,’AB’,’O’

Contoh VARCHAR :

Data	Tipe	Contoh
Judul Film	VARCHAR(50)	Avengers Endgame
Nama Hewan	VARCHAR(30)	Kucing
Nama Perusahaan	VARCHAR(100)	PT. Pertamina

3. Jelaskan perbedaan antara konstanta bertipe numerik dengan konstanta bertipe karakter serta berikan contohnya (contoh tidak boleh sama).

   Jawab :

   Konstanta bertipe numerik adalah konstanta yang berupa angka dan digunakan untuk operasi perhitungan matematika. Konstanta numerik tidak ditulis menggunakan tanda petik. Konstanta ini dapat berupa bilangan bulat maupun bilangan desimal.

   Contoh : 10, 25, 3.14, 1000

   Sedangkan konstanta bertipe karakter adalah konstanta yang berupa huruf, simbol, atau teks. Konstanta karakter biasanya ditulis menggunakan tanda petik satu atau tanda petik dua.

   Contoh : ‘A’, ‘B’, ‘Ferdy’, ‘Sidoarjo’

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

4. Buatlah perintah MySQL berikut ini serta berikan contoh dan print screen hasil kompilasinya.

   a. Perintah fasilitas bantuan pada MySQL.

```
prakDB/Ferdy(060) > help;

General information about MariaDB can be found at
http://mariadb.org

List of all client commands:
Note that all text commands must be first on line and end with ';'
?         (\?) Synonym for `help'.
clear     (\c) Clear the current input statement.
connect   (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set statement delimiter.
edit      (\e) Edit command with $EDITOR.
ego       (\G) Send command to MariaDB server, display result vertically.
exit      (\q) Exit mysql. Same as quit.
go        (\g) Send command to MariaDB server.
help      (\h) Display this help.
nopager   (\n) Disable pager, print to stdout.
notee     (\t) Don't write into outfile.
pager     (\P) Set PAGER [to_pager]. Print the query results via PAGER.
print     (\p) Print current command.
prompt    (\R) Change your mysql prompt.
quit      (\q) Quit mysql.
rehash    (\#) Rebuild completion hash.
source    (\.) Execute an SQL script file. Takes a file name as an argument.
status    (\s) Get status information from the server.
system    (\!) Execute a system shell command.
tee       (\T) Set outfile [to_outfile]. Append everything into given outfile.
use       (\u) Use another database. Takes database name as argument.
charset   (\C) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.
warnings  (\W) Show warnings after every statement.
nowarning (\w) Don't show warnings after every statement.

For server side help, type 'help contents'
```

   b. Perintah menampilkan versi MySQL yang digunakan.

```
prakDB/Ferdy(060) > select version();
+-----------------+
| version()       |
+-----------------+
| 10.4.27-MariaDB |
+-----------------+
1 row in set (0.001 sec)
```

   c. Perintah untuk membuat database di MySQL.

```
prakDB/Ferdy(060) > Create Database BasisData;
Query OK, 1 row affected (0.002 sec)
```

   d. Perintah untuk mengaktifkan basis data yang akan digunakan.

```
prakDB/Ferdy(060) > Use BasisData;
Database changed
```

   e. Perintah menghasilkan tanggal sekarang dari sistem.

```
prakDB/Ferdy(060) > Select Curdate();
+------------+
| Curdate()  |
+------------+
| 2026-05-08 |
+------------+
1 row in set (0.000 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 – 2026

Lembar Asistensi

Praktikum Basis Data

Pokok Bahasan 3

Judul	: Data Definition Language (DDL)
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 6
Dilaksanakan	: 07 Mei 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
13/05 26	12/05 2026
(Ika Ratna Indra Astutik, S.Kom., MT.)	**(Zianisa Azzahra)**

POKOK BAHASAN 3

Data Definition Language (DDL)

A. PENDAHULUAN

Pada pokok bahasan ini akan dibahas mengenai data definition language pada SQL, dimana DDL digunakan untuk memanipulasi data dalam basis data.

B. TUJUAN

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Mahasiswa mampu memahami dan membuat basis data.
2. Mahasiswa mampu memahami dan membuat tabel dari basis data.
3. Mahasiswa mampu mengelolah dan memanipulasi basis data dan tabel-tabelnya.

C. PENYAJIAN (TUTORIAL)

A. Data Definiton Language (DLL)

DLL merupakan bagian dari sql yang digunakan untuk mendefinisikan struktur dan kerangka data dan obyek basis data. Bisa juga dikatakan merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut-atribut basis data, tabel, batasan-batasan terhadap suatu atribut, serta hubungan antar tabel.

Tabel 3.1 Perintah-perintah dalam DDL

Perintah	Keterangan
Create Database	Membuat basis data
Drop Database	Menghapus basis data
Create Table	Membuat tabel
Alter Table	Mengubah atau menyisipkan kolom ke dalam tabel
Drop Table	Menghapus tabel dari basis data
Create Index	Membuat Index
Drop Index	Menghapus Index

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

B. Perintah-perintah DDL

Berikut ini perintah-perintah sql untuk Data Definiton Language :

a. Membuat Database

Syntax :

```
CREATE DATABASE namadatabase;
```

Dimana :

Nama database yang dibuat tidak boleh mengandung spasi dan tidak boleh memiliki nama yang sama dengan database lain di MySQL. Berikut ini perintah untuk membuat basis data dengan nama **perpustakaan** :

```
mysql> create database perpustakaan;
```

```
prakDB/Ferdy(060) > create database perpustakaan;
Query OK, 1 row affected (0.001 sec)
```

b. Menampilkan daftar Database

Untuk menampilkan daftar basis data yang ada di Mysql dapat menggunakan perintah :

```
SHOW DATABASES;
```

Berikut ini perintah untuk menampilkan daftar basis data:

```
mysql> show databases;
```

```
prakDB/Ferdy(060) > SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| basisdata          |
| db_sekolah         |
| information_schema |
| klinik_sehat       |
| mysql              |
| namadatabase       |
| performance_schema |
| perpustakaan       |
| phpmyadmin         |
| test               |
+--------------------+
10 rows in set (0.001 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

c. Menghapus Database

Untuk melakukan penghapusan terhadap basis data yang sudah dibuat.

Syntax :

```
DROP DATABASE namadatabase;
```

Dimana :

Database yang akan dihapus harus sesuai dengan nama database. Berikut ini perintah untuk menghapus database dengan nama perpustakaan :

```
Mysql> drop database perpustakaan;
```

```
rakDB/Ferdy(060) > DROP DATABASE namadatabase;
uery OK, 0 rows affected (0.001 sec)
```

d. Mengaktifkan Database

Sebelum membuat suatu tabel, terlebih dahulu harus mengaktifkan database yang akan digunakan untuk menyimpan tabel-tabel tersebut dengan perintah :

```
USE namadatabase;
```

karena database yang sudah dibuat telah dihapus maka buat kembali database **perpustakaan**. Kemudian aktifkan database tersebut dengan perintah :

```
Mysql> use perpustakaan;
```

```
rakDB/Ferdy(060) > use perpustakaan;
Database changed
```

e. Membuat Tabel

Dalam basis data tabel atau field berfungsi untuk menyimpan record atau data. Untuk membuat table Syntaxnya adalah :

```
CREATE TABLE namatabel
(
Field1 TipeData1 ([lebar]),
Field2 TipeData2 ([lebar]),
...
Field3 TipeData3 ([lebar])
);
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Keterangan :

Nama tabel tidak boleh mengandung spasi (space) tetapi jika menginginkan ada spasi harus menggunakan tanda penghubung (nama_tabel). Field1 merupakan atribut pertama dan TipeData1 merupakan tipe data untuk atribut pertama. Jika ingin membuat tabel dengan atribut lebih dari satu, maka setelah pendefinisian tipe data sebelumnya diberikan tanda koma (,).

Berikut ini perintah untuk membuat tabel dengan nama **pengarang** :

```
mysql> create table pengarang (
      kode_pengarang varchar(5),
      nama_pengarang varchar(35));
```

```
prakDB/Ferdy(060) > CREATE TABLE pengarang (
    ->     kode_pengarang VARCHAR(5),
    ->     nama_pengarang VARCHAR(35)
    -> );
Query OK, 0 rows affected (0.016 sec)
```

Syntax tambahan :

Maka tabel pengarang telah terbentuk, untuk melihat hasilnya dapat digunakan perintah :

```
Mysql> SHOW TABLES;
```

```
prakDB/Ferdy(060) > SHOW TABLES;
+------------------------+
| Tables_in_perpustakaan |
+------------------------+
| pengarang              |
+------------------------+
1 row in set (0.001 sec)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Untuk melihat struktur tabel yang telah dibuat (dalam hal ini buku) syntaxnya adalah :

```
DESC namatabel;
```

Contoh:

```
Mysql> desc pengarang;
```

```
prakDB/Ferdy(060) > desc pengarang;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| kode_pengarang | varchar(5)  | YES  |     | NULL    |       |
| nama_pengarang | varchar(35) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
2 rows in set (0.017 sec)
```

f. Mendefinisikan null/not null

Ketika membuat tabel, beberapa field harus diatur agar field tertentu harus diisi. Biasanya field ini adalah sebagai field utama atau kunci, juga sebagai identikasi sehingga tidak boleh kosong.

Syntax :

```
CREATE TABLE namatabel
(
Field1 TipeData1 ([lebar]) NOT NULL,
Field2 TipeData2 ([lebar]) NOT NULL,
...
Field3 TipeData3 ([lebar])
);
```

Contoh:

```
mysql> create table pengarang (
      kode_pengarang varchar(5) not null,
      nama_pengarang varchar(35) not null);
```

```
prakDB/Ferdy(060) > CREATE TABLE pengarang (
    ->     kode_pengarang VARCHAR(5) NOT NULL,
    ->     nama_pengarang VARCHAR(35) NOT NULL
    -> );
Query OK, 0 rows affected (0.012 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

g. Mendefinisikan Nilai Bawaan (Default)

Nilai default adalah nilai yang otomatis diberikan oleh sistem untuk suatu atribut ketika ada penambahan baris baru, sementara nilai pada atribut tersebut tidak diisi oleh pengguna. Syntax :

```
CREATE TABLE namatabel
(
Field1 TipeData1 ([lebar]),
Field2 TipeData2 DEFAULT nilai
);
```

Dimana nilai adalah nilai default dari atribut tersebut.

Contoh:

```
Mysql> create table buku (
      Kode_buku varchar(5) not null,
      Judul_buku varchar(15) not null,
      harga integer default 0,
      tahun_terbit varchar(5),
      kode_pengarang varchar(5),
      kode_penerbit varchar(5));
```

```
prakDB/Faisal(063)> create table buku (
    -> Kode_buku varchar(5) not null,
    -> Judul_buku varchar(15) not null,
    -> Harga integer default 0,
    -> Tahun_terbit varchar(5),
    -> Kode_pengarang varchar(5),
    -> Kode_penerbit varchar(5));
Query OK, 0 rows affected (0.009 sec)
```

h. Menentukan kunci primer (Primary Key) Pada Tabel

Key adalah satu gabungan dari beberapa atribut yang dapat membedakan semua basis data (row) dalam tabel secara unik. Key di dalam database berfungsi sebagai suatu cara untuk mengidentifikasi dan menghubungkan satu tabel data dengan tabel yang lain.

Primary Key adalah suatu atribut atau satu set minimal atribut yang tidak hanya mendefinisikan secara unik suatu kejadian spesifik tetapi juga dapat mewakili setiap kejadian dari suatu kejadian.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Terdapat tiga cara untuk membuat primary key. Berikut ini adalah Syntax membuat primary key untuk Field1

Cara 1 :

```
CREATE TABLE namatabel
(
Field1 TipeData1 ([lebar]) NOT NULL PRIMARY KEY,
Field2 TipeData2 ([lebar])
);
```

Cara 2 :

```
CREATE TABLE namatabel
(
Field1 TipeData1 ([lebar]),
Field2 TipeData2 ([lebar]),
PRIMARY KEY(Field1)
);
```

Cara 3 :

```
ALTER TABLE namatabel ADD CONSTRAINT namaconstraint
PRIMARY KEY (namakolom);
```

Berikut ini perintah untuk membuat tabel pengarang dengan atribut kode_pengarang tipe datanya varchar(5), nama_pengarang tipe datanya varchar(15) dengan mendefinisikan nilai not null dan primary key untuk atribut kode_pengarang :

Contoh 1 :

```
Mysql> create table pengarang (
      Kode_pengarang varchar(5) not null primary key,
      Nama_pengarang varchar(15) not null);
```

```
prakDB/Ferdy(060) > CREATE TABLE pengarang (
    ->     kode_pengarang VARCHAR(5) NOT NULL PRIMARY KEY,
    ->     nama_pengarang VARCHAR(15) NOT NULL
    -> );
Query OK, 0 rows affected (0.011 sec)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Contoh 2 :

```
Mysql> create table pengarang (
       Kode_pengarang varchar(5) not null,
       Nama_pengarang varchar(15) not null,
       Primary key (kode_pengarang));
```

```
prakDB/Ferdy(060) > CREATE TABLE pengarang (
    ->     kode_pengarang VARCHAR(5) NOT NULL,
    ->     nama_pengarang VARCHAR(15) NOT NULL,
    ->     PRIMARY KEY (kode_pengarang)
    -> );
Query OK, 0 rows affected (0.013 sec)
```

Contoh 3 :

```
Mysql> create table pengarang (
       Kode_pengarang varchar(5) not null,
       Nama_pengarang varchar(15) not null);
```

```
prakDB/Ferdy(060) > CREATE TABLE pengarang (
    ->     kode_pengarang VARCHAR(5) NOT NULL,
    ->     nama_pengarang VARCHAR(15) NOT NULL
    -> );
Query OK, 0 rows affected (0.010 sec)
```

penambahan primary key :

```
Mysql> alter table pengarang add constraint pk primary
key (kode_pengarang);
```

```
prakDB/Ferdy(060) > ALTER TABLE pengarang
    -> ADD CONSTRAINT pk
    -> PRIMARY KEY (kode_pengarang);
Query OK, 0 rows affected, 1 warning (0.028 sec)
Records: 0  Duplicates: 0  Warnings: 1
```

i. Menghapus Primary Key Pada Tabel

Perintah :

Cara 1 : Jika primary key dibuat dengan menggunakan alter table :

```
ALTER TABLE namatabel DROP CONSTRAINT namaconstraint;
```

Cara 2 : Jika primary key dibuat melalui create table :

```
ALTER TABLE namatabel DROP PRIMARY KEY;
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Berikut ini perintah yang digunakan untuk menghapus primary key pada tabel buku :

```
Mysql> alter table pengarang drop primary key;
```

```
prakDB/Ferdy(060) > alter table pengarang drop primary key;
Query OK, 0 rows affected (0.038 sec)
Records: 0  Duplicates: 0  Warnings: 0

prakDB/Ferdy(060) > desc pengarang;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| kode_pengarang | varchar(5)  | NO   |     | NULL    |       |
| nama_pengarang | varchar(15) | NO   |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
2 rows in set (0.015 sec)
```

j. Menentukan Foreign Key Pada Tabel

Foreign Key adalah satu set atribut atau set atribut sebagai key penghubung kedua tabel dan melengkapi satu relationship (hubungan) terhadap primary key yang menunjukan keinduknya. Jika sebuah primary key terhubungan ke table/entity lain, maka keberadaan primary key pada entity tersebut di sebut sebagai foreign key.

Untuk membuat foreign key, maka harus dipastikan bahwa tabel dan atribut yang dirujuk (tabel induk dari foreign key) sudah didefinisikan terlebih dahulu. Perintah yang digunakan sebagai berikut :

```
CREATE TABLE namatabel
(
Field1 TipeData1 ([lebar]),
Field2 TipeData2 ([lebar]),
FOREIGN KEY (Field2) REFERENCES namatabelinduk
      (namafieldinduk)ON UPDATE CASCADE
      ON DELETE NO ACTION
)
```

atau

```
ALTER TABLE namatabel ADD CONSTRAINT namaconstraint
FOREIGN KEY (namafield) REFERENCES namatabelinduk
(namafieldinduk) ON UPDATE CASCADE ON DELETE NO
ACTION;
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Berikut ini perintah untuk membuat tabel buku beserta kolom-kolomnya :

```
Mysql> create table buku (
       Kode_buku varchar(5) not null primary key,
       Judul_buku varchar(15) not null,
       harga integer default 0,
       tahun_terbit varchar(5),
       kode_pengarang varchar(5),
       kode_penerbit varchar(5),
       Foreign key(kode_pengarang) references
       pengarang(kode_pengarang) on update cascade on
       delete no action);
```

```
prakDB/Ferdy(060) > CREATE TABLE buku (
    ->     kode_buku VARCHAR(5) NOT NULL PRIMARY KEY,
    ->     judul_buku VARCHAR(15) NOT NULL,
    ->     harga INTEGER DEFAULT 0,
    ->     tahun_terbit VARCHAR(5),
    ->     kode_pengarang VARCHAR(5),
    ->     kode_penerbit VARCHAR(5)
    -> );
Query OK, 0 rows affected (0.146 sec)
```

atau

```
Mysql> create table buku (
       Kode_buku varchar(5) not null primary key,
       Judul_buku varchar(15) not null,
       harga integer default 0,
       tahun_terbit varchar(5),
       kode_pengarang varchar(5),
       kode_penerbit varchar(5));
```

```
prakDB/Faisal(063)> create table buku (
    -> kode_buku varchar(5) not null primary key,
    -> judul_buku varchar(15) not null,
    -> harga integer default 0,
    -> tahun_terbit varchar(5),
    -> kode_pengarang varchar(5),
    -> kode_penerbit varchar(5));
Query OK, 0 rows affected (0.013 sec)

prakDB/Faisal(063)> desc buku;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| kode_buku      | varchar(5)  | NO   | PRI | NULL    |       |
| judul_buku     | varchar(15) | NO   |     | NULL    |       |
| harga          | int(11)     | YES  |     | 0       |       |
| tahun_terbit   | varchar(5)  | YES  |     | NULL    |       |
| kode_pengarang | varchar(5)  | YES  |     | NULL    |       |
| kode_penerbit  | varchar(5)  | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
6 rows in set (0.013 sec)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
ALTER TABLE buku
ADD CONSTRAINT fk
FOREIGN KEY (kode_pengarang)
REFERENCES pengarang(kode_pengarang)
ON UPDATE CASCADE
ON DELETE NO ACTION;
```

```
prakDB/Ferdy(060) > ALTER TABLE buku
    -> ADD CONSTRAINT fk
    -> FOREIGN KEY (kode_pengarang)
    -> REFERENCES pengarang(kode_pengarang)
    -> ON UPDATE CASCADE
    -> ON DELETE NO ACTION;
Query OK, 0 rows affected (0.045 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

k. Menghapus Foreign Key

Foreign key yang sudah dibuat dapat di hapus dengan perintah :

```
ALTER TABLE namatabel DROP FOREIGN KEY namaconstraint;
```

Berikut ini perintah untuk menghapus foreign key pada tabel buku :

```
Mysql> alter table buku drop foreign key fk;
```

```
prakDB/Ferdy(060) > ALTER TABLE buku
    -> DROP FOREIGN KEY fk;
Query OK, 0 rows affected (0.011 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

l. Mengubah Struktur Tabel

Tabel yang sudah dibuat dapat dilakukan perubahan strukturnya seperti penambahan atribut (field), penghapusan atribut (field) bahkan mengganti lebar field dari tabel tersebut. Perintah yang digunakan adalah ALTER TABLE.

✓ **Menambah Atribut Baru Pada Tabel**

Syntax :

```
ALTER TABLE namatabel ADD fieldbaru tipe;
```

Dimana :

namatabel adalah nama tabel yang akan ditambah fieldnya. Fieldbaru adalah nama atribut yang akan ditambahkan, tipe adalah tipe data dari atribut yang akan ditambahkan. Berikut ini perintah

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

untuk menambah atribut keterangan dengan tipe data varchar(25) ke dalam tabel buku :

```
Mysql> ALTER TABLE buku
ADD keterangan VARCHAR(25);
```

```
prakDB/Ferdy(060) > ALTER TABLE buku
    -> ADD keterangan VARCHAR(25);
Query OK, 0 rows affected (0.010 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

✓ **Mengubah Tipe Data atau Lebar Atribut Pada Tabel**

Syntax :

```
ALTER TABLE namatabel MODIFY COLUMN field tipe;
```

Dimana :

namatabel adalah nama tabel yang akan diubah tipe data atau lebar atributnya. Field adalah atribut yang akan diubah tipe data atau lebarnya. Tipe adalah tipe data baru atau tipe data lama dengan lebar atribut yang berbeda. Berikut ini perintah untuk mengubah tipe data untuk atribut keterangan dengan char(20) :

```
mysql> alter table buku modify column keterangan
char(20);
```

```
prakDB/Ferdy(060) > ALTER TABLE buku
    -> MODIFY COLUMN keterangan CHAR(20);
Query OK, 0 rows affected (0.067 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

✓ **Mengubah Nama Atribut (Field) pada Tabel**

Syntax :

```
ALTER TABLE namatabel CHANGE COLUMN namalamafield
namabarufield tipedatanya;
```

Dimana :

namatabel adalah nama tabel yang akan diubah nama atributnya, namalamafield adalah atribut yang akan diganti namanya, namabarufield adalah nama baru atribut, tipedatanya adalah tipe

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

data dari atribut tersebut. Berikut ini perintah untuk mengubah nama atribut keterangan menjadi ket :

```
mysql> alter table buku change column keterangan
ket char(20);
```

```
prakDB/Ferdy(060) > ALTER TABLE buku
    -> CHANGE COLUMN keterangan ket CHAR(20);
Query OK, 0 rows affected (0.011 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

✓ **Menghapus Atribut (Field) Pada Tabel**

Syntax :

```
ALTER TABLE namatabel DROP COLUMN namakolom;
```

Berikut ini perintah untuk menghapus atribut ket pada tabel buku:

```
Mysql> alter table buku drop ket;
```

```
prakDB/Ferdy(060) > ALTER TABLE buku
    -> DROP COLUMN ket;
Query OK, 0 rows affected (0.008 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

m. Menghapus Tabel

Tabel sudah di buat dapat di hapus dengan menggunakan perintah DROP TABLE. Syntax sebagai berikut:

```
DROP TABLE namatabel;
```

Tabel yang akan dihapus sesuai dengan namatabel, berikut ini perintah untuk menghapus tabel dengan nama pengarang :

```
Mysql> drop table buku;
```

```
prakDB/Ferdy(060) > drop table buku;
Query OK, 0 rows affected (0.008 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

LEMBAR KERJA DAN TUGAS

1. Buatlah sebuah database sesuai dengan hasil analisa Anda pada tugas di pokok bahasan 1 ke dalam MySQL.

   Jawab :

```
create database kepegawaian;
```

```
prakDB/Ferdy(060) > create database kepegawaian;
Query OK, 1 row affected (0.001 sec)
```

   Aktifkan Database

```
prakDB/Ferdy(060) > use kepegawaian;
Database changed
```

2. Buatlah tabel-tabelnya beserta Primery key dan Foreign Keynya yang juga sesuai dengan hasil analisa tugas di pokok bahasan 1 ke dalam MySQL

   Jawab :

   Tabel pegawai

```
CREATE TABLE pegawai (
    nip VARCHAR(5) NOT NULL PRIMARY KEY,
    nama VARCHAR(25) NOT NULL,
    kelamin ENUM('L', 'P'),
    tmp_lahir VARCHAR(20),
    tgl_lahir DATE,
    alamat VARCHAR(50),
    kabupaten VARCHAR(25),
    telpon VARCHAR(15),
    pend_akhir VARCHAR(10),
    kd_jabatan VARCHAR(5),
    gol VARCHAR(5),
    jenis ENUM('T', 'K')
);
```

```
prakDB/Ferdy(060) > CREATE TABLE pegawai (
    ->     nip VARCHAR(5) NOT NULL PRIMARY KEY,
    ->     nama VARCHAR(25) NOT NULL,
    ->     kelamin ENUM('L', 'P'),
    ->     tmp_lahir VARCHAR(20),
    ->     tgl_lahir DATE,
    ->     alamat VARCHAR(50),
    ->     kabupaten VARCHAR(25),
    ->     telpon VARCHAR(15),
    ->     pend_akhir VARCHAR(10),
    ->     kd_jabatan VARCHAR(5),
    ->     gol VARCHAR(5),
    ->     jenis ENUM('T', 'K')
    -> );
Query OK, 0 rows affected (0.013 sec)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
ALTER TABLE pegawai
ADD CONSTRAINT unique_key
UNIQUE (nip);
```

```
prakDB/Ferdy(060) > ALTER TABLE pegawai
    -> ADD CONSTRAINT unique_key
    -> UNIQUE (nip);
Query OK, 0 rows affected (0.015 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
ALTER TABLE pegawai
ADD CONSTRAINT fk
FOREIGN KEY (nip)
REFERENCES pegawai(nip)
ON UPDATE CASCADE
ON DELETE NO ACTION;
```

```
prakDB/Ferdy(060) > ALTER TABLE pegawai
    -> ADD CONSTRAINT fk
    -> FOREIGN KEY (nip)
    -> REFERENCES pegawai(nip)
    -> ON UPDATE CASCADE
    -> ON DELETE NO ACTION;
Query OK, 0 rows affected (0.055 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Tabel jabatan

```
CREATE TABLE jabatan (
    kode VARCHAR(5) NOT NULL PRIMARY KEY,
    jabatan VARCHAR(35) NOT NULL
);
```

```
prakDB/Ferdy(060) > CREATE TABLE jabatan (
    ->     kode VARCHAR(5) NOT NULL PRIMARY KEY,
    ->     jabatan VARCHAR(35) NOT NULL
    -> );
Query OK, 0 rows affected (0.011 sec)
```

Tabel golongan

```
CREATE TABLE golongan (
    id_golongan VARCHAR(5) NOT NULL PRIMARY KEY,
    golongan VARCHAR(35) NOT NULL
);
```

```
prakDB/Ferdy(060) > CREATE TABLE golongan (
    ->     id_golongan VARCHAR(5) NOT NULL PRIMARY KEY,
    ->     golongan VARCHAR(35) NOT NULL
    -> );
Query OK, 0 rows affected (0.011 sec)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Tabel absensi

```
CREATE TABLE absensi (
    id_absensi INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    tgl_absen DATE,
    nip VARCHAR(5),
    jam_masuk TIME,
    jam_keluar TIME,
    keterangan VARCHAR(15),

    CONSTRAINT fk_pegawai
    FOREIGN KEY (nip)
    REFERENCES pegawai(nip)
    ON UPDATE CASCADE
);
```

```
prakDB/Ferdy(060) > CREATE TABLE absensi (
    ->     id_absensi INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ->     tgl_absen DATE,
    ->     nip VARCHAR(5),
    ->     jam_masuk TIME,
    ->     jam_keluar TIME,
    ->     keterangan VARCHAR(15),
    ->
    ->     CONSTRAINT fk_pegawai
    ->     FOREIGN KEY (nip)
    ->     REFERENCES pegawai(nip)
    ->     ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.024 sec)
```

3. Pegawai di Universitas Muhammadiyah Sidoarjo juga dibedakan berdasarkan bagian yang meliputi : Rektorat, Biro Administrasi Akademik (BAA), Biro Administrasi Keuangan (BAK), Biro Administrasi Umum (BAU), Perpustakaan, PMB, LPPM, BPM, Pusdakom. Buatlah tabel dengan ketentuan diatas.

   Jawab :

   Menambahkan tabel baru yaitu tabel bagian

NO.	Field	Type	Size	Keterangan
1.	kd_bagian	Varchar	5	Primary key
2.	nama_bagian	varchar	50	

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
CREATE TABLE bagian (
    kd_bagian VARCHAR(5) NOT NULL PRIMARY KEY,
    nama_bagian VARCHAR(35) NOT NULL
);
```

```
prakDB/Ferdy(060) > CREATE TABLE bagian (
    ->     kd_bagian VARCHAR(5) NOT NULL PRIMARY KEY,
    ->     nama_bagian VARCHAR(35) NOT NULL
    -> );
Query OK, 0 rows affected (0.014 sec)
```

4. Relasikan tabel pegawai dengan tabel bagian serta lihat perubahan tabelnya.

Jawab :

```
ALTER TABLE pegawai
ADD kd_bagian VARCHAR(5);
```

```
prakDB/Ferdy(060) > ALTER TABLE pegawai
    -> ADD kd_bagian VARCHAR(5);
Query OK, 0 rows affected (0.011 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
ALTER TABLE pegawai
ADD CONSTRAINT fk_bagian
FOREIGN KEY (kd_bagian)
REFERENCES bagian(kd_bagian)
ON UPDATE CASCADE
ON DELETE NO ACTION;
```

```
prakDB/Ferdy(060) > ALTER TABLE pegawai
    -> ADD CONSTRAINT fk_bagian
    -> FOREIGN KEY (kd_bagian)
    -> REFERENCES bagian(kd_bagian)
    -> ON UPDATE CASCADE
    -> ON DELETE NO ACTION;
Query OK, 0 rows affected (0.050 sec)
Records: 0  Duplicates: 0  Warnings: 0
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Lihat perubahan table pegawai

```
prakDB/Ferdy(060) > desc pegawai;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| nip        | varchar(5)   | NO   | PRI | NULL    |       |
| nama       | varchar(25)  | NO   |     | NULL    |       |
| kelamin    | enum('L','P') | YES  |     | NULL    |       |
| tmp_lahir  | varchar(20)  | YES  |     | NULL    |       |
| tgl_lahir  | date         | YES  |     | NULL    |       |
| alamat     | varchar(50)  | YES  |     | NULL    |       |
| kabupaten  | varchar(25)  | YES  |     | NULL    |       |
| telpon     | varchar(15)  | YES  |     | NULL    |       |
| pend_akhir | varchar(10)  | YES  |     | NULL    |       |
| kd_jabatan | varchar(5)   | YES  |     | NULL    |       |
| gol        | varchar(5)   | YES  |     | NULL    |       |
| jenis      | enum('T','K') | YES  |     | NULL    |       |
| kd_bagian  | varchar(5)   | YES  | MUL | NULL    |       |
+------------+--------------+------+-----+---------+-------+
13 rows in set (0.017 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 – 2026

Lembar Asistensi

Praktikum Basis Data

Pokok Bahasan 4

Judul	: Data Manipulation Language (DML)
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 6
Dilaksanakan	: 02 Juli 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
9/07/26	6/7/2026
(Ika Ratna Indra Astutik, S.Kom., MT.)	**(Zianisa Azzahra)**

POKOK BAHASAN 4

Data Manipulation Language (DML)

A. PENDAHULUAN

Pada pokok bahasan ini akan dibahas mengenai data manipulation language (DML), dimana data pada basis data dapat di kelolah dan dimanipulasi dengan menggunakan perintah insert, select, update dan delete.

B. TUJUAN

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Mahasiswa mampu memasukkan data ke tabel di MySQL.
2. Mahasiswa mampu memanipulasi data dalam basis data di MySQL.
3. Mahasiswa mampu melakukan query dalam basis data di MySQL.

C. PENYAJIAN (TUTORIAL)

A. Data Manipulation Language (DML)

Data Manipulation Language (DDL) merupakan perintah-perintah yang berfungsi untuk melakukan manipulasi data ataupun objek-objek yang ada didalam tabel. Antara lain: perintah untuk memilih data (query), menyisipkan, mengubah dan menghapus data dalam basis data. Bentuk manipulasi yang dapat dilakukan oleh DML diantaranya adalah :

1. Melakukan pencarian kembali data lama,
2. Penyisipan data baru ke dalam tabel
3. Penghapusan data
4. Pengubahan data
5. Menampilkan data dengan kreiteria tertentu
6. Menampilkan data secara terurut.

DML menurut jenisnya dapat dibagi menjadi 2 jenis yaitu :

1. Prosedural, DML membutuhkan pemakai untuk menspesifikasikan data apa yang dibutuhkan dan bagaimana cara mendapatkannya, Contoh paket bahasa prosedural adalah dBase III, FoxBase.
2. Non Prosedural, DML membutuhkan pemakai untuk menspesifikasikan data apa yang dibutuhkan tanpa tahu bagaimana cara mendapatkannya. Contoh paket bahasa non prosedural adalah SQL (Structured Query Language) atau Query By Example (QBE).

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

B. Perintah DML sebagai berikut :

a. INSERT

Perintah INSERT digunakan untuk menambahkan baris pada suatu tabel. Terdapat dua cara untuk menambah baris, yaitu :

Cara 1 :

Menambah baris dengan mengisi data langsung pada setiap kolom tanpa menyertakan struktur tabel :

```
Mysql>INSERT INTO namatabel VALUES (nilail, nilai2, nilai-n);
```

Cara 2:

Menambah baris dengan menyertakan strnktur tabel dalam mengisi data pada setiap kolom:

```
Mysql> INSERT INTO namatabel (koloml, kolom2, kolom-n) VALUES (nilail, nilai2, nilai-n);
```

Berikut ini perintah untuk menambahkan baris pada tabel buku:

Cara 1:

```
PrakDB/Ferdy(060)> INSERT INTO buku VALUES ('B001', 'sistem basis data', '25000', '2004', 'P001', 'T001');
```

```
PrakDB/Ferdy(060)>INSERT INTO buku VALUES ('B001', 'sistem basis data', '25000', '2004', 'P001', 'T001');
Query OK, 1 row affected, 1 warning (0.004 sec)
```

Cara 2:

```
PrakDB/Ferdy(060)> INSERT INTO buku (kode_buku, judul_buku, harga, tahun_terbit, kode_pengarang, kode_penerbit) VALUES ('B002', 'sistem informasi', '50000', '2003', 'P001', 'T001');
```

```
PrakDB/Fikri(027) > INSERT INTO buku (kode_buku, judul_buku, harga, tahun_terbit, kode_pengarang, kode_penerbit) VALUES
('B002', 'sistem informasi', '50000', '2003', 'P001', 'T001');
Query OK, 1 row affected (0.006 sec)
```

Keterangan:

Jika data bertipe string, date atau time (contoh : B001, Sistem Basis Data, 2007-11-10) maka pemberian nilainya diapit dengan tanda petik tunggal (‘B001’) atau petik ganda (“B001”). Jika data bertipe numerik (2500, 400) maka pemberian nilainya tidak diapit tanda petik tunggal maupun ganda.

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

b. UPDATE

Perintah UPDATE digunakan untuk mengubah isi data pada satu atau beberapa kolom pada suatu tabel.

Syntax:

```
UPDATE namatabel SET koloml = nilail, kolom2 = nilai2
[WHERE
```

Perintah dalam tanda [] bersifat opsional untuk mengubah suatu baris dengan suatu kondisi tertentu. Berikut ini perintah untuk mengubah baris pada tabel pengarang dengan data sebagai berikut :

```
PrakDB/Ferdy(060)> SELECT * FROM buku;
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku;
+-----------+-----------------+-------+--------------+----------------+---------------+
| kode_buku | judul_buku      | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+-----------------+-------+--------------+----------------+---------------+
| B001      | sistem basis da | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+-----------+-----------------+-------+--------------+----------------+---------------+
2 rows in set (0.000 sec)
```

Contoh 1 : mengubah semua nilai pada kolom judul_ buku menjadi 'Basis Data'

```
PrakDB/Ferdy(060)> UPDATE buku SET judul_buku='basis
data';
```

```
PrakDB/Ferdy(060)>UPDATE buku SET judul_buku='basis data';
Query OK, 2 rows affected (0.002 sec)
Rows matched: 2  Changed: 2  Warnings: 0

PrakDB/Ferdy(060)>select * from buku;
+-----------+------------+-------+--------------+----------------+---------------+
| kode_buku | judul_buku | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+------------+-------+--------------+----------------+---------------+
| B001      | basis data | 25000 | 2004         | P001           | T001          |
| B002      | basis data | 50000 | 2003         | P001           | T001          |
+-----------+------------+-------+--------------+----------------+---------------+
2 rows in set (0.001 sec)
```

Contoh 2 : mengubah nilai pada kolom judul_buku menjadi Basis Data Terpadu dimana nilai pada kolom kode_buku adalah B001 :

```
PrakDB/Ferdy(060)> UPDATE buku SET judul_buku='basis data
terpadu' WHERE kode_buku='B001';
```

```
PrakDB/Ferdy(060)>UPDATE buku SET judul_buku='basis data terpadu' WHERE kode_buku='B001';
Query OK, 1 row affected, 1 warning (0.006 sec)
Rows matched: 1  Changed: 1  Warnings: 1

PrakDB/Ferdy(060)>select * from buku;
+-----------+-----------------+-------+--------------+----------------+---------------+
| kode_buku | judul_buku      | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+-----------------+-------+--------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+-----------+-----------------+-------+--------------+----------------+---------------+
2 rows in set (0.000 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

c. SELECT

Perintah SELECT digunakan untuk menampilkan isi dari suatu tabel yang dapat dihubungkan dengan tabel yang lainnya.

1) Menampilkan data untuk semua kolom menggunakan asterisk (*)

Syntax :

```
SELECT * FROM nama tabel;
```

Berikut ini perintah untuk menampilkan semua data pada tabel buku:

```
PrakDB/Ferdy(060)> SELECT * FROM buku;
```

```
PrakDB/Ferdy(060)>select * from buku;
+-----------+----------------+-------+--------------+----------------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+----------------+-------+--------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+-----------+----------------+-------+--------------+----------------+---------------+
2 rows in set (0.000 sec)
```

2) Menampilkan data untuk kolom tertentu

Syntax:

```
SELECT koloml, kolom2, kolom - n FROM namatabel;
```

Berikut ini perintah untuk menampilkan data pada tabel buku dengan kolom yang ditampilkan adalah kolom kode _ buku :

```
PrakDB/Ferdy(060)> SELECT kode_buku FROM buku;
```

```
PrakDB/Ferdy(060)>SELECT kode_buku FROM buku;
+------------+
| kode_buku |
+------------+
| B001       |
| B002       |
+------------+
2 rows in set (0.000 sec)
```

3) Menampilkan data dengan kondisi data tertentu dengan klausa WHERE

Syntax:

```
SELECT * FROM namatabel WHERE kondisi;
```

Berikut ini perintah untuk menampilkan data pada tabel buku dimana nilai pada kolom kode buku adalah B001 :

```
PrakDB/Ferdy(060)> SELECT * FROM buku WHERE
kode_buku='B001';
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
PrakDB/Ferdy(060)>SELECT * FROM buku WHERE kode_buku='B001';
+-----------+----------------+-------+-------------+----------------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+----------------+-------+-------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
+-----------+----------------+-------+-------------+----------------+---------------+
1 row in set (0.000 sec)
```

Beberapa operator perbandingan yang dapat digunakan pada klausa WHERE selain "=" adalah : > (lebih dari), < (kurang dari), < > (tidak sama dengan), >= (lebih dari atau sama dengan), <= (kurang dari atau sama dengan). Adapun operator lain, yaitu : AND, OR, NOT, BETWEEN-AND, IN dan LIKE. Berikut ini data yang ada pada tabel pengarang:

```
Mysql> SELECT * FROM buku;
```

Contoh 1 : perintah untuk menampilkan data pada tabel buku dimana nilai harga berkisar dari 25000 hingga 50000 :

```
PrakDB/Ferdy(060)> SELECT * FROM buku WHERE harga
>= 25000 AND harga <= 50000;
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku WHERE harga >= 25000 AND harga <= 50000;
+-----------+----------------+-------+-------------+----------------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+----------------+-------+-------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+-----------+----------------+-------+-------------+----------------+---------------+
2 rows in set (0.000 sec)
```

Atau

```
PrakDB/Ferdy(060)> SELECT * FROM buku WHERE harga
BETWEEN 25000 AND 50000;
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku WHERE harga BETWEEN 25000 AND 50000;
+-----------+----------------+-------+-------------+----------------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+----------------+-------+-------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+-----------+----------------+-------+-------------+----------------+---------------+
2 rows in set (0.001 sec)
```

Contoh 2 : perintah untuk menampilkan data pada tabel buku dimana nilai harga sama dengan 25000 atau 50000:

```
PrakDB/Ferdy(060)> SELECT * FROM buku WHERE harga
= 25000 OR harga = 50000;
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku WHERE harga =
    -> 25000 OR harga = 50000;
+-----------+----------------+-------+-------------+----------------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+----------------+-------+-------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+-----------+----------------+-------+-------------+----------------+---------------+
2 rows in set (0.000 sec)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Atau

```
PrakDB/Ferdy(060)> SELECT * FROM buku WHERE harga
IN (25000, 50000);
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku WHERE harga IN (25000, 50000);
+-----------+----------------+-------+--------------+----------------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+----------------+-------+--------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+-----------+----------------+-------+--------------+----------------+---------------+
2 rows in set (0.000 sec)
```

Contoh 3 : perintah untuk menampilkan data pada tabel bukudimana nilai pada kolom judul_buku tidak sama dengan basis data:

```
PrakDB/Ferdy(060)> SELECT * FROM buku WHERE not
judul buku = 'basis data terpadu';
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku WHERE not judul_buku = 'basis data terpadu';
+-----------+----------------+-------+--------------+----------------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+----------------+-------+--------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+-----------+----------------+-------+--------------+----------------+---------------+
2 rows in set (0.000 sec)
```

Atau

```
PrakDB/Ferdy(060)> SELECT * FROM buku WHERE
judul buku <> 'basis data terpadu';
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku WHERE
    -> judul_buku <> 'basis data terpadu';
+-----------+----------------+-------+--------------+----------------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+----------------+-------+--------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+-----------+----------------+-------+--------------+----------------+---------------+
2 rows in set (0.000 sec)
```

Contoh 4 : Isi tabel buku

```
PrakDB/Ferdy(060)> SELECT * FROM buku;
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku;
+-----------+----------------+-------+--------------+----------------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+----------------+-------+--------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+-----------+----------------+-------+--------------+----------------+---------------+
2 rows in set (0.000 sec)
```

perintah untuk menampilkan data pada tabel buku dimana data pada kolom tertentu diawali dengan nilai tertentu, misalnya pada kolom judul_buku dimana diawali dengan karakter ‘B’ :

```
PrakDB/Ferdy(060)> SELECT * FROM buku WHERE
judul_buku LIKE 'B%';
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
PrakDB/Ferdy(060)>SELECT * FROM buku WHERE
    -> judul_buku LIKE 'B%';
+-----------+----------------+-------+--------------+----------------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+----------------+-------+--------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
+-----------+----------------+-------+--------------+----------------+---------------+
1 row in set (0.001 sec)
```

4) Memberikan nama lain pada kolom

Syntax:

```
SELECT namakolomlama AS namakolombaru FROM namatabel;
```

Berikut ini perintah untuk memberikan nama lain pada kolom judul_buku menjadi judul pada tabel pengarang :

```
PrakDB/Ferdy(060)> SELECT judul_buku AS judul FROM buku;
```

```
PrakDB/Ferdy(060)>SELECT judul_buku AS judul FROM
    -> buku;
+-----------------+
| judul           |
+-----------------+
| basis data terp |
| sistem informas |
+-----------------+
2 rows in set (0.000 sec)
```

5) Menggunakan alias untuk nama tabel

Syntax :

```
SELECT namaalias.jenis, namaalias.harga FROM namatabel namaalias;
```

Berikut ini perintah untuk memberikan alias pada tabel buku :

```
PrakDB/Ferdy(060)> SELECT j.judul_buku, j.harga FROM buku j;
```

```
PrakDB/Ferdy(060)>SELECT j.judul_buku, j.harga
    -> FROM buku j;
+-----------------+-------+
| judul_buku      | harga |
+-----------------+-------+
| basis data terp | 25000 |
| sistem informas | 50000 |
+-----------------+-------+
2 rows in set (0.000 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

6) Menampilkan data lebih dari dua tabel

Syntax:

```
SELECT namaalias.jenis, namaalias.harga FROM
namatabel namaalias;
```

Isi tabel pengarang:

```
PrakDB/Ferdy(060)> SELECT * FROM pengarang;
```

```
PrakDB/Ferdy(060)>SELECT * FROM pengarang;
+----------------+----------------+
| kode_pengarang | nama_pengarang |
+----------------+----------------+
| 12332          | juno           |
| P001           | Subianto       |
+----------------+----------------+
2 rows in set (0.000 sec)
```

Isi tabel buku:

```
PrakDB/Ferdy(060)> SELECT * FROM buku;
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku;
+-----------+----------------+-------+--------------+----------------+----------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+----------------+-------+--------------+----------------+----------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001           |
| B002      | sistem informas | 50000 | 2003         | P001           | T001           |
+-----------+----------------+-------+--------------+----------------+----------------+
2 rows in set (0.000 sec)
```

Berikut ini perintah untuk menampilkan semua data pada tabel pengarang dan buku:

```
PrakDB/Ferdy(060)> SELECT * FROM pengarang, buku;
```

```
PrakDB/Ferdy(060)>SELECT * FROM pengarang, buku;
+----------------+----------------+-----------+----------------+-------+--------------+----------------+----------------+
| kode_pengarang | nama_pengarang | kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+----------------+----------------+-----------+----------------+-------+--------------+----------------+----------------+
| 12332          | juno           | B001      | basis data terp | 25000 | 2004         | P001           | T001           |
| P001           | Subianto       | B001      | basis data terp | 25000 | 2004         | P001           | T001           |
| 12332          | juno           | B002      | sistem informas | 50000 | 2003         | P001           | T001           |
| P001           | Subianto       | B002      | sistem informas | 50000 | 2003         | P001           | T001           |
+----------------+----------------+-----------+----------------+-------+--------------+----------------+----------------+
4 rows in set (0.000 sec)
```

7) Operator comparison ANY dan ALL

a. Operator ANY digunakan berkaitan dengan subquery. Operator ini menghasilkan TRUE (benar) jika paling tidak salah satu perbandingan dengan hasil subquery menghasilkan nilai TRUE. Ilustrasinya :

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Gaji > ANY (S)

Jika subquery S menghasilkan Gl, G2, ... , Gn, maka kondisi di atas identik dengan:

(gaji > Gl) OR (gaji > G2) OR ... OR (gaji > Gn)

Contoh : perintah untuk menampilkan semua data pengarang

```
PrakDB/Ferdy(060) >SELECT * FROM buku WHERE harga > ANY (SELECT kode_pengarang FROM pengarang);
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku WHERE harga > ANY (SELECT kode_pengarang FROM pengarang);
+----------+-----------------+-------+-------------+----------------+---------------+
| kode_buku | judul_buku      | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+----------+-----------------+-------+-------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+----------+-----------------+-------+-------------+----------------+---------------+
2 rows in set (0.001 sec)
```

b. Operator ALL digunakan untuk melakukan perbandingan dengan subquery. Kondisi dengan ALL menghasilkan nilai TRUE (benar) jika subquery tidak menghasilkan apapun atau jika perbandingan menghasilkan TRUE untuk setiap nilai query terhadap hasil subquery. Contoh: perintah untuk menampilkan data pengarang yang harganya paling Tinggi

```
PrakDB/Ferdy(060)> SELECT * FROM buku WHERE harga >= ALL (SELECT kode_pengarang FROM pengarang);
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku WHERE harga > ANY (SELECT kode_pengarang FROM pengarang);
+----------+-----------------+-------+-------------+----------------+---------------+
| kode_buku | judul_buku      | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+----------+-----------------+-------+-------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+----------+-----------------+-------+-------------+----------------+---------------+
2 rows in set (0.001 sec)
```

8) Aggregate Functions (COUNT, SUM, AVG, MIN, MAX)

a) COUNT

Perintah yang digunakan untuk menghitung jumlah baris suatu kolom pada tabel. Contoh : perintah untuk menghitung jumlah baris kolom kode _ buku pada tabel buku :

```
PrakDB/Ferdy(060)> SELECT COUNT(kode_buku) FROM buku;
```

```
PrakDB/Ferdy(060)>SELECT COUNT(kode_buku) FROM
    -> buku;
+-----------------+
| COUNT(kode_buku) |
+-----------------+
|               2 |
+-----------------+
1 row in set (0.001 sec)
```

b) SUM

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Perintah yang digunakan untuk menghitung jumlah nilai suatu kolom pada tabel Contoh : perintah untuk menghitung jumlah nilai kolom harga pada tabel pengarang:

```
PrakDB/Ferdy(060)> SELECT SUM(harga) FROM
buku;
```

```
PrakDB/Ferdy(060)>SELECT SUM(harga) FROM buku;
+------------+
| SUM(harga) |
+------------+
|      75000 |
+------------+
1 row in set (0.001 sec)
```

c) AVG

Perintah yang digunakan untuk menghitung rata-rata dari nilai suatu kolom pada tabel. Contoh: perintah untuk menghitung rata- rata dari kolom harga pada tabel pengarang:

```
PrakDB/Ferdy(060)> SELECT AVG(harga) FROM
buku;
```

```
PrakDB/Ferdy(060)>SELECT AVG(harga) FROM
    -> buku;
+-------------+
| AVG(harga) |
+-------------+
| 37500.0000 |
+-------------+
1 row in set (0.000 sec)
```

d) MIN

Perintah yang digunakan untuk menampilkan nilai terkecil dari suatu kolom pada tabel. Contoh : perintah untuk menampilkan nilai terkecil dari kolom harga pada tabel buku:

```
PrakDB/Ferdy(060)> SELECT MIN(harga) FROM
buku;
```

```
PrakDB/Ferdy(060)>SELECT MIN(harga) FROM buku;
+-------------+
| MIN(harga) |
+-------------+
|       25000 |
+-------------+
1 row in set (0.000 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

e) MAX

Perintah yang digunakan untuk menampilkan nilai terbesar dari suatu kolom pada tabel. Contoh : perintah untuk menampilkan nilai terbesar dari kolom harga pada tabel buku:

```
PrakDB/Ferdy(060)> SELECT MAX(harga) FROM buku;
```

```
PrakDB/Ferdy(060)> SELECT MAX(harga) FROM buku;
+-------------+
| MAX(harga) |
+-------------+
|       50000 |
+-------------+
1 row in set (0.000 sec)
```

9) SQL dengan (GROUP BY dan HAVING)

Klausa GROUP BY digunakan untuk melakukan pengelompokan data. Sebagai contoh, terdapat tabel buku dengan data sebagai berikut:

```
PrakDB/Ferdy(060)> SELECT * FROM buku;
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku;
+-----------+-----------------+-------+--------------+----------------+---------------+
| kode_buku | judul_buku      | harga | tahun_terbit | kode_pengarang | kode_penerbit |
+-----------+-----------------+-------+--------------+----------------+---------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |
+-----------+-----------------+-------+--------------+----------------+---------------+
2 rows in set (0.000 sec)
```

akan ditampilkan hanya kolom tahun_masuk dan digabungkan dengan SUM (jml_buku) yang dikelompokkan berdasarkan kolom tahun rnasuk pada tabel buku:

```
PrakDB/Ferdy(060)> SELECT SUM(jml_buku) FROM buku
GROUP BY tahun_penerbit;
```

```
PrakDB/Ferdy(060)>SELECT SUM(jml_buku) FROM buku GROUP BY tahun_penerbit;
+---------------+
| SUM(jml_buku) |
+---------------+
|            10 |
|            15 |
+---------------+
2 rows in set (0.001 sec)
```

Klausa HAVING digunakan untuk menentukan kondisi bagi klausa GROUP BY. Kelompok yang memenuhi HAVING saja yang akan dihasilkan. Contoh: perintah untuk menampilkan data hanya kolom tahun_masuk yang dikelompokkan berdasarkan kolom tahun_masuk, dimana jumlah buku berdasarkan

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

kelompoknya harus lebih besar dari satu pada tabel buku:

```
PrakDB/Ferdy(060)> SELECT kode_pengarang FROM buku
GROUP BY kode_pengarang HAVING COUNT(kode_buku) >
1;
```

```
PrakDB/Ferdy(060)>SELECT kode_pengarang FROM buku
    -> GROUP BY kode_pengarang HAVING COUNT(kode_buku) >
    -> 1;
+-----------------+
| kode_pengarang |
+-----------------+
| P001           |
+-----------------+
1 row in set (0.001 sec)
```

10) ORDER BY

Klausa ORDER BY digunakan untuk mengurutkan data berdasarkan kolom tertentu sesuai dengan tipe data yang dimiliki. Contoh: perintah untuk mengurntkan data buku berdasarkan kolom judul:

```
PrakDB/Ferdy(060)> SELECT * FROM buku ORDER BY
judul_buku;
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku ORDER BY
   -> judul_buku;
+----------+---------------+-------+--------------+----------------+---------------+---------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit | jml_buku | tahun_penerbit |
+----------+---------------+-------+--------------+----------------+---------------+---------+---------------+
| B001     | basis data terp | 25000 | 2004         | P001           | T001          |       10 | 2004          |
| B002     | sistem informas | 50000 | 2003         | P001           | T001          |       15 | 2005          |
+----------+---------------+-------+--------------+----------------+---------------+---------+---------------+
2 rows in set (0.001 sec)
```

atau tambahkan ASC untuk pengurutan secara ascending (menaik)

```
PrakDB/Ferdy(060)> SELECT * FROM buku ORDER BY
judul_buku ASC;
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku ORDER BY
   -> judul_buku ASC;
+----------+---------------+-------+--------------+----------------+---------------+---------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit | jml_buku | tahun_penerbit |
+----------+---------------+-------+--------------+----------------+---------------+---------+---------------+
| B001     | basis data terp | 25000 | 2004         | P001           | T001          |       10 | 2004          |
| B002     | sistem informas | 50000 | 2003         | P001           | T001          |       15 | 2005          |
+----------+---------------+-------+--------------+----------------+---------------+---------+---------------+
2 rows in set (0.000 sec)
```

atau tambahkan DESC untuk pengurutan secara descending (menurun)

```
PrakDB/Ferdy(060)> SELECT * FROM buku ORDER BY
judul_buku DESC;
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku ORDER BY
   -> judul_buku DESC;
+----------+---------------+-------+--------------+----------------+---------------+---------+---------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit | jml_buku | tahun_penerbit |
+----------+---------------+-------+--------------+----------------+---------------+---------+---------------+
| B002     | sistem informas | 50000 | 2003         | P001           | T001          |       15 | 2005          |
| B001     | basis data terp | 25000 | 2004         | P001           | T001          |       10 | 2004          |
+----------+---------------+-------+--------------+----------------+---------------+---------+---------------+
2 rows in set (0.000 sec)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

d. DELETE

Perintah DELETE digunakan untuk menghapus satu baris, baris dengan kondisi tertentu atau seluruh baris.

Syntax :

```
Mysql>DELETE FROM namatabel [WHERE kondisi];
```

Perintah dalam tanda [] bersifat opsional untuk menghapus suatu baris dengan suatu kondisi tertentu. Berikut ini perintah untuk menghapus baris pada tabel buku dengan data sebagai berikut:

```
PrakDB/Ferdy(060)> SELECT * FROM buku;
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku;
+-----------+-----------------+-------+--------------+----------------+---------------+----------+----------------+
| kode_buku | judul_buku      | harga | tahun_terbit | kode_pengarang | kode_penerbit | jml_buku | tahun_penerbit |
+-----------+-----------------+-------+--------------+----------------+---------------+----------+----------------+
| B001      | basis data terp | 25000 | 2004         | P001           | T001          |       10 | 2004           |
| B002      | sistem informas | 50000 | 2003         | P001           | T001          |       15 | 2005           |
+-----------+-----------------+-------+--------------+----------------+---------------+----------+----------------+
2 rows in set (0.000 sec)
```

Contoh 1 : jika ingin menghapus seluruh baris pada tabel pengarang :

```
PrakDB/Ferdy(060)> DELETE FROM buku;
```

```
PrakDB/Ferdy(060)>DELETE FROM buku;
Query OK, 2 rows affected (0.005 sec)

PrakDB/Ferdy(060)>select * from buku;
Empty set (0.000 sec)
```

Contoh 2 : jika ingin menghapus baris yang memiliki nilai 'B001' pada kolom kode_buku pada tabel buku maka perintahnya sebagai berikut:

```
PrakDB/Ferdy(060)> DELETE FROM buku WHERE kode_buku =
"B001";
```

```
PrakDB/Ferdy(060)>SELECT * FROM buku;
+-----------+-----------------+-------+--------------+----------------+---------------+----------+----------------+
| kode_buku | judul_buku      | harga | tahun_terbit | kode_pengarang | kode_penerbit | jml_buku | tahun_penerbit |
+-----------+-----------------+-------+--------------+----------------+---------------+----------+----------------+
| B002      | Sistem Informas | 50000 | 2003         | P001           | T001          |     NULL | NULL           |
+-----------+-----------------+-------+--------------+----------------+---------------+----------+----------------+
1 row in set (0.000 sec)
```

Contoh 3 : jika ingin menghapus baris yang memiliki nilai 'Basis Data Terpadu? pada kolom judul_buku pada tabel buku maka perintahnya sebagai berikut:

fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
PrakDB/Ferdy(060)> DELETE FROM buku WHERE judul_buku =
"basis data terpadu";
```

```
PrakDB/Ferdy(060)>DELETE FROM buku WHERE judul_buku
    -> = "basis data terpadu";
Query OK, 0 rows affected (0.000 sec)

PrakDB/Ferdy(060)>select * from buku;
+-----------+----------------+-------+--------------+----------------+---------------+----------+----------------+
| kode_buku | judul_buku     | harga | tahun_terbit | kode_pengarang | kode_penerbit | jml_buku | tahun_penerbit |
+-----------+----------------+-------+--------------+----------------+---------------+----------+----------------+
| B002      | Sistem Informas | 50000 | 2003         | P001           | T001          |     NULL | NULL           |
+-----------+----------------+-------+--------------+----------------+---------------+----------+----------------+
1 row in set (0.000 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

LEMBAR KERJA DAN TUGAS

a. Setelah Anda membuat basis data dan tabel sesuai dengan hasil analisa hasil pokok bahasan 3 maka masukkan data kedalam masing-masing tabel minimal 5 data (setiap mahasiswa tidak boleh sama datanya).

Jawab:

Tabel Pegawai:

```
PrakDB/Ferdy(060) > INSERT INTO pegawai (nip, nama,
kelamin, tmp_lahir, tgl_lahir, alamat, kabupaten, telpon,
pend_akhir, kd_jabatan, gol, jenis)
VALUES
('P001', 'Ferdyansyah', 'L', 'Sidoarjo', '2005-01-10',
'Candi', 'Sidoarjo', '085162914550', 'S1', 'J001', 'IVD',
'T'),
('P002', 'Ayu Lestari', 'P', 'Pasuruan', '2002-03-15',
'Bangil', 'Pasuruan', '081234567890', 'S1', 'J002',
'IIIC', 'K'),
('P003', 'Bambang S', 'L', 'Mojokerto', '1998-11-20',
'Pacet', 'Mojokerto', '081398765432', 'S2', 'J003',
'IVA', 'T'),
('P004', 'Siti Aminah', 'P', 'Surabaya', '2000-07-08',
'Waru', 'Sidoarjo', '085711223344', 'SMA', 'J004', 'IIB',
'K'),
('P005', 'Dedi Kurniawan', 'L', 'Gresik', '1995-12-30',
'Driyorejo', 'Gresik', '089677889900', 'S1', 'J005',
'IIID', 'T');
```

```
PrakDB/Ferdy(060)>INSERT INTO pegawai (nip, nama, kelamin, tmp_lahir, tgl_lahir, alamat, kabupaten, telpon, pend_akhir, kd_jabatan, gol, jenis)
    -> VALUES
    -> ('P001', 'Ferdyansyah', 'L', 'Sidoarjo', '2005-01-10', 'Candi', 'Sidoarjo', '085162914550', 'S1', 'J001', 'IVD', 'T'),
    -> ('P002', 'Ayu Lestari', 'P', 'Pasuruan', '2002-03-15', 'Bangil', 'Pasuruan', '081234567890', 'S1', 'J002', 'IIIC', 'K'),
    -> ('P003', 'Bambang S', 'L', 'Mojokerto', '1998-11-20', 'Pacet', 'Mojokerto', '081398765432', 'S2', 'J003', 'IVA', 'T'),
    -> ('P004', 'Siti Aminah', 'P', 'Surabaya', '2000-07-08', 'Waru', 'Sidoarjo', '085711223344', 'SMA', 'J004', 'IIB', 'K'),
    -> ('P005', 'Dedi Kurniawan', 'L', 'Gresik', '1995-12-30', 'Driyorejo', 'Gresik', '089677889900', 'S1', 'J005', 'IIID', 'T');
Query OK, 5 rows affected (0.013 sec)
Records: 5  Duplicates: 0  Warnings: 0

PrakDB/Ferdy(060)>select * from pegawai;
+------+--------------+---------+-----------+------------+-----------+-----------+--------------+------------+------------+-----+-------+-----------+
| nip  | nama         | kelamin | tmp_lahir | tgl_lahir  | alamat    | kabupaten | telpon       | pend_akhir | kd_jabatan | gol | jenis | kd_bagian |
+------+--------------+---------+-----------+------------+-----------+-----------+--------------+------------+------------+-----+-------+-----------+
| P001 | Ferdyansyah  | L       | Sidoarjo  | 2005-01-10 | Candi     | Sidoarjo  | 085162914550 | S1         | J001       | IVD | T     | NULL      |
| P002 | Ayu Lestari  | P       | Pasuruan  | 2002-03-15 | Bangil    | Pasuruan  | 081234567890 | S1         | J002       | IIIC | K    | NULL      |
| P003 | Bambang S    | L       | Mojokerto | 1998-11-20 | Pacet     | Mojokerto | 081398765432 | S2         | J003       | IVA | T     | NULL      |
| P004 | Siti Aminah  | P       | Surabaya  | 2000-07-08 | Waru      | Sidoarjo  | 085711223344 | SMA        | J004       | IIB | K     | NULL      |
| P005 | Dedi Kurniawan | L     | Gresik    | 1995-12-30 | Driyorejo | Gresik    | 089677889900 | S1         | J005       | IIID | T    | NULL      |
+------+--------------+---------+-----------+------------+-----------+-----------+--------------+------------+------------+-----+-------+-----------+
5 rows in set (0.001 sec)
```

Tabel Jabatan:

```
PrakDB/Ferdy(060) > INSERT INTO jabatan (kode, jabatan)
VALUES
('J001', 'IT Manager'),
('J002', 'HRD'),
('J003', 'Direktur Keuangan'),
('J004', 'Kepala Logistik'),
('J005', 'Customer Service');
```

```
PrakDB/Ferdy(060)>INSERT INTO jabatan (kode, jabatan) VALUES
    -> ('J001', 'IT Manager'),
    -> ('J002', 'HRD'),
    -> ('J003', 'Direktur Keuangan'),
    -> ('J004', 'Kepala Logistik'),
    -> ('J005', 'Customer Service');
Query OK, 5 rows affected (0.006 sec)
Records: 5  Duplicates: 0  Warnings: 0

PrakDB/Ferdy(060)>select * from jabatan;
+-------+--------------------+
| kode  | jabatan            |
+-------+--------------------+
| J001  | IT Manager         |
| J002  | HRD                |
| J003  | Direktur Keuangan  |
| J004  | Kepala Logistik    |
| J005  | Customer Service   |
+-------+--------------------+
5 rows in set (0.000 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Tabel Bagian:

```
PrakDB/Ferdy(060) > INSERT INTO bagian (kd_bagian, nama_bagian) VALUES
('K12', 'Teknologi Informasi'),
('B13', 'Recruitment'),
('D29', 'Akuntansi'),
('G22', 'Persediaan'),
('L22', 'Layanan Pelanggan');
```

```
PrakDB/Ferdy(060)>INSERT INTO bagian (kd_bagian, nama_bagian) VALUES
    -> ('K12', 'Teknologi Informasi'),
    -> ('B13', 'Recruitment'),
    -> ('D29', 'Akuntansi'),
    -> ('G22', 'Persediaan'),
    -> ('L22', 'Layanan Pelanggan');
Query OK, 5 rows affected (0.005 sec)
Records: 5  Duplicates: 0  Warnings: 0

PrakDB/Ferdy(060)>select * from bagian;
+-----------+---------------------+
| kd_bagian | nama_bagian         |
+-----------+---------------------+
| B13       | Recruitment         |
| D29       | Akuntansi           |
| G22       | Persediaan          |
| K12       | Teknologi Informasi |
| L22       | Layanan Pelanggan   |
+-----------+---------------------+
5 rows in set (0.000 sec)
```

Tabel Golongan:

```
PrakDB/Ferdy(060) > INSERT INTO golongan (id_golongan, golongan) VALUES
('047', 'Class 4'),
('021', 'Class 2'),
('012', 'Class 1'),
('058', 'Class 5'),
('082', 'Class 8');
```

```
PrakDB/Ferdy(060)>INSERT INTO golongan (id_golongan, golongan) VALUES
    -> ('047', 'Class 4'),
    -> ('021', 'Class 2'),
    -> ('012', 'Class 1'),
    -> ('058', 'Class 5'),
    -> ('082', 'Class 8');
Query OK, 5 rows affected (0.007 sec)
Records: 5  Duplicates: 0  Warnings: 0

PrakDB/Ferdy(060)>select * from golongan;
+-------------+----------+
| id_golongan | golongan |
+-------------+----------+
| 012         | Class 1  |
| 021         | Class 2  |
| 047         | Class 4  |
| 058         | Class 5  |
| 082         | Class 8  |
+-------------+----------+
5 rows in set (0.000 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Tabel Absensi:

```
PrakDB/Ferdy(060) > INSERT INTO absensi (id_absensi, tgl_absen, nip, jam_masuk, jam_keluar, keterangan)
VALUES
('01', '2023-07-06', 'P001', '07:30:00', '17:00:00', 'masuk'),
('02', '2023-07-06', 'P002', '09:00:00', '18:00:00', 'telat'),
('03', '2023-07-06', 'P003', '07:00:00', '17:00:00', 'masuk'),
('04', '2023-07-06', 'P004', '08:00:00', '17:00:00', 'masuk'),
('05', '2023-07-06', 'P005', '07:15:00', '17:00:00', 'masuk');
```

```
PrakDB/Ferdy(060)>INSERT INTO absensi (id_absensi, tgl_absen, nip, jam_masuk, jam_keluar, keterangan)
    -> VALUES
    -> ('01', '2023-07-06', 'P001', '07:30:00', '17:00:00', 'masuk'),
    -> ('02', '2023-07-06', 'P002', '09:00:00', '18:00:00', 'telat'),
    -> ('03', '2023-07-06', 'P003', '07:00:00', '17:00:00', 'masuk'),
    -> ('04', '2023-07-06', 'P004', '08:00:00', '17:00:00', 'masuk'),
    -> ('05', '2023-07-06', 'P005', '07:15:00', '17:00:00', 'masuk');
Query OK, 5 rows affected (0.006 sec)
Records: 5  Duplicates: 0  Warnings: 0

PrakDB/Ferdy(060)>select * from absensi;
+------------+------------+------+-----------+------------+------------+
| id_absensi | tgl_absen  | nip  | jam_masuk | jam_keluar | keterangan |
+------------+------------+------+-----------+------------+------------+
|          1 | 2023-07-06 | P001 | 07:30:00  | 17:00:00   | masuk      |
|          2 | 2023-07-06 | P002 | 09:00:00  | 18:00:00   | telat      |
|          3 | 2023-07-06 | P003 | 07:00:00  | 17:00:00   | masuk      |
|          4 | 2023-07-06 | P004 | 08:00:00  | 17:00:00   | masuk      |
|          5 | 2023-07-06 | P005 | 07:15:00  | 17:00:00   | masuk      |
+------------+------------+------+-----------+------------+------------+
5 rows in set (0.000 sec)
```

b. Berilah contoh penerapan manipulasi data pada database yang sudah Anda buat yang meliputi :

- Perintah Update

```
PrakDB/Ferdy(060) > UPDATE pegawai SET alamat='Sidoarjo' WHERE nama='Ferdyansyah';
'masuk');
```

```
PrakDB/Ferdy(060)>UPDATE pegawai SET alamat='Gempol' WHERE nama='Ferdyansyah';
Query OK, 1 row affected (0.005 sec)
Rows matched: 1  Changed: 1  Warnings: 0

PrakDB/Ferdy(060)>select * from pegawai;
+------+-----------------+---------+-----------+------------+-----------+-----------+--------------+------------+------------+------+-------+-----------+
| nip  | nama            | kelamin | tmp_lahir | tgl_lahir  | alamat    | kabupaten | telpon       | pend_akhir | kd_jabatan | gol  | jenis | kd_bagian |
+------+-----------------+---------+-----------+------------+-----------+-----------+--------------+------------+------------+------+-------+-----------+
| P001 | Ferdyansyah     | L       | Sidoarjo  | 2005-01-10 | Gempol    | Sidoarjo  | 085162914550 | S1         | J001       | IVD  | T     | NULL      |
| P002 | Ayu Lestari     | P       | Pasuruan  | 2002-03-15 | Bangil    | Pasuruan  | 081234567890 | S1         | J002       | IIIC | K     | NULL      |
| P003 | Bambang S       | L       | Mojokerto | 1998-11-20 | Pacet     | Mojokerto | 081398765432 | S2         | J003       | IVA  | T     | NULL      |
| P004 | Siti Aminah     | P       | Surabaya  | 2000-07-08 | Waru      | Sidoarjo  | 085711223344 | SMA        | J004       | IIB  | K     | NULL      |
| P005 | Dedi Kurniawan  | L       | Gresik    | 1995-12-30 | Driyorejo | Gresik    | 089677889900 | S1         | J005       | IIID | T     | NULL      |
+------+-----------------+---------+-----------+------------+-----------+-----------+--------------+------------+------------+------+-------+-----------+
5 rows in set (0.000 sec)
```

- Perintah Delete

```
PrakDB/Ferdy(060) > DELETE FROM absensi WHERE id_absensi='05';
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
PrakDB/Ferdy(060)>DELETE FROM absensi WHERE
    -> id_absensi='05';
Query OK, 1 row affected (0.005 sec)

PrakDB/Ferdy(060)>select * from absensi;
+------------+------------+------+-----------+------------+------------+
| id_absensi | tgl_absen  | nip  | jam_masuk | jam_keluar | keterangan |
+------------+------------+------+-----------+------------+------------+
|          1 | 2023-07-06 | P001 | 07:30:00  | 17:00:00   | masuk      |
|          2 | 2023-07-06 | P002 | 09:00:00  | 18:00:00   | telat      |
|          3 | 2023-07-06 | P003 | 07:00:00  | 17:00:00   | masuk      |
|          4 | 2023-07-06 | P004 | 08:00:00  | 17:00:00   | masuk      |
+------------+------------+------+-----------+------------+------------+
4 rows in set (0.000 sec)
```

- Perintah Select tanpa klausa

```
PrakDB/Ferdy(060) > DELETE FROM absensi WHERE
id_absensi='05';
```

```
PrakDB/Ferdy(060)>SELECT * FROM pegawai;
+-----+--------------+---------+-----------+------------+---------+-----------+--------------+-------------+-----+------+-----------+
| nip | nama         | kelamin | tmp_lahir | tgl_lahir  | alamat  | kabupaten | telpon       | pend_akhir  | kd_jabatan | gol | jenis | kd_bagian |
+-----+--------------+---------+-----------+------------+---------+-----------+--------------+-------------+-----+------+-----------+
| P001 | Ferdyansyah | L       | Sidoarjo  | 2005-01-10 | Gempol  | Sidoarjo  | 085162914550 | S1          | J001       | IVD | T     | NULL      |
| P002 | Ayu Lestari | P       | Pasuruan  | 2002-03-15 | Bangil  | Pasuruan  | 081234567890 | S1          | J002       | IIIC | K     | NULL      |
| P003 | Bambang S   | L       | Mojokerto | 1998-11-20 | Pacet   | Mojokerto | 0813987654321 | S2          | J003       | IVA | T     | NULL      |
| P004 | Siti Aminah | P       | Surabaya  | 2000-07-08 | Waru    | Sidoarjo  | 085711223344 | SMA         | J004       | IIB | K     | NULL      |
| P005 | Dedi Kurniawan | L    | Gresik    | 1995-12-30 | Driyorejo | Gresik  | 089677889900 | S1          | J005       | IIID | T     | NULL      |
+-----+--------------+---------+-----------+------------+---------+-----------+--------------+-------------+-----+------+-----------+
5 rows in set (0.000 sec)
```

- Perintah Select disertai klausa order by (urutkan berdasarkan ascending dan descending)

```
PrakDB/Ferdy(060) > SELECT * FROM absensi ORDER BY
id_absensi ASC;
```

```
PrakDB/Ferdy(060)>SELECT * FROM absensi ORDER BY
    -> id_absensi ASC;
+------------+------------+------+-----------+------------+------------+
| id_absensi | tgl_absen  | nip  | jam_masuk | jam_keluar | keterangan |
+------------+------------+------+-----------+------------+------------+
|          1 | 2023-07-06 | P001 | 07:30:00  | 17:00:00   | masuk      |
|          2 | 2023-07-06 | P002 | 09:00:00  | 18:00:00   | telat      |
|          3 | 2023-07-06 | P003 | 07:00:00  | 17:00:00   | masuk      |
|          4 | 2023-07-06 | P004 | 08:00:00  | 17:00:00   | masuk      |
+------------+------------+------+-----------+------------+------------+
4 rows in set (0.001 sec)
```

```
PrakDB/Ferdy(060) > SELECT * FROM absensi ORDER BY
id_absensi DESC;
```

```
PrakDB/Ferdy(060)>SELECT * FROM absensi ORDER BY
    -> id_absensi DESC;
+------------+------------+------+-----------+------------+------------+
| id_absensi | tgl_absen  | nip  | jam_masuk | jam_keluar | keterangan |
+------------+------------+------+-----------+------------+------------+
|          4 | 2023-07-06 | P004 | 08:00:00  | 17:00:00   | masuk      |
|          3 | 2023-07-06 | P003 | 07:00:00  | 17:00:00   | masuk      |
|          2 | 2023-07-06 | P002 | 09:00:00  | 18:00:00   | telat      |
|          1 | 2023-07-06 | P001 | 07:30:00  | 17:00:00   | masuk      |
+------------+------------+------+-----------+------------+------------+
4 rows in set (0.001 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

- Perintah Select disertai klausa group by

```
PrakDB/Ferdy(060) > SELECT keterangan FROM absensi GROUP
BY keterangan;
```

```
PrakDB/Ferdy(060)>SELECT keterangan FROM absensi GROUP
    -> BY keterangan;
+------------+
| keterangan |
+------------+
| masuk      |
| telat      |
+------------+
2 rows in set (0.001 sec)
```

- Perintah Select disertai klausa having

```
PrakDB/Ferdy(060) > SELECT nip FROM pegawai GROUP BY
kd_jabatan HAVING COUNT(kd_jabatan) >= 1;
```

```
PrakDB/Ferdy(060)>SELECT nip FROM pegawai GROUP BY kd_jabatan HAVING COUNT(kd_jabatan) >= 1;
+------+
| nip  |
+------+
| P001 |
| P002 |
| P003 |
| P004 |
| P005 |
+------+
5 rows in set (0.001 sec)
```

- Perintah Select disertai klausa Limit

```
PrakDB/Ferdy(060) > SELECT * FROM pegawai LIMIT 3;
```

```
PrakDB/Ferdy(060)>SELECT * FROM pegawai LIMIT 3;
+------+-------------+---------+-----------+------------+---------+-----------+--------------+-------------+-----------+------+-------+-----------+
| nip  | nama        | kelamin | tmp_lahir | tgl_lahir  | alamat  | kabupaten | telpon       | pend_akhir  | kd_jabatan | gol  | jenis | kd_bagian |
+------+-------------+---------+-----------+------------+---------+-----------+--------------+-------------+-----------+------+-------+-----------+
| P001 | Ferdyansyah | L       | Sidoarjo  | 2005-01-10 | Gempol  | Sidoarjo  | 085162914550 | S1          | J001      | IVD  | T     | NULL      |
| P002 | Ayu Lestari | P       | Pasuruan  | 2002-03-15 | Bangil  | Pasuruan  | 081234567890 | S1          | J002      | IIIC | K     | NULL      |
| P003 | Bambang S   | L       | Mojokerto | 1998-11-20 | Pacet   | Mojokerto | 081398765432 | S2          | J003      | IVA  | T     | NULL      |
+------+-------------+---------+-----------+------------+---------+-----------+--------------+-------------+-----------+------+-------+-----------+
3 rows in set (0.001 sec)
```

- Perintah Select disertai klausa where

```
PrakDB/Ferdy(060) > SELECT nip, nama FROM pegawai WHERE
nip = 'P001';
```

```
PrakDB/Ferdy(060)>SELECT nip, nama FROM pegawai WHERE nip = 'P001';
+------+-------------+
| nip  | nama        |
+------+-------------+
| P001 | Ferdyansyah |
+------+-------------+
1 row in set (0.000 sec)
```

Catatan : antar mahasiswa atau kelompok tidak boleh sama

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 – 2026

Lembar Asistensi

Praktikum Basis Data

Pokok Bahasan 5

Judul	: Query dan View
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 6
Dilaksanakan	: 02 Juli 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
9/07 26	6/7 2026
(Ika Ratna Indra Astutik, S.Kom., MT.)	**(Zianisa Azzahra)**

POKOK BAHASAN 5

QUERY DAN VIEW

A. PENDAHULUAN

Pada pokok bahasan ini akan dibahas mengenai query dan view dalam basis data. Query adalah permintaan untuk mengambil atau mengubah data yang ada dalam basis data. Biasanya, query ditulis dalam bahasa SQL (Structured Query Language). View adalah objek dalam basis data yang berfungsi untuk menyimpan query SQL yang kompleks dan memberikan tampilan data yang bersifat virtual.

B. TUJUAN

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Mengelolah data dengan kriteria tertentu.
2. Mengelolah data dari beberapa tabel.
3. Memahami dan membuat View
4. Dapat Memanggil data melalui View
5. Merubah definisi View
6. Insert, Update, dan Delete data melalui View
7. Menghapus (drop) view

C. PENYAJIAN (TUTORIAL)

A. Query

Query merupakan suatu proses pengolahan data yang digunakan untuk memberikan hasil dari basis data berdasarkan kriteria tertentu. Query tidak hanya membaca atau mengambil data, query biasanya melibatkan beberapa tabel yang direlasikan dengan menggunakan field kunci. Namun query juga dapat digunakan pada satu tabel saja, tetapi hasilnya kurang informatif dan terbatas.

1. Aturan dalam melakukan query antar tabel:
   a. Setiap field disebutkan bersama dengan nama tabelnya, dipisahkan tanda titik (.).

Syntax :

```
Namatabel.namafield.
```

Contoh :

```
buku.kode_buku artinya field kode_buku
dari tabel
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

b. Setiap tabel yang terlibat dalam proses query harus disebutkan dalam klausa FROM, dengan pemisah koma (,).Dimana urutan tabel tidak mempengaruhi proses query.

Contoh :
```
FROM buku, anggota.
```

c. Kondisi dalam klausa WHERE mempengaruhi jenis join yang tercipta.

2. Jenis-jenis join pada query :

a. Operator Cross Join

Operator ini berguna untuk melakukan operasi penggabungan dengan perkalian kartesain. Namun penggabungan jenis ini jarang digunakan karena tidak menghasilkan nilai informasi yang efektif.

Contoh:
```
PrakDB/Ferdy(060) > SELECT * FROM buku CROSS JOIN pengarang LIMIT 5;
```

```
PrakDb/Ferdy(060) >SELECT * FROM buku CROSS JOIN pengarang LIMIT 5;
+-----------+--------------+--------+--------------+----------------+----------------+----------+--------------+----------------+-----------------+
| kode_buku | judul_buku   | harga  | tahun_terbit | kode_pengarang | kode_penerbit  | jml_buku | tahun_penerbit | kode_pengarang | nama_pengarang  |
+-----------+--------------+--------+--------------+----------------+----------------+----------+--------------+----------------+-----------------+
| B002      | Sistem Informas | 50000 | 2003         | P001           | T001           |     NULL | NULL         | 12332          | juno            |
| B002      | Sistem Informas | 50000 | 2003         | P001           | T001           |     NULL | NULL         | P001           | Subianto        |
+-----------+--------------+--------+--------------+----------------+----------------+----------+--------------+----------------+-----------------+
2 rows in set (0.001 sec)
```

b. Operator Inner Join

Inner join digunakan untuk menampilkan data dari dua tabel yang berisi data sesuai dengan syarat dibelakang on (tidak boleh null), dengan kata lain semua data dari tabel kiri mendapat pasangan data dari tabel sebelah kanan. Berikut ini perintah untuk menampilkan data dari tabel pengarang dan buku dengan syarat berdasarkan kolom kode_pengarang :

```
PrakDB/Ferdy(060) > SELECT * FROM pengarang JOIN buku on (pengarang.kode_pengarang = buku.kode_pengarang);
```

```
PrakDb/Ferdy(060) >SELECT * FROM pengarang JOIN buku
    -> on (pengarang.kode_pengarang = buku.kode_pengarang);
+----------------+----------------+-----------+--------------+--------+--------------+----------------+----------------+----------+--------------+
| kode_pengarang | nama_pengarang | kode_buku | judul_buku   | harga  | tahun_terbit | kode_pengarang | kode_penerbit  | jml_buku | tahun_penerbit |
+----------------+----------------+-----------+--------------+--------+--------------+----------------+----------------+----------+--------------+
| P001           | Subianto       | B002      | Sistem Informas | 50000 | 2003         | P001           | T001           |     NULL | NULL         |
+----------------+----------------+-----------+--------------+--------+--------------+----------------+----------------+----------+--------------+
1 row in set (0.000 sec)
```

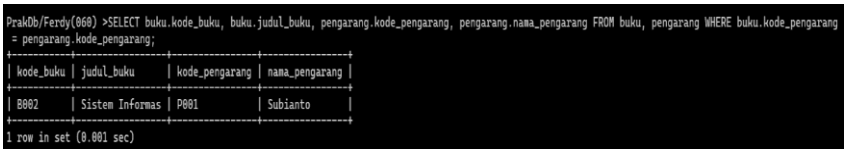
c. Operator Equijoin

Equijoin adalah penggabungan antar tabel dengan menggunakan operator ‘=’ pada kondisi klausa WHERE

Contoh :

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
PrakDB/Ferdy(060)    >    SELECT    buku.kode_buku,
buku.judul_buku,          pengarang.kode_pengarang,
pengarang.nama_pengarang FROM buku, pengarang WHERE
buku.kode_pengarang = pengarang.kode_pengarang;
```

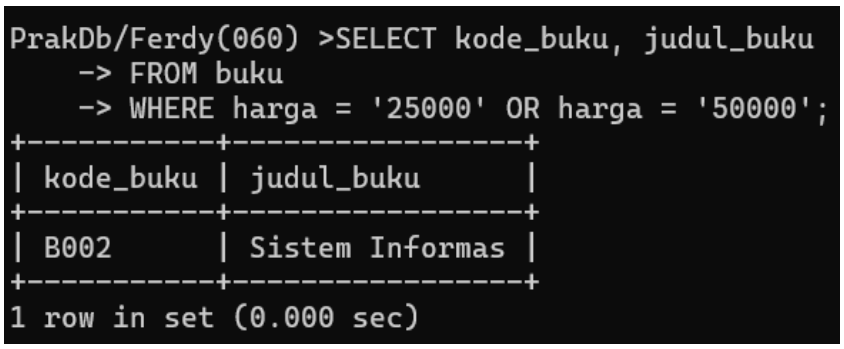


d. Operator Self-Join

Self-join adalah jenis penggabungan antar field dari tabel yang sama. Untuk melakukan penggabungan self-join menggunakan alias.

Contoh:

```
PrakDB/Ferdy(060) > SELECT kode_buku, judul_buku
FROM buku
WHERE harga = '25000' OR harga = '50000';
```

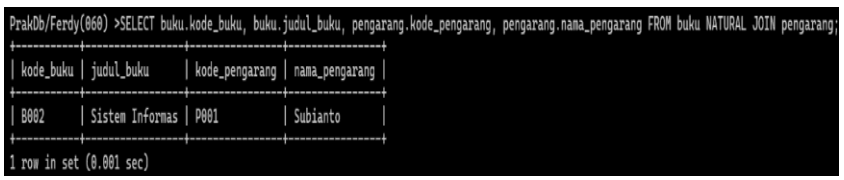


e. Operator Natural Join

Operator ini digunakan untuk melakukan operasi equijoin dengan memperlakukan nama-nama kolom yang sama sebagai kolom penghubung.

Contoh :

```
PrakDB/Ferdy(060)    >    SELECT    buku.kode_buku,
buku.judul_buku,          pengarang.kode_pengarang,
pengarang.nama_pengarang  FROM  buku  NATURAL  JOIN
pengarang;
```



fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Natural Join dibedakan menjadi 2 yaitu :

- Natural Left Join
  Natural left join digunakan untuk menampilkan semua data dari tabel sebelah kiri perintah natural left join beserta pasangannya dari tabel sebelah kanan. Meskipun terdapat data dari sebelah kiri tidak memiliki pasangan, tetap akan ditampilkan dengan pasangannya berupa nilai NULL.

```
PrakDB/Ferdy(060) > SELECT * FROM pengarang
NATURAL LEFT JOIN buku;
```

```
PrakDb/Ferdy(060) >SELECT * FROM pengarang NATURAL LEFT JOIN buku;
+----------------+-----------------+-----------+-------------+--------+--------------+---------------+---------+--------------+
| kode_pengarang | nama_pengarang  | kode_buku | judul_buku  | harga  | tahun_terbit | kode_penerbit | jml_buku | tahun_penerbit |
+----------------+-----------------+-----------+-------------+--------+--------------+---------------+---------+--------------+
| P001           | Subianto        | B002      | Sistem Informas | 50000 | 2003         | T001          |     NULL | NULL           |
| 12332          | juno            | B003      | Basis Data  | 75000  | 2026         | T001          |     NULL | NULL           |
+----------------+-----------------+-----------+-------------+--------+--------------+---------------+---------+--------------+
2 rows in set (0.000 sec)
```

- Natural Right Join

  Natural right join digunakan untuk menampilkan semua data dari table sebelah kanan perintah natural right join beserta pasangannya dari tabel sebelah kiri. Meskipun terdapat data dari sebelah kanan tidak memiliki pasangan, tetap akan ditampilkan dengan pasangannya berupa nilai NULL.

```
PrakDB/Ferdy(060) > SELECT * FROM pengarang
NATURAL RIGHT JOIN buku;
```

```
PrakDb/Ferdy(060) >SELECT * FROM pengarang
    -> NATURAL RIGHT JOIN buku;
+----------------+-----------+-------------+--------+--------------+---------------+---------+--------------+-----------------+
| kode_pengarang | kode_buku | judul_buku  | harga  | tahun_terbit | kode_penerbit | jml_buku | tahun_penerbit | nama_pengarang |
+----------------+-----------+-------------+--------+--------------+---------------+---------+--------------+-----------------+
| P001           | B002      | Sistem Informas | 50000 | 2003         | T001          |     NULL | NULL           | Subianto        |
| 12332          | B003      | Basis Data  | 75000  | 2026         | T001          |     NULL | NULL           | juno            |
+----------------+-----------+-------------+--------+--------------+---------------+---------+--------------+-----------------+
2 rows in set (0.001 sec)
```

3. UNION, INTERSECT dan EXCEPT
   1. UNION

      UNION merupakan operator yang digunakan untuk menggabungkan basil query, dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing table yang akan ditampilkan datanya harus sama. Berikut ini perintah untuk memperoleh data pada tabel buku dimana tahun penerbitan 2003 clan 2004 :

```
PrakDB/Ferdy(060) > SELECT tahun_terbit, judul_buku
FROM buku WHERE tahun_terbit = '2003' UNION SELECT
tahun_terbit, judul_buku FROM buku WHERE
tahun_terbit = '2026';
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
PrakDb/Ferdy(060) >SELECT tahun_terbit, judul_buku FROM buku WHERE tahun_terbit = '2003' UNION SELECT tahun_terbit, judul_buku FROM buku WHERE tahun_terbit
= '2026';
+--------------+----------------+
| tahun_terbit | judul_buku     |
+--------------+----------------+
| 2003         | Sistem Informas |
| 2026         | Basis Data     |
+--------------+----------------+
2 rows in set (0.001 sec)
```

Perintah di atas identik dengan

```
PrakDB/Ferdy(060) > SELECT tahun_terbit, judul_buku FROM buku WHERE tahun_terbit='2003' OR tahun_terbit='2026';
```

```
PrakDb/Ferdy(060) >SELECT tahun_terbit, judul_buku FROM buku WHERE tahun_terbit='2003' OR tahun_terbit='2026';
+--------------+----------------+
| tahun_terbit | judul_buku     |
+--------------+----------------+
| 2003         | Sistem Informas |
| 2026         | Basis Data     |
| 2026         | Basis Data     |
+--------------+----------------+
3 rows in set (0.001 sec)
```

Namun tidak semua penggabungan dapat dilakukan dengan OR, yaitu jika bekerja pada dua tabel atau lebih.

2. INTERSECT

   INTERSECT merupakan operator yang digunakan untuk memperoleh data dari dua buah query dimana data yang ditampilkan adalah yang memenuhi kedua query tersebut dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama.

   Syntax:

```
SELECT * FROM namatabel1 INTERSECT SELECT * FROM namatabel2
```

   Pada MySQL tidak terdapat operator INTERSECT namun sebagai gantinya dapat menggunakan operator IN seperti contoh 1 pada bagian Nested Queries.

3. EXCEPT / Set Difference

   EXCEPT merupakan operator yang digunakan untuk memperoleh data dari dua buah query dimana data yang ditampilkan adalah data yang ada pada hasil query 1 dan tidak terdapat pada data dari hasil query 2 dengan ketentuan jumlah, nama dan tipe kolom dari masing-masing tabel yang akan ditampilkan datanya harus sama.

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Syntax:

```
SELECT * FROM namatabel1 EXCEPT SELECT * FROM namatabel2
```

Pada MySQL tidak terdapat operator EXCEPT namun sebagai gantinya dapat menggunakan operator NOT IN seperti contoh 2 pada bagian Nested Queries.

4. Nested Queries / Subquery (IN, NOT IN, EXISTS, NOT EXISTS)

   Subquery berarti query di dalam query. Dengan menggunakan subquery, hasil dari query akan menjadi bagian dari query di atasnya. Subquery terletak di dalam klausa WHERE atau HAVING. Pada klausa WHERE, subquery digunakan untuk memilih baris-baris tertentu yang kemudian digunakan oleh query. Sedangkan pada klausa HAVING, subquery digunakan untuk memilih kelompok baris yang kemudian digunakan oleh query.

   Contoh 1 : perintah untuk menampilkan data pada tabel pengarang yang mana data pada kolom kode_pengarang-nya tercantum pada tabel buku menggunakan IN :

```
PrakDB/Ferdy(060) > SELECT * FROM pengarang WHERE kode_pengarang IN (SELECT kode_pengarang FROM buku);
```

```
PrakDb/Ferdy(060) >SELECT * FROM pengarang WHERE kode_pengarang IN (SELECT kode_pengarang FROM buku);
+----------------+----------------+
| kode_pengarang | nama_pengarang |
+----------------+----------------+
| 12332          | juno           |
| P001           | Subianto       |
+----------------+----------------+
2 rows in set (0.001 sec)
```

Atau menggunakan EXISTS

```
PrakDB/Ferdy(060) > SELECT * FROM pengarang WHERE EXISTS (SELECT * FROM buku WHERE pengarang.kode_pengarang = buku.kode_pengarang);
```

```
PrakDb/Ferdy(060) >SELECT * FROM pengarang WHERE EXISTS (SELECT * FROM buku WHERE pengarang.kode_pengarang = buku.kode_pengarang);
+----------------+----------------+
| kode_pengarang | nama_pengarang |
+----------------+----------------+
| 12332          | juno           |
| P001           | Subianto       |
+----------------+----------------+
2 rows in set (0.001 sec)
```

Pada contoh di atas :

```
PrakDB/Ferdy(060) > SELECT kode pengarang FROM buku;
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
PrakDb/Ferdy(060) >SELECT kode_pengarang FROM buku;
+-----------------+
| kode_pengarang  |
+-----------------+
| P001            |
| 12332           |
| P001            |
+-----------------+
3 rows in set (0.000 sec)
```

Disebut subquery, sedangkan :

```
PrakDB/Ferdy(060) > SELECT * FROM pengarang;
```

```
PrakDb/Ferdy(060) >SELECT * FROM pengarang;
+-----------------+-----------------+
| kode_pengarang  | nama_pengarang  |
+-----------------+-----------------+
| 12332           | juno            |
| P001            | Subianto        |
+-----------------+-----------------+
2 rows in set (0.000 sec)
```

berkedudukan sebagai query. Perhatikan, terdapat data jenis dan harga pada tabel pengarang yang tidak ditampilkan. Hal ini disebabkan data pada kolom jenis tidak terdapat pada kolom jenis di tabel buku.

Contoh 2 : perintah untuk menampilkan data pada tabel pengarang yang mana data pada kolom jenis-nya tidak tercantum pada tabel buku menggunakan NOT IN:

```
PrakDB/Ferdy(060) > SELECT * FROM pengarang WHERE
kode_pengarang NOT IN (SELECT kode_pengarang FROM
buku);
```

```
PrakDb/Ferdy(060) >SELECT * FROM pengarang WHERE kode_pengarang NOT IN (SELECT kode_pengarang FROM buku);
+---------------+---------------+
| kode_pengarang | nama_pengarang |
+---------------+---------------+
| P002          | Jono          |
+---------------+---------------+
1 row in set (0.001 sec)
```

Atau menggunakan NOT EXISTS

```
PrakDB/Ferdy(060) > SELECT * FROM pengarang WHERE
NOT EXISTS (SELECT * FROM buku WHERE
pengarang.kode_pengarang = buku.kode_pengarang);
```

```
PrakDb/Ferdy(060) >SELECT * FROM pengarang WHERE NOT EXISTS (SELECT * FROM buku WHERE pengarang.kode_pengarang = buku.kode_pengarang);
+---------------+---------------+
| kode_pengarang | nama_pengarang |
+---------------+---------------+
| P002          | Jono          |
+---------------+---------------+
1 row in set (0.001 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

B. View

View adalah perintah query yang disimpan pada database dengan suatu nama tertentu, sehingga bisa digunakan setiap saat untuk melihat data tanpa menuliskan ulang query tersebut.

Syntax dasar perintah untuk membuat view adalah sebagai berikut :

```
CREATE
[OR REPLACE]
VIEW view_name [(column_list)]
AS select_statement
```

Kita menggunakan opsi OR REPLACE jika kita ingin mengganti view dengan nama yang sama dengan perintah tersebut. Jika tidak maka perintah CREATE VIEW akan menghasilkan error jika nama view yang ingin dibuat sudah ada sebelumnya.

C. Penggunaan view

1. View antar 2 tabel

   Kita akan membuat view dari relasi antara tabel "buku" dan "penerbit" untuk menampilkan data buku dan penerbitnya dari database perpustakaan dengan nama "view_buku". Perintahnya adalah sebagai berikut :

```
PrakDB/Ferdy(060) > CREATE VIEW view_buku AS SELECT
kode_buku, judul_buku, tahun_terbit, nama_pengarang
FROM buku JOIN pengarang;
```

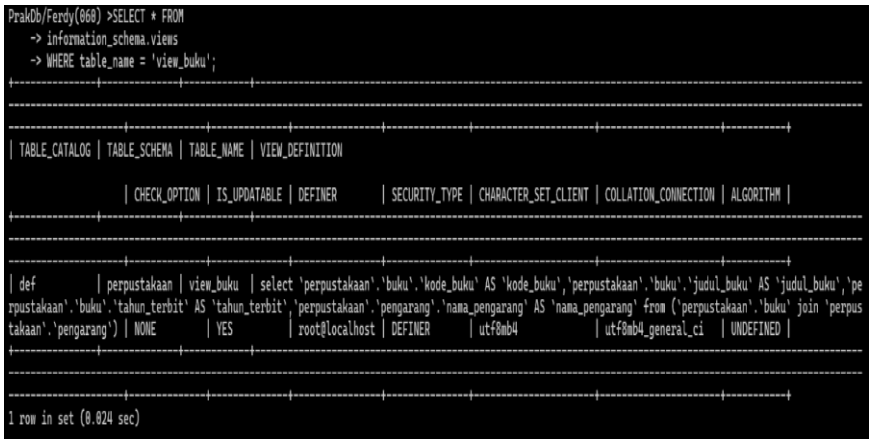
```
PrakDb/Ferdy(060) >CREATE VIEW view_buku AS SELECT kode_buku, judul_buku, tahun_terbit, nama_pengarang FROM buku JOIN pengarang;
Query OK, 0 rows affected (0.018 sec)

PrakDb/Ferdy(060) >select * from view_buku;
+-----------+---------------+-------------+---------------+
| kode_buku | judul_buku    | tahun_terbit | nama_pengarang |
+-----------+---------------+-------------+---------------+
| B002      | Sistem Informas | 2003        | juno          |
| B003      | Basis Data    | 2026        | juno          |
| B004      | Basis Data    | 2026        | juno          |
| B002      | Sistem Informas | 2003        | Subianto      |
| B003      | Basis Data    | 2026        | Subianto      |
| B004      | Basis Data    | 2026        | Subianto      |
| B002      | Sistem Informas | 2003        | Jono          |
| B003      | Basis Data    | 2026        | Jono          |
| B004      | Basis Data    | 2026        | Jono          |
+-----------+---------------+-------------+---------------+
9 rows in set (0.005 sec)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

Eksekusi perintah berikut untuk memastikan view telah dibuat :

```
PrakDB/Ferdy(060) > SELECT * FROM information_schema.views WHERE table_name = 'view_buku';
```



Lihat basil query view view buku :

```
PrakDB/Ferdy(060) > SELECT * FROM view_buku;
```

```
PrakDb/Ferdy(060) >SELECT * FROM view_buku;
+-----------+-----------------+--------------+-----------------+
| kode_buku | judul_buku      | tahun_terbit | nama_pengarang  |
+-----------+-----------------+--------------+-----------------+
| B002      | Sistem Informas | 2003         | juno            |
| B003      | Basis Data      | 2026         | juno            |
| B004      | Basis Data      | 2026         | juno            |
| B002      | Sistem Informas | 2003         | Subianto        |
| B003      | Basis Data      | 2026         | Subianto        |
| B004      | Basis Data      | 2026         | Subianto        |
| B002      | Sistem Informas | 2003         | Jono            |
| B003      | Basis Data      | 2026         | Jono            |
| B004      | Basis Data      | 2026         | Jono            |
+-----------+-----------------+--------------+-----------------+
9 rows in set (0.001 sec)
```

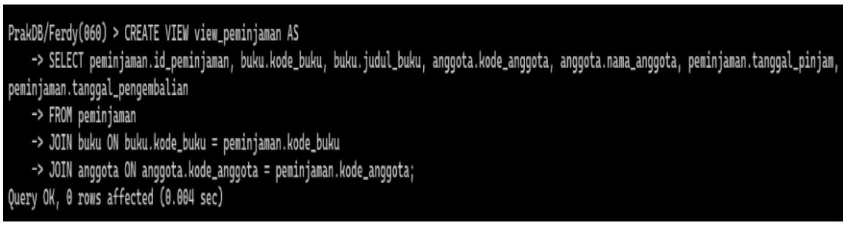
2. View dengan 3 tabel

   Membuat view dari relasi antara tabel **“buku”, “angota”** dan “peminjaman” untuk menampilkan data peminjaman buku dari database **perpustakaan** dengan nama **"view_peminjaman".** Perintahnya adalah sebagai berikut :

```
PrakDB/Ferdy(060) > CREATE VIEW view_peminjaman AS
SELECT peminjaman.id_peminjaman, buku.kode_buku,
buku.judul_buku, anggota.kode_anggota,
anggota.nama_anggota, peminjaman.tanggal_pinjam,
peminjaman.tanggal_pengembalian
FROM peminjaman
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
JOIN buku ON buku.kode_buku = peminjaman.kode_buku
JOIN    anggota    ON    anggota.kode_anggota    =
peminjaman.kode_anggota;
```



```
PrakDB/Ferdy(060) > CREATE VIEW view_peminjaman AS
    -> SELECT peminjaman.id_peminjaman, buku.kode_buku, buku.judul_buku, anggota.kode_anggota, anggota.nama_anggota, peminjaman.tanggal_pinjam, peminjaman.tanggal_pengembalian
    -> FROM peminjaman
    -> JOIN buku ON buku.kode_buku = peminjaman.kode_buku
    -> JOIN anggota ON anggota.kode_anggota = peminjaman.kode_anggota;
Query OK, 0 rows affected (0.004 sec)
```

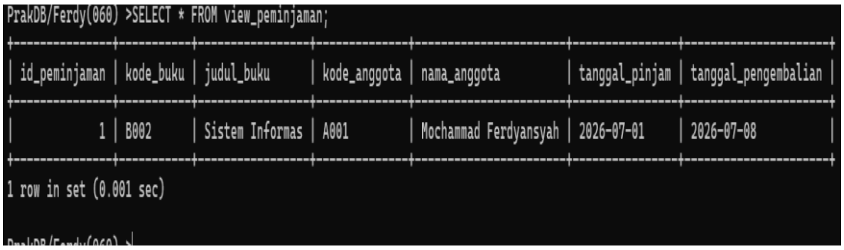
Eksekusi perintah berikut untuk memastikan view telah dibuat :

```
PrakDB/Ferdy(060)     >     SELECT     *     FROM
information_schema.views   WHERE   table_name   =
'view_peminjaman';
```

```
PrakDB/Ferdy(060) > SELECT * FROM information_schema.views WHERE table_name = 'view_peminjaman';
| TABLE_CATALOG | TABLE_SCHEMA | TABLE_NAME | VIEW_DEFINITION
| CHECK_OPTION | IS_UPDATABLE | DEFINER | SECURITY_TYPE | CHARACTER_SET_CLIENT | COLLATION_CONNECTION
| ALGORITHM |
| def | perpustakaan | view_peminjaman | select `perpustakaan`.`peminjaman`.`id_peminjaman` AS `id_peminjaman`,`perpustakaan`.`buku`.`kode_buku` AS `kode_buku`,`perpustakaan`.`buku`.`judul_buku` AS `judul_buku`,`perpustakaan`.`anggota`.`kode_anggota` AS `kode_anggota`,`perpustakaan`.`anggota`.`nama_anggota` AS `nama_anggota`,`perpustakaan`.`peminjaman`.`tanggal_pinjam` AS `tanggal_pinjam`,`perpustakaan`.`peminjaman`.`tanggal_pengembalian` AS `tanggal_pengembalian` from ((`perpustakaan`.`peminjaman` join `perpustakaan`.`buku` on(`perpustakaan`.`buku`.`kode_buku` = `perpustakaan`.`peminjaman`.`kode_buku`)) join `perpustakaan`.`anggota` on(`perpustakaan`.`anggota`.`kode_anggota` = `perpustakaan`.`peminjaman`.`kode_anggota`)) | NONE | YES | root@localhost | DEFINER | utf8mb4 | utf8mb4_general_ci
| UNDEFINED |
1 row in set (0.020 sec)
```

Lihat hasil query view view_peminjaman:

```
PrakDB/Ferdy(060) > SELECT * FROM view_peminjaman;
```



PrakDB/Ferdy(060) >SELECT * FROM view_peminjaman;

id_peminjaman	kode_buku	judul_buku	kode_anggota	nama_anggota	tanggal_pinjam	tanggal_pengembalian
1	B002	Sistem Informas	A001	Mochammad Ferdyansyah	2026-07-01	2026-07-08

1 row in set (0.001 sec)

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

LEMBAR KERJA DAN TUGAS

1. Buatlah contoh pengolahan data dari database hasil analisa pokok bahasan 4 menggunakan query : Inner Join, Equijoin, self-join dan Natural Join (antar mahasiswa atau kelompok tidak boleh sama), print screen basil eksekusi script.

   a. Inner join

```
PrakDB/Ferdy(060) > SELECT pegawai.nip, pegawai.nama,
pegawai.kelamin, jabatan.id_jabatan,
jabatan.jenis_jabatan, bagian.id_bagian,
bagian.nama_bagian, golongan.id_golongan,
golongan.golongan
FROM pegawai
INNER JOIN jabatan ON pegawai.id_jabatan =
jabatan.id_jabatan
INNER JOIN bagian ON pegawai.id_bagian = bagian.id_bagian
INNER JOIN golongan ON pegawai.id_golongan =
golongan.id_golongan;
```

```
PrakDB/Ferdy(060) >SELECT pegawai.nip, pegawai.nama, pegawai.kelamin, jabatan.id_jabatan, jabatan.jenis_jabatan, bagian.id_bagian, bagian.nama_bagian, golongan.id_golonga
n, golongan.golongan
    -> FROM pegawai
    -> INNER JOIN jabatan ON pegawai.id_jabatan = jabatan.id_jabatan
    -> INNER JOIN bagian ON pegawai.id_bagian = bagian.id_bagian
    -> INNER JOIN golongan ON pegawai.id_golongan = golongan.id_golongan;
+-----+-------+---------+------------+-----------------+-----------+--------------------+-------------+---------+
| nip | nama  | kelamin | id_jabatan | jenis_jabatan   | id_bagian | nama_bagian        | id_golongan | golongan |
+-----+-------+---------+------------+-----------------+-----------+--------------------+-------------+---------+
| PK1 | Fikri | L       | PK1        | IT manager      | K12       | teknologi informasi | 047         | class 4  |
| PK2 | Rio   | L       | P2H        | HRD             | B13       | recruitment        | 021         | class 2  |
| PK3 | Maya  | P       | PH9        | direktur keuangan | D29     | akuntansi          | 012         | class 1  |
| PK4 | Eko   | L       | PG1        | kepala logistik | G22       | persediaan         | 058         | class 5  |
| PK5 | Budi  | L       | P6C        | customer service | L22      | layanan_pelanggan  | 082         | class 8  |
+-----+-------+---------+------------+-----------------+-----------+--------------------+-------------+---------+
5 rows in set (0.001 sec)
```

   b. Equijoin

```
PrakDB/Ferdy(060) > SELECT a.nip, a.nama, a.kelamin,
b.id_bagian, b.nama_bagian
FROM pegawai a, bagian b
WHERE a.id_bagian = b.id_bagian;
```

```
PrakDB/Ferdy(060) >SELECT a.nip, a.nama, a.kelamin, b.id_bagian, b.nama_bagian
    -> FROM pegawai a, bagian b
    -> WHERE a.id_bagian = b.id_bagian;
+-----+-------+---------+-----------+--------------------+
| nip | nama  | kelamin | id_bagian | nama_bagian        |
+-----+-------+---------+-----------+--------------------+
| PK1 | Fikri | L       | K12       | teknologi informasi |
| PK2 | Rio   | L       | B13       | recruitment        |
| PK3 | Maya  | P       | D29       | akuntansi          |
| PK4 | Eko   | L       | G22       | persediaan         |
| PK5 | Budi  | L       | L22       | layanan_pelanggan  |
+-----+-------+---------+-----------+--------------------+
5 rows in set (0.001 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

c. Self join

```
ProkDB/Ferdy(060) > SELECT a.nip, a.nama, b.tgl_lahir
FROM pegawai a, pegawai b
WHERE a.nama='Ferdy' AND b.nama='Ferdy';
```

```
ProkDB/Ferdy(060) >SELECT a.nip, a.nama, b.tgl_lahir
    -> FROM pegawai a, pegawai b
    -> WHERE a.nama='Ferdy' AND b.nama='Ferdy';
+-----+-------+-------------+
| nip | nama  | tgl_lahir   |
+-----+-------+-------------+
| PK1 | Ferdy | 2005-05-04 |
+-----+-------+-------------+
1 row in set (0.001 sec)
```

d. Natural join

- Natural join

```
ProkDB/Ferdy(060) > SELECT a.nip, a.nama, a.kelamin,
b.id_golongan, b.golongan
FROM pegawai a NATURAL JOIN golongan b;
```

```
ProkDB/Ferdy(060) >SELECT a.nip, a.nama, a.kelamin, b.id_golongan, b.golongan
    -> FROM pegawai a NATURAL JOIN golongan b;
+-----+-------+---------+-------------+----------+
| nip | nama  | kelamin | id_golongan | golongan |
+-----+-------+---------+-------------+----------+
| PK1 | Ferdy | L       | 047         | class 4  |
| PK2 | Rio   | L       | 021         | class 2  |
| PK3 | Maya  | P       | 012         | class 1  |
| PK4 | Eko   | L       | 058         | class 5  |
| PK5 | Budi  | L       | 082         | class 8  |
+-----+-------+---------+-------------+----------+
5 rows in set (0.001 sec)
```

- Natural left join

```
ProkDB/Ferdy(060) > SELECT a.nip, a.nama, a.kelamin,
b.id_golongan, b.golongan
FROM pegawai a NATURAL JOIN golongan b;
```

```
ProkDB/Ferdy(060) >SELECT a.nip, a.nama, a.kelamin, b.id_golongan, b.golongan
    -> FROM pegawai a NATURAL JOIN golongan b;
+-----+-------+---------+-------------+----------+
| nip | nama  | kelamin | id_golongan | golongan |
+-----+-------+---------+-------------+----------+
| PK1 | Ferdy | L       | 047         | class 4  |
| PK2 | Rio   | L       | 021         | class 2  |
| PK3 | Maya  | P       | 012         | class 1  |
| PK4 | Eko   | L       | 058         | class 5  |
| PK5 | Budi  | L       | 082         | class 8  |
+-----+-------+---------+-------------+----------+
5 rows in set (0.001 sec)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

- Natural right join

```
PrakDB/Ferdy(060) > SELECT * FROM pegawai NATURAL RIGHT JOIN golongan;
```

```
PrakDB/Ferdy(060) >SELECT * FROM pegawai NATURAL RIGHT JOIN golongan;
+--------------+----------+------+-------+---------+------------+-----------+------------+
| id_golongan | golongan | nip  | nama  | kelamin | id_jabatan | id_bagian | tgl_lahir  |
+--------------+----------+------+-------+---------+------------+-----------+------------+
| 047          | class 4  | PK1  | Ferdy | L       | PK1        | K12       | 2005-05-04 |
| 021          | class 2  | PK2  | Rio   | L       | P2H        | B13       | NULL       |
| 012          | class 1  | PK3  | Maya  | P       | PH9        | D29       | NULL       |
| 058          | class 5  | PK4  | Eko   | L       | PG1        | G22       | NULL       |
| 082          | class 8  | PK5  | Budi  | L       | P6C        | L22       | NULL       |
+--------------+----------+------+-------+---------+------------+-----------+------------+
5 rows in set (0.001 sec)
```

2. Buatlah view yang menampilkan:

   a. Data lengkap pegawai (jabatan, bagian dan golongan yang ditampilkan bukan kode saja, tapi juga namanya)

```
PrakDB/Ferdy(060) > CREATE VIEW view_pegawai AS
SELECT p.nip, p.nama, p.kelamin, p.tgl_lahir, p.alamat,
p.telpon, p.pend_akhir, j.id_jabatan, j.jenis_jabatan,
g.id_golongan, g.golongan, b.id_bagian, b.nama_bagian
FROM pegawai p
INNER JOIN jabatan j ON p.id_jabatan = j.id_jabatan
INNER JOIN golongan g ON p.id_golongan = g.id_golongan
INNER JOIN bagian b ON p.id_bagian = b.id_bagian;
```

```
PrakDB/Ferdy(060) >CREATE VIEW view_pegawai AS
    -> SELECT p.nip, p.nama, p.kelamin, p.tgl_lahir, p.alamat, p.telpon, p.pend_akhir, j.id_jabatan, j.jenis_jabatan, g.id_golongan, g.golongan, b.id_bagian, b.nama_bagia
n
    -> FROM pegawai p
    -> INNER JOIN jabatan j ON p.id_jabatan = j.id_jabatan
    -> INNER JOIN golongan g ON p.id_golongan = g.id_golongan
    -> INNER JOIN bagian b ON p.id_bagian = b.id_bagian;
Query OK, 0 rows affected (0.003 sec)

PrakDB/Ferdy(060) >show tables;
+-----------------------+
| Tables_in_perpustakaan |
+-----------------------+
| anggota               |
| bagian                |
| buku                  |
| golongan              |
| jabatan               |
| pegawai               |
| peminjaman            |
| pengarang             |
| view_buku             |
| view_pegawai          |
| view_peminjaman       |
+-----------------------+
11 rows in set (0.001 sec)
```

```
PrakDB/Ferdy(060) > Select * form view_pegawai;
```

```
PrakDB/Ferdy(060) >SELECT * FROM view_pegawai;
+-----+-------+---------+------------+--------+--------+-----------+------------+-----------------+------------+----------+-----------+-------------------+
| nip | nama  | kelamin | tgl_lahir  | alamat | telpon | pend_akhir | id_jabatan | jenis_jabatan   | id_golongan | golongan | id_bagian | nama_bagian       |
+-----+-------+---------+------------+--------+--------+-----------+------------+-----------------+------------+----------+-----------+-------------------+
| PK1 | Ferdy | L       | 2005-05-04 | NULL   | NULL   | NULL      | PK1        | IT manager      | 047        | class 4  | K12       | teknologi informasi |
| PK2 | Rio   | L       | NULL       | NULL   | NULL   | NULL      | P2H        | HRD             | 021        | class 2  | B13       | recruitment       |
| PK3 | Maya  | P       | NULL       | NULL   | NULL   | NULL      | PH9        | direktur keuangan | 012        | class 1  | D29       | akuntansi         |
| PK4 | Eko   | L       | NULL       | NULL   | NULL   | NULL      | PG1        | kepala logistik | 058        | class 5  | G22       | persediaan        |
| PK5 | Budi  | L       | NULL       | NULL   | NULL   | NULL      | P6C        | customer service | 082        | class 8  | L22       | layanan_pelanggan |
+-----+-------+---------+------------+--------+--------+-----------+------------+-----------------+------------+----------+-----------+-------------------+
5 rows in set (0.002 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

b. Absensi Pegawai

```
PrakDB/Ferdy(060) > SELECT a.nip, a.nama, a.kelamin,
a.id_jabatan, e.tgl_absensi, e.jam_masuk, e.jam_keluar,
e.keterangan
FROM pegawai a
INNER JOIN jabatan b ON a.id_jabatan = b.id_jabatan
INNER JOIN absensi e ON a.nip = e.nip;
```

```
PrakDB/Ferdy(060) >SELECT a.nip, a.nama, a.kelamin, a.id_jabatan, e.tgl_absensi, e.jam_masuk, e.jam_keluar, e.keterangan
    -> FROM pegawai a
    -> INNER JOIN jabatan b ON a.id_jabatan = b.id_jabatan
    -> INNER JOIN absensi e ON a.nip = e.nip;
+-----+-------+---------+-------------+--------------+-----------+------------+-------------+
| nip | nama  | kelamin | id_jabatan  | tgl_absensi  | jam_masuk | jam_keluar | keterangan  |
+-----+-------+---------+-------------+--------------+-----------+------------+-------------+
| PK1 | Ferdy | L       | PK1         | 2023-07-06   | 07:30:00  | 17:00:00   | masuk       |
| PK2 | Rio   | L       | P2H         | 2023-07-06   | 09:00:00  | 18:00:00   | telat       |
| PK3 | Maya  | P       | PH9         | 2023-07-06   | 07:00:00  | 17:00:00   | masuk       |
| PK4 | Eko   | L       | PG1         | 2023-07-06   | 08:00:00  | 17:00:00   | masuk       |
| PK5 | Budi  | L       | P6C         | 2023-07-06   | 07:00:00  | 17:00:00   | masuk       |
+-----+-------+---------+-------------+--------------+-----------+------------+-------------+
5 rows in set (0.001 sec)
```

```
PrakDB/Ferdy(060) > Select * from view pegawai;
```

```
PrakDB/Ferdy(060) >SELECT * FROM view_pegawai;
+-----+-------+---------+------------+--------+-------+-----------+-------------+-------------------+-------------+---------+-----------+-----------------+
| nip | nama  | kelamin | tgl_lahir  | alamat | telpon | pend_akhir | id_jabatan | jenis_jabatan     | id_golongan | golongan | id_bagian | nama_bagian     |
+-----+-------+---------+------------+--------+-------+-----------+-------------+-------------------+-------------+---------+-----------+-----------------+
| PK1 | Ferdy | L       | 2005-05-04 | NULL   | NULL  | NULL      | PK1         | IT manager        | 047         | class 4 | K12       | teknologi informasi |
| PK2 | Rio   | L       | NULL       | NULL   | NULL  | NULL      | P2H         | HRD               | 021         | class 2 | B13       | recruitment     |
| PK3 | Maya  | P       | NULL       | NULL   | NULL  | NULL      | PH9         | direktur keuangan | 012         | class 1 | D29       | akuntansi       |
| PK4 | Eko   | L       | NULL       | NULL   | NULL  | NULL      | PG1         | kepala logistik   | 058         | class 5 | G22       | persediaan      |
| PK5 | Budi  | L       | NULL       | NULL   | NULL  | NULL      | P6C         | customer service  | 082         | class 8 | L22       | layanan_pelanggan |
+-----+-------+---------+------------+--------+-------+-----------+-------------+-------------------+-------------+---------+-----------+-----------------+
5 rows in set (0.001 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 – 2026

Lembar Asistensi

Praktikum Basis Data

Pokok Bahasan 6

Judul	: Data Control Language (DCL)
Nama	: Mochammad Ferdyansyah
NIM	: 251080200060
Kelompok	: 6
Dilaksanakan	: 02 Juli 2026

Mengetahui,

Dosen Praktikum	Asisten Praktikum
9/07/26	6/7/2026
(Ika Ratna Indra Astutik, S.Kom., MT.)	**(Zianisa Azzahra)**

POKOK BAHASAN 6

Data Control Language (DCL) / Hak Akses User

A. PENDAHULUAN

Pokok bahasan ini akan dibahas mengenai manajemen hak akses user terhadap basis data. Manajemen hak akses pengguna terhadap basis data sangat penting untuk menjaga keamanan dan integritas data. Dalam basis data, hak akses merujuk pada izin atau peran yang diberikan kepada pengguna untuk mengakses atau memodifikasi data dan objek-objek dalam basis data (misalnya, tabel, view, prosedur, dll.).

B. TUJUAN

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Mengetahui dan memahami hak akses di basis data.
2. Mengetahui dan memahami pengaturan hak akses user.
3. Memahami dan menerapkan batasan-batasan hak akses user

C. PENYAJIAN (TUTORIAL)

A. Pemahaman Hak Akses

Basis data yang telah dibuat perlu diatur agar data selalu dalam keadaan aman dari pemakai yang tidak berhak. Pengaturan hak akses berguna dalam hal pembatasan pengaksesan suatu data, misalkan hanya pemakai tertentu yang bisa membaca atau pemakai lain yang justru dapat melakukan perubahan dan penghapusan data.

Macam-macam perintah yang terkait dengan hak akses adalah SELECT, INSERT, UPDATE, DELETE, REFERENCES, INDEX, CREATE, ALTER dan DROP.

B. Mengatur Hak Akses

Untul MySQL versi 3.22. keatas dalam manajemen user dapat menggunakan perintah GRANT dan REVOKE untuk mengatur hak akses pemakai (user).

1. Perintah GRANT

   Dipergunakan untuk membuat user baru dengan izin aksesnya.

   Bentuk umum :

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
GRANT jenis_akses (nama_kolom) ON nama_database TO
nama_user IDENTIFIED BY ”nama_password” [WITH GRANT
pilihan_akses]
```

Atau

```
GRANT hak akses ON namatabel TO pemakai;
```

Dimana :

- Hak_akses merupakan hak yang diberikan kepada pemakai berupa SELECT, INSERT saja atau keduanya. Bila hak akses lebih dari satu antar hak akses dipisahkan dengann koma (,).
- Nama tabel, menyatakan nama tabel yang akan diakses dan diatur.
- Pemakai, nama pemakai yang telah didaftarkan pada sistem database. Sejumlah pemakai bisa disebutkan dengan dipisahkan tanda koma (,).

Contoh:

Misalkan kita sebagai Administrator basis data yang mempunyai wewenang untuk mengatur hak akses para pemakai. Kita akan mengatur hak akses pengguna siska dan edi (sebagai user).

```
PrakDB/Ferdy(060) > GRANT SELECT ON buku TO siska;
```

```
PrakDB/Ferdy(060) >SHOW GRANTS FOR siska;
+-----------------------------------------------------+
| Grants for siska@%                                  |
+-----------------------------------------------------+
| GRANT USAGE ON *.* TO `siska`@`%`                   |
| GRANT SELECT ON `perpustakaan`.`buku` TO `siska`@`%` |
+-----------------------------------------------------+
2 rows in set (0.000 sec)
```

Perintah diatas digunakan untuk memberikan hak akses SELECT terhadap tabel buku kepada user siska sehingga user siska dapat menggunakan perintah SELECT untuk melakukan proses query pada tabel buku.

Hak akses lebih dari satu:

```
PrakDB/Ferdy(060) > GRANT SELECT, INSERT, UPDATE, DELETE
ON buku TO siska, edi;
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
PrakDB/Ferdy(060) >GRANT SELECT, INSERT, UPDATE,
    -> DELETE ON buku TO siska, edi;
Query OK, 0 rows affected (0.002 sec)

PrakDB/Ferdy(060) >show grants for siska;
+-----------------------------------------------------------------------------------+
| Grants for siska@%                                                                |
+-----------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `siska`@`%`                                                 |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `perpustakaan`.`buku` TO `siska`@`%`      |
+-----------------------------------------------------------------------------------+
2 rows in set (0.000 sec)

PrakDB/Ferdy(060) >show grants for edi;
+-----------------------------------------------------------------------------------+
| Grants for edi@%                                                                  |
+-----------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `edi`@`%`                                                   |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `perpustakaan`.`buku` TO `edi`@`%`        |
+-----------------------------------------------------------------------------------+
2 rows in set (0.000 sec)
```

C. Membatasi Hak Akses

Hak akses perlu dibatasi untuk memudahkan dalam mengatur dan mengawasi pemakaian data serta menjaga keamanan data.

Contoh:

Administrator akan memberikan hak akses kepada edi dalam melakukan query tabel buku untuk field tertentu saja. Perintahnya:

```
PrakDB/Ferdy(060) > GRANT SELECT, UPDATE (kode_buku, judul_buku, tahun_terbit) ON buku TO edi;
```

```
PrakDB/Ferdy(060) >GRANT SELECT, UPDATE (kode_buku,
    -> judul_buku, tahun_terbit) ON buku TO edi;
Query OK, 0 rows affected (0.005 sec)

PrakDB/Ferdy(060) >show grants for edi;
+-----------------------------------------------------------------------------------------------------------------------+
| Grants for edi@%                                                                                                      |
+-----------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `edi`@`%`                                                                                       |
| GRANT SELECT, INSERT, UPDATE, UPDATE (tahun_terbit, kode_buku, judul_buku), DELETE ON `perpustakaan`.`buku` TO `edi`@`%` |
+-----------------------------------------------------------------------------------------------------------------------+
2 rows in set (0.000 sec)
```

Dari perintah diatas user niko hanya dapat melakukan SELECT dan UPDATE terhadap tiga field yaitu kode_buku, judul_buku, tahun_terbit.

D. Hak Akses Penuh

Untuk memberikan hak akses penuh kepada pemakai, dapat memakai perintah klausa ALL PRIVILEGES. Tentunya dengan pemberian hak akses penuh kepada pemakai (user).

Contoh:

```
PrakDB/Ferdy(060) > GRANT ALL PRIVILEGES ON perpustakaan to siska;
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
PrakDB/Ferdy(060) >GRANT ALL PRIVILEGES ON perpustakaan
    -> to siska;
Query OK, 0 rows affected (0.002 sec)

PrakDB/Ferdy(060) >show grants for siska;
+-----------------------------------------------------------------------------+
| Grants for siska@%                                                          |
+-----------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `siska`@`%`                                           |
| GRANT ALL PRIVILEGES ON `perpustakaan`.`perpustakaan` TO `siska`@`%`        |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `perpustakaan`.`buku` TO `siska`@`%` |
+-----------------------------------------------------------------------------+
3 rows in set (0.000 sec)
```

Atau menggunakan

```
PrakDB/Ferdy(060) > GRANT ALL PRIVILEGES ON kepegawaian.pegawai TO 'root'@'localhost';
```

```
PrakDB/Ferdy(060) >GRANT ALL PRIVILEGES ON
    -> kepegawaian.pegawai TO 'root'@'localhost';
Query OK, 0 rows affected (0.002 sec)

PrakDB/Ferdy(060) >show grants for 'root'@'localhost';
+-----------------------------------------------------------------------+
| Grants for root@localhost                                             |
+-----------------------------------------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO `root`@`localhost` WITH GRANT OPTION   |
| GRANT ALL PRIVILEGES ON `kepegawaian`.`pegawai` TO `root`@`localhost` |
+-----------------------------------------------------------------------+
2 rows in set (0.000 sec)
```

E. Hak Akses kepada Public

Untuk memberikan hak akses kepada banyak user dapat menggunakan klausa PUBLIC. Bebrapa DBMS ada yang menggunakan klausa WORLD.

Contoh:

```
PrakDB/Ferdy(060) > GRANT SELECT, INSERT ON buku TO ''@'%';
```

```
PrakDB/Ferdy(060) >show grants for ''@'%';
+-----------------------------------------------------------+
| Grants for @%                                             |
+-----------------------------------------------------------+
| GRANT USAGE ON *.* TO ``@`%`                              |
| GRANT SELECT, INSERT ON `perpustakaan`.`buku` TO ``@`%`   |
+-----------------------------------------------------------+
2 rows in set (0.000 sec)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

F. Pencabutan Hak Akses

1. Pencabutan Hak Akses Sementara

   Untuk melakukan pencabutan atau penghapusan hak akses user menggunakan perintah REVOKE. Perintah ini juga mampu melakukan pencabutan hak akses sebagian pemakai atau secara keseluruhan. Bentuk umum :

```
REVOKE hak akses ON nama database FROM nama user;
```

   Atau

```
REVOKE hak akses ON namatabel FROM nama user;
```

   Contoh:

   Admistrator ingin mencabut hak akses user siska, maka perintahnya:

```
PrakDB/Ferdy(060) > REVOKE SELECT ON buku FROM siska;
```

```
PrakDB/Ferdy(060) >REVOKE SELECT ON buku FROM siska;
Query OK, 0 rows affected (0.005 sec)

PrakDB/Ferdy(060) >show grants for siska;
+-----------------------------------------------------------------------+
| Grants for siska@%                                                    |
+-----------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `siska`@`%`                                     |
| GRANT ALL PRIVILEGES ON `perpustakaan`.`perpustakaan` TO `siska`@`%` |
| GRANT INSERT, UPDATE, DELETE ON `perpustakaan`.`buku` TO `siska`@`%` |
+-----------------------------------------------------------------------+
3 rows in set (0.000 sec)
```

   Atau

```
PrakDB/Ferdy(060) > REVOKE SELECT, INSERT ON buku FROM
edi;
```

```
PrakDB/Ferdy(060) >REVOKE SELECT, INSERT ON  buku
    -> FROM edi;
Query OK, 0 rows affected (0.002 sec)

PrakDB/Ferdy(060) >show grants for edi;
+-----------------------------------------------------------------------------------------------------------+
| Grants for edi@%                                                                                          |
+-----------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `edi`@`%`                                                                           |
| GRANT UPDATE, UPDATE (tahun_terbit, kode_buku, judul_buku), DELETE ON `perpustakaan`.`buku` TO `edi`@`%` |
+-----------------------------------------------------------------------------------------------------------+
2 rows in set (0.000 sec)
```

2. Untuk menghapus user secara pemanen dari basis data.

```
PrakDB/Ferdy(060) > DROP USER 'siska'@'%';
```

```
PrakDB/Ferdy(060) >DROP USER 'siska'@'%';
Query OK, 0 rows affected (0.002 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

LEMBAR KERJA DAN TUGAS

1. Buatlah hak akses user dengan privilege
   a. User dengan hak akses SELECT pada salah satu tabel di database Anda

```
PrakDB/Ferdy(060) > CREATE USER 'Ferdy'@'localhost' IDENTIFIED BY 'password123';
```

```
PrakDB/Ferdy(060) >CREATE USER 'Ferdy'@'localhost' IDENTIFIED BY 'password123';
Query OK, 0 rows affected (0.003 sec)

PrakDB/Ferdy(060) >show create user 'Ferdy'@'localhost';
+-----------------------------------------------------------------------------------------------------------+
| CREATE USER for Ferdy@localhost                                                                           |
+-----------------------------------------------------------------------------------------------------------+
| CREATE USER `Ferdy`@`localhost` IDENTIFIED BY PASSWORD '*A0F874BC7F54EE086FCE60A37CE7887D8B31086B' |
+-----------------------------------------------------------------------------------------------------------+
1 row in set (0.001 sec)
```

```
PrakDB/Ferdy(060) > GRANT SELECT, INSERT, UPDATE ON kepegawaian.* TO 'Ferdy'@'localhost';
```

```
PrakDB/Ferdy(060) >GRANT SELECT, INSERT, UPDATE ON kepegawaian.* TO 'Ferdy'@'localhost';
Query OK, 0 rows affected (0.004 sec)

PrakDB/Ferdy(060) >SHOW GRANTS FOR 'Ferdy'@'localhost';
+-----------------------------------------------------------------------+
| Grants for Ferdy@localhost                                            |
+-----------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `Ferdy`@`localhost`                             |
| GRANT SELECT, INSERT, UPDATE ON `perpustakaan`.* TO `Ferdy`@`localhost` |
| GRANT SELECT, INSERT, UPDATE ON `kepegawaian`.* TO `Ferdy`@`localhost` |
+-----------------------------------------------------------------------+
3 rows in set (0.000 sec)
```

```
#mysql -u Ferdy -p
Enter password: ***
MariaDB [(none)]> show databases;
```

```
# mysql -u Ferdy -p
Enter password: ***
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 10.4.27-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| kepegawaian        |
| test               |
+--------------------+
3 rows in set (0.029 sec)
```


fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
MariaDB[(none)]> USE kepegawaian;
```

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

```
MariaDB[(kepegawaian)]> SHOW TABLES;
```

```
MariaDB [(none)]> use kepegawaian;
Database changed
MariaDB [kepegawaian]> show tables;
+------------------------+
| Tables_in_kepegawaian  |
+------------------------+
| absensi                |
| bagian                 |
| golongan               |
| jabatan                |
| pegawai                |
+------------------------+
5 rows in set (0.001 sec)
```

```
MariaDB[(kepegawaian)]> SELECT * FROM pegawai;
```

```
MariaDB [kepegawaian]> SELECT * FROM pegawai;
+-----+---------------+---------+-----------+------------+----------+-----------+--------------+-----------+------+-------+-----------+
| nip | nama          | kelamin | tmp_lahir | tgl_lahir  | alamat   | kabupaten | telpon       | pend_akhir | kd_jabatan | gol | jenis | kd_bagian |
+-----+---------------+---------+-----------+------------+----------+-----------+--------------+-----------+------+-------+-----------+
| P001 | Ferdyansyah  | L       | Sidoarjo  | 2005-01-10 | Gempol   | Sidoarjo  | 085162914550 | S1        | J001       | IVD | T     | NULL      |
| P002 | Ayu Lestari  | P       | Pasuruan  | 2002-03-15 | Bangil   | Pasuruan  | 081234567890 | S1        | J002       | IIIC | K    | NULL      |
| P003 | Bambang S    | L       | Mojokerto | 1998-11-20 | Pacet    | Mojokerto | 081398765432 | S2        | J003       | IVA | T     | NULL      |
| P004 | Siti Aminah  | P       | Surabaya  | 2000-07-08 | Waru     | Sidoarjo  | 085711223344 | SMA       | J004       | IIB | K     | NULL      |
| P005 | Dedi Kurniawan | L     | Gresik    | 1995-12-30 | Driyorejo | Gresik   | 089677889900 | S1        | J005       | IIID | T    | NULL      |
+-----+---------------+---------+-----------+------------+----------+-----------+--------------+-----------+------+-------+-----------+
5 rows in set (0.033 sec)
```

b. User dengan hak akses SELECT, INSERT dan UPDATE pada salah satu tabel di database Anda.

```
PrakDB/Ferdy(060) > GRANT SELECT, INSERT, UPDATE ON
kepegawaian.pegawai TO 'Ferdy'@'localhost';
```

```
PrakDB/Ferdy(060) >GRANT SELECT, INSERT, UPDATE ON kepegawaian.pegawai TO 'Ferdy'@'localhost';
Query OK, 0 rows affected (0.005 sec)

PrakDB/Ferdy(060) >show grants for 'Ferdy'@'localhost';
+-----------------------------------------------------------------------------------------------------------------------------+
| Grants for Ferdy@localhost                                                                                                  |
+-----------------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `Ferdy`@`localhost` IDENTIFIED BY PASSWORD '*23AE809DDACAF96AF0FD78ED04B6A265E05AA257'                |
| GRANT SELECT, INSERT, UPDATE ON `kepegawaian`.* TO `Ferdy`@`localhost`                                                      |
| GRANT SELECT, INSERT, UPDATE ON `kepegawaian`.`pegawai` TO `Ferdy`@`localhost`                                              |
+-----------------------------------------------------------------------------------------------------------------------------+
3 rows in set (0.000 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
#mysql -u Ferdy -p;
MariaDB[(none)]>use kepegawaian;
```

```
# mysql -u Ferdy -p
Enter password: ***
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.4.27-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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

```
MariaDB [kepegawaian]> show tables;
```

```
MariaDB [kepegawaian]> show tables;
+-------------------------+
| Tables_in_kepegawaian   |
+-------------------------+
| absensi                 |
| bagian                  |
| golongan                |
| jabatan                 |
| pegawai                 |
+-------------------------+
5 rows in set (0.001 sec)
```

```
MariaDB [kepegawaian]> SELECT * FROM pegawai;
```

```
MariaDB [kepegawaian]> SELECT * FROM pegawai;
+------+---------------+---------+-----------+------------+-----------+-----------+--------------+-----------+-----------+------+-------+-----------+
| nip  | nama          | kelamin | tmp_lahir | tgl_lahir  | alamat    | kabupaten | telpon       | pend_akhir | kd_jabatan | gol  | jenis | kd_bagian |
+------+---------------+---------+-----------+------------+-----------+-----------+--------------+-----------+-----------+------+-------+-----------+
| P001 | Ferdyansyah   | L       | Sidoarjo  | 2005-01-10 | Gempol    | Sidoarjo  | 085162914550 | S1        | J001      | IVD  | T     | NULL      |
| P002 | Ayu Lestari   | P       | Pasuruan  | 2002-03-15 | Bangil    | Pasuruan  | 081234567890 | S1        | J002      | IIIC | K     | NULL      |
| P003 | Bambang S     | L       | Mojokerto | 1998-11-20 | Pacet     | Mojokerto | 081398765432 | S2        | J003      | IVA  | T     | NULL      |
| P004 | Siti Aminah   | P       | Surabaya  | 2000-07-08 | Waru      | Sidoarjo  | 085711223344 | SMA       | J004      | IIB  | K     | NULL      |
| P005 | Dedi Kurniawan | L      | Gresik    | 1995-12-30 | Driyorejo | Gresik    | 089677889900 | S1        | J005      | IIID | T     | NULL      |
+------+---------------+---------+-----------+------------+-----------+-----------+--------------+-----------+-----------+------+-------+-----------+
5 rows in set (0.000 sec)
```

```
MariaDB [kepegawaian]> INSERT INTO pegawai VALUES (
    'PK6',
    'Nia',
    'P',
    'sidoarjo',
    '1993-09-12',
    'prambon',
    'sidoarjo',
    '0818874',
    'S1',
    'PH7',
    '030',
    'K',
    'B13'
);
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

```
MariaDB [kepegawaian]> INSERT INTO pegawai VALUES (
    ->     'PK6',
    ->     'Nia',
    ->     'P',
    ->     'sidoarjo',
    ->     '1993-09-12',
    ->     'prambon',
    ->     'sidoarjo',
    ->     '0818874',
    ->     'S1',
    ->     'PH7',
    ->     '030',
    ->     'K',
    ->     'B13'
    -> );
Query OK, 1 row affected (0.036 sec)

MariaDB [kepegawaian]> select * from pegawai;
+------+---------------+---------+-----------+------------+-----------+-----------+--------------+-----------+------+-------+-----------+
| nip  | nama          | kelamin | tmp_lahir | tgl_lahir  | alamat    | kabupaten | telpon       | pend_akhir | kd_jabatan | gol  | jenis | kd_bagian |
+------+---------------+---------+-----------+------------+-----------+-----------+--------------+-----------+------+-------+-----------+
| P001 | Ferdyansyah   | L       | Sidoarjo  | 2005-01-10 | Gempol    | Sidoarjo  | 085162914550 | S1        | J001       | IVD  | T     | NULL      |
| P002 | Ayu Lestari   | P       | Pasuruan  | 2002-03-15 | Bangil    | Pasuruan  | 081234567890 | S1        | J002       | IIIC | K     | NULL      |
| P003 | Bambang S     | L       | Mojokerto | 1998-11-20 | Pacet     | Mojokerto | 081398765432 | S2        | J003       | IVA  | T     | NULL      |
| P004 | Siti Aminah   | P       | Surabaya  | 2000-07-08 | Waru      | Sidoarjo  | 085711223344 | SMA       | J004       | IIB  | K     | NULL      |
| P005 | Dedi Kurniawan| L       | Gresik    | 1995-12-30 | Driyorejo | Gresik    | 089677889900 | S1        | J005       | IIID | T     | NULL      |
| PK6  | Nia           | P       | sidoarjo  | 1993-09-12 | prambon   | sidoarjo  | 0818874      | S1        | PH7        | 030  | K     | B13       |
+------+---------------+---------+-----------+------------+-----------+-----------+--------------+-----------+------+-------+-----------+
6 rows in set (0.000 sec)
```

```
MariaDB [kepegawaian]> UPDATE pegawai SET nama = 'aya'
WHERE nip = 'PK6';
```

```
MariaDB [kepegawaian]> UPDATE pegawai SET nama = 'aya' WHERE nip = 'PK6';
Query OK, 1 row affected (0.020 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [kepegawaian]> select * from pegawai;
+------+---------------+---------+-----------+------------+-----------+-----------+--------------+-----------+------+-------+-----------+
| nip  | nama          | kelamin | tmp_lahir | tgl_lahir  | alamat    | kabupaten | telpon       | pend_akhir | kd_jabatan | gol  | jenis | kd_bagian |
+------+---------------+---------+-----------+------------+-----------+-----------+--------------+-----------+------+-------+-----------+
| P001 | Ferdyansyah   | L       | Sidoarjo  | 2005-01-10 | Gempol    | Sidoarjo  | 085162914550 | S1        | J001       | IVD  | T     | NULL      |
| P002 | Ayu Lestari   | P       | Pasuruan  | 2002-03-15 | Bangil    | Pasuruan  | 081234567890 | S1        | J002       | IIIC | K     | NULL      |
| P003 | Bambang S     | L       | Mojokerto | 1998-11-20 | Pacet     | Mojokerto | 081398765432 | S2        | J003       | IVA  | T     | NULL      |
| P004 | Siti Aminah   | P       | Surabaya  | 2000-07-08 | Waru      | Sidoarjo  | 085711223344 | SMA       | J004       | IIB  | K     | NULL      |
| P005 | Dedi Kurniawan| L       | Gresik    | 1995-12-30 | Driyorejo | Gresik    | 089677889900 | S1        | J005       | IIID | T     | NULL      |
| PK6  | aya           | P       | sidoarjo  | 1993-09-12 | prambon   | sidoarjo  | 0818874      | S1        | PH7        | 030  | K     | B13       |
+------+---------------+---------+-----------+------------+-----------+-----------+--------------+-----------+------+-------+-----------+
6 rows in set (0.000 sec)
```

c. User dengan hak akses penuh terhadap basis data Anda.

```
PrakDB/Ferdy(060)    >    GRANT    ALL    PRIVILEGES    ON
kepegawaian.pegawai TO 'Ferdy'@'localhost';
```

```
PrakDB/Ferdy(060) >GRANT ALL PRIVILEGES ON kepegawaian.pegawai TO 'Ferdy'@'localhost';
Query OK, 0 rows affected (0.002 sec)

PrakDB/Ferdy(060) >show grants for 'Ferdy'@'localhost';
+-----------------------------------------------------------------------------------------------------------------------+
| Grants for Ferdy@localhost                                                                                            |
+-----------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `Ferdy`@`localhost` IDENTIFIED BY PASSWORD '*23AE809DDACAF96AF0FD78ED04B6A265E05AA257'          |
| GRANT SELECT, INSERT, UPDATE ON `kepegawaian`.* TO `Ferdy`@`localhost`                                                |
| GRANT ALL PRIVILEGES ON `kepegawaian`.`pegawai` TO `Ferdy`@`localhost`                                                |
+-----------------------------------------------------------------------------------------------------------------------+
3 rows in set (0.000 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

d. Hak akses untuk public pada salah satu tabel dalam basis data Anda.

```
PrakDB/Ferdy(060) > GRANT SELECT ON kepegawaian.absensi
TO ''@'%';
```

```
PrakDB/Ferdy(060) >GRANT SELECT ON kepegawaian.absensi TO ''@'%';
Query OK, 0 rows affected (0.005 sec)

PrakDB/Ferdy(060) >show grants for ''@'%';
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| Grants for @%                                                                                                                                                                   |
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO ``@`%`                                                                                                                                                    |
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, EVENT, TRIGGER, DELETE HISTORY ON `test`.* TO ``@`%`     |
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, EVENT, TRIGGER, DELETE HISTORY ON `test\_%`.* TO ``@`%` |
| GRANT SELECT ON `kepegawaian`.`absensi` TO ``@`%`                                                                                                                               |
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
4 rows in set (0.000 sec)
```

2. Mencabut hak akses User:

a. Cabutlah hak akses salah satu user yang sudah Anda buat.

```
PrakDB/Ferdy(060) > PrakDB/Ferdy(060) > REVOKE INSERT,
UPDATE ON kepegawaian.pegawai FROM 'Ferdy'@'localhost';
```

```
PrakDB/Ferdy(060) >REVOKE INSERT, UPDATE ON kepegawaian.pegawai FROM 'Ferdy'@'localhost';
Query OK, 0 rows affected (0.002 sec)

PrakDB/Ferdy(060) >show grants for 'Ferdy'@'localhost';
+-------------------------------------------------------------------------------------------------------------------------------------------------+
| Grants for Ferdy@localhost                                                                                                                      |
+-------------------------------------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `Ferdy`@`localhost` IDENTIFIED BY PASSWORD '*23AE809DDACAF96AF0FD78ED04B6A265E05AA257'                                    |
| GRANT SELECT, INSERT, UPDATE ON `kepegawaian`.* TO `Ferdy`@`localhost`                                                                          |
| GRANT SELECT, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE VIEW, SHOW VIEW, TRIGGER, DELETE HISTORY ON `kepegawaian`.`pegawai` TO `Ferdy`@`localhost` |
+-------------------------------------------------------------------------------------------------------------------------------------------------+
3 rows in set (0.000 sec)
```

b. Cabutlah secara permanen user yang mempunyai hak akses penuh.

```
PrakDB/Ferdy(060) > REVOKE ALL PRIVILEGES ON
kepegawaian.pegawai FROM 'Ferdy'@'localhost';
```

```
PrakDB/Ferdy(060) >REVOKE ALL PRIVILEGES ON kepegawaian.pegawai FROM 'Ferdy'@'localhost';
Query OK, 0 rows affected (0.003 sec)

PrakDB/Ferdy(060) >show grants for 'Ferdy'@'localhost';
+----------------------------------------------------------------------------------------------------------------+
| Grants for Ferdy@localhost                                                                                     |
+----------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `Ferdy`@`localhost` IDENTIFIED BY PASSWORD '*23AE809DDACAF96AF0FD78ED04B6A265E05AA257' |
| GRANT SELECT, INSERT, UPDATE ON `kepegawaian`.* TO `Ferdy`@`localhost`                                         |
+----------------------------------------------------------------------------------------------------------------+
2 rows in set (0.000 sec)
```

```
PrakDB/Ferdy(060) > DROP USER 'Ferdy'@'localhost';
```

```
PrakDB/Ferdy(060) >DROP USER 'Ferdy'@'localhost';
Query OK, 0 rows affected (0.005 sec)
```


fst.umsida.ac.id
Labterpadufst@umsida.ac.id
+62 812-2019-3773
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

DAFTAR PUSTAKA

Buku Ajar Basis Data. Penerbit UMSIDA Press. 2020

Basis Data, Khairi Budayawan, Mafy Media Literasi Indonesia, 2023.

Basis Data 2023 Konsep Dasar Membangun Database, Muhammad Ihksan, Herman Susilo, Nurul Abdillah, CV. Suluah Kato Khatulistiwa, 2023.

Desain Basis Data, M. Abu Jihad Plaza R, Penerbit Deepublish, 2021

Teori Basis Data. Ni Ketut Dwi Ari Jayanti. Penerbit ANDI. 2018.

Manajemen Basis Data Menggunakan MySQL. Robi Yanto, Penerbit Deepublish, 2016

Indra Astutik, I. R. ., & Wignya Radhitya, Y. (2024). Sistem Informasi Pemesanan Jasa Fotografer Berbasis Web Pada Studio Fotograferku. Journal of Technology and System Information, 1(1), 1â€“15. https://doi.org/10.47134/jtsi.v1i1.2142

Indra Astutik, I. R. (2023) â€œResponsive website for TK Aisyiyah VI Candi based on content management system wordpressâ€•, Jurnal Mantik, 7(3), pp. 2144 2153. doi: 10.35335/mantik.v7i3.4293.

Indra Astutik, I. R. ., Imanudin, G. F., Rosid, M. A., & Aji, S. (2024). Designing a Posyandu Scheduling System using a Data-base to Improve Patient Service [Perancangan Sistem Penjadwalan Posyandu Menggunakan Basis Data untuk Meningkatkan Layanan Pasien]. Indonesian Journal of Applied Technology, 1(1), 12â€“24. https://doi.org/10.47134/ijat.v1i1.2109

Rini, D. D. O. ., Hanif, A. ., & Astutik, I. R. I. . (2023). TATA KELOLA KEUANGAN KOPERASI DINAR AMANTA MELALUI PENERAPAN TEKNOLOGI SISTEM INFORMASI. Jurnal Abdimas Bina Bangsa, 4(1), 379-383. https://doi.org/10.46306/jabb.v4i1.377

fst.umsida.ac.id
+62 812-2019-3773
Labterpadufst@umsida.ac.id
Jl. Raya Gelam No.250, Candi, Sidoarjo, Jatim 61271

KARTU ASISTENSI
PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH SIDOARJO
2025 – 2026

Nama : Mochammad Ferdyansyah

NIM : 251080200060

Kelompok : 6

No	Judul Praktikum	Tanggal Praktikum	Tanggal Asistensi	Catatan Asisten
1.	Basis Data, Model Data, Diagram E-R	07 Mei 2026	12/5 2026	Baik
2.	Structured Query Language (SQL)	07 Mei 2026	12/5 2026	Baik
3.	Data Definition Language (DDL)	07 Mei 2026	12/5 2026	Baik
4.	Data Manipulation Language (DML)	02 Juli 2026	6/7 2026	Baik
5.	Query dan View	02 Juli 2026	6/7 2026	Baik
6.	Data Control Language (DCL) / Hak Akses User	02 Juli 2026	6/7 2026	Baik

Mengetahui,

Dosen Praktikum

9/07 26

(Ika Ratna Indra Astutik, S.Kom., MT.)

Asisten Praktikum

6/7 2026

(Zianisa Azzahra)