

**LAPORAN TUGAS-TUGAS
MANAJEMEN BASIS DATA
SEMESTER GANJIL 2024/2025**



Oleh :

Ahmad Azka Al Husaini

NIM : 09021282227096

**TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA**

2024

KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya sehingga laporan yang berjudul " **Laporan Tugas-Tugas Manajemen Basis Data Semester Ganjil 2024/2025**" ini dapat diselesaikan dengan baik.

Laporan ini disusun sebagai bentuk dokumentasi dan rangkuman dari berbagai tugas yang telah dilaksanakan, yang bertujuan untuk memberikan panduan langkah-langkah pengelolaan basis data mulai dari instalasi, migrasi, hingga implementasi fitur keamanan dan optimalisasi performa serta paduan lainnya. Dalam laporan ini, pembaca akan menemukan pembahasan terstruktur yang mencakup berbagai aspek penting dalam pengelolaan basis data, seperti instalasi database, proses migrasi, pengamanan data dengan enkripsi, penerapan indexing, pemanfaatan struktur data B-Tree dan Hash, monitoring sistem, hingga transaksi berbasis prinsip ACID.

Harapan kami, laporan ini dapat digunakan sebagai panduan praktis yang dapat dimanfaatkan oleh mahasiswa, praktisi, maupun siapa saja yang ingin mempelajari lebih lanjut tentang pengelolaan basis data secara efektif dan efisien. Kami menyadari bahwa laporan ini masih memiliki kekurangan, oleh karena itu kami membuka diri untuk menerima kritik dan saran yang membangun demi kesempurnaan laporan di masa mendatang. Akhir kata, kami mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan laporan ini. Semoga laporan ini memberikan manfaat dan juga inspirasi bagi pembaca.

Palembang, 25 November 2024



Ahmad Azka Al Husaini
NIM 09021282227096

DAFTAR ISI

KATA PENGANTAR.....	ii
BAB I.....	2
PENDAHULUAN	2
BAB II.....	3
INSTALASI DATABASE.....	3
2.1 Pendahuluan.....	3
2.2 Instalasi MySQL.....	3
2.3 Instalasi Oracle	5
2.4 Instalasi SQLite	6
2.5 Instalasi PostgreSQL.....	8
BAB II.....	10
MIGRASI DATABASE.....	10
3.1 MySQL to ORACLE.....	10
3.2 MySQL to SQLite	12
3.3 MySQL to MongoDB.....	14
3.4 MySQL to PostgreSQL	16
BAB III.....	19
USER.....	19
3.1 Membuat User	19
3.2 Penerapan Hashing Password	26
BAB IV	30
SECURITY	30
4.1 Pengertian dan Tahapan Security	30
BAB V	32
INDEXING.....	32
5.1 Index.....	32
5.2 B-Tree.....	33
5.3 Hash Index.....	34
BAB VI.....	36
MONITORING	36
6.1 Pengertian Monitoring Databases.....	36
6.2 Manfaat Monitoring Databases.....	36
6.3 Langkah-langkah Monitoring Databases	37
BAB VII.....	39

REPLIKASI DATABASE.....	39
7.1 Pengertian Replikasi Databases	39
7.2 Langkah-langkah Replikasi Databases	39
BAB VIII	41
TRANSAKSI (ACID)	41
8.1 Pengertian Transaksi (ACID)	41
8.2 Implementasi Transaksi (ACID)	41
DAFTAR PUSTAKA.....	46

BAB I

PENDAHULUAN

Basis data merupakan komponen utama dalam pengelolaan informasi yang efisien dan akurat. Dalam era digital saat ini, basis data tidak hanya digunakan untuk menyimpan data, tetapi juga untuk mendukung berbagai operasi penting, seperti transaksi, analisis, dan pengambilan keputusan. Oleh karena itu, pemahaman mendalam tentang instalasi, pengelolaan, dan pengoptimalan basis data sangat diperlukan bagi individu maupun organisasi.

Namun, tantangan yang muncul dalam pengelolaan basis data, seperti keamanan, kecepatan akses, serta efisiensi proses transaksi, sering kali menjadi hambatan. Untuk mengatasi tantangan tersebut, diperlukan pendekatan sistematis yang mencakup instalasi, migrasi, enkripsi, pengindeksan, monitoring, hingga replikasi basis data. Dengan pendekatan ini, pengelolaan basis data dapat dilakukan secara efektif dan aman.

Laporan ini bertujuan untuk mendokumentasikan tugas-tugas dalam mata kuliah Manajemen Basis Data ke dalam bentuk laporan yang sistematis. Laporan ini juga memberikan panduan langkah-langkah dalam pengelolaan basis data, termasuk instalasi, migrasi, enkripsi, pengindeksan, monitoring, replikasi, dan transaksi. Panduan ini diharapkan dapat menjadi referensi yang bermanfaat bagi praktisi, mahasiswa, maupun pengembang basis data.

Melalui laporan ini, diharapkan pembaca dapat memahami langkah-langkah pengelolaan basis data secara terstruktur, serta mampu mengimplementasikan teknik-teknik dasar yang diperlukan untuk mengoptimalkan sistem basis data.

BAB II

INSTALASI DATABASE

2.1 Pendahuluan

Databases adalah kumpulan data terkait yang disimpan bersama dengan redundansi terkontrol untuk melayani satu atau lebih aplikasi secara optimal. Database Management System (DBMS) adalah perangkat lunak yang digunakan untuk mengelola basis data. DBMS bertanggung jawab atas penyimpanan, pengambilan, pembaruan, dan penghapusan data dalam databases (Dirgantara et al., 2023). DBMS dikategorikan berdasarkan distribusi database, model data, dan jumlah user yang didukung. Jenis DBMS yang pertama ialah Relational Database Management System (RDBMS), RDBMS menyimpan data dalam tabel dengan baris dan kolom, dan paling baik untuk data terstruktur. Basis data relasional juga dikenal sebagai basis data SQL. Basis data ini bagus untuk aplikasi yang memerlukan integritas data tinggi dan kueri SQL yang kompleks. Jenis DBMS yang kedua ialah Non-relational databases Management System (NRDBMS), NRDBMS Menyimpan data dalam berbagai format non-tabular, seperti key-value pairs, dokumen JSON, atau grafik. Basis data ini juga dikenal sebagai basis data NoSQL, yang merupakan singkatan dari "Not Only SQL". NRDBMS paling baik untuk data yang tidak terstruktur atau terus berkembang, dan untuk aplikasi yang memerlukan volume data besar, latensi rendah, dan model data yang fleksibel.

2.2 Instalasi MySQL

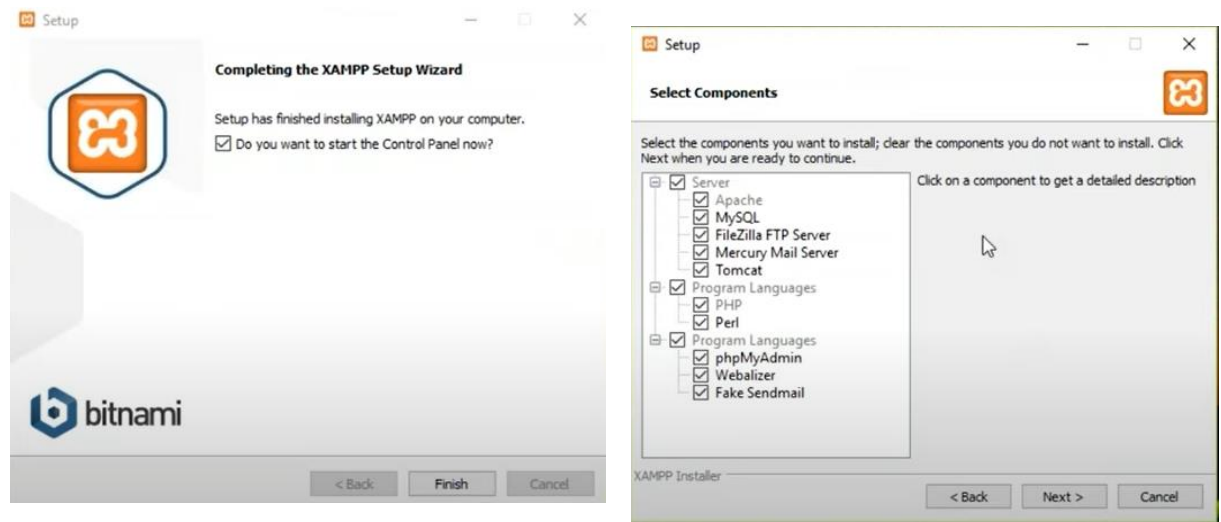
MySQL adalah sistem manajemen database relasional berbasis SQL dan arsitektur client-server. MySQL termasuk salah satu DBMS yang paling banyak digunakan, karena kompatibel dengan banyak platform komputasi seperti distribusi Linux, Windows, dan macOS. MySQL juga mendukung C, C++, Java, Perl, PHP, Python, dan Ruby. MySQL merupakan salah satu RDBMS sehingga sistem database ini menggunakan format tabel untuk mengatur data dan menjaga relasi antar-elemen.

Langkah-langkah instalasi :

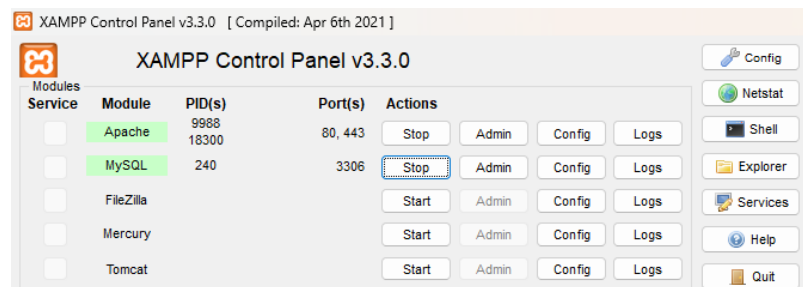
Pertama ialah klik download pada link berikut :

<https://www.apachefriends.org/download.html>

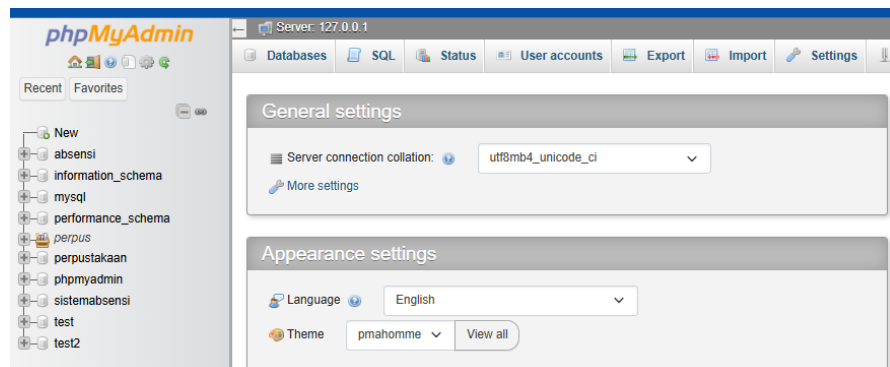
Proses instalasi



Setelah berhasil Setup, aktifkan Apache dan MySQL pada XAMPP



Uji coba melalui PHPMyAdmin



2.3 Instalasi Oracle

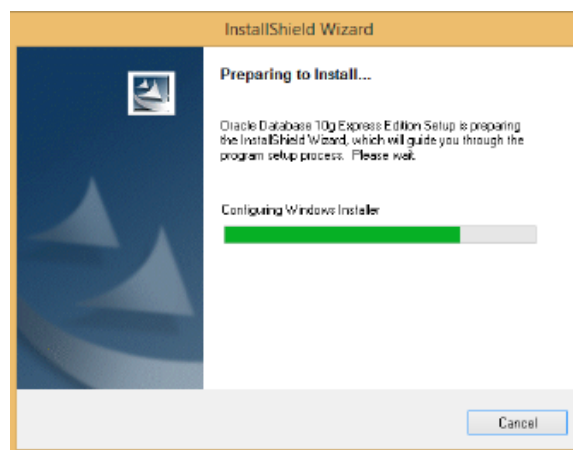
Menurut peringkat DB-Engine, Oracle adalah DBMS paling populer. Oracle merupakan RDBMS dengan arsitektur yang terbagi antara struktur logis dan fisik. Salah satu fitur terbaik database Oracle adalah komputasi grid perusahaan yang menggunakan penyimpanan fisik modular dan server yang ukurannya bisa disesuaikan oleh pengguna.

Langkah-langkah instalasi :

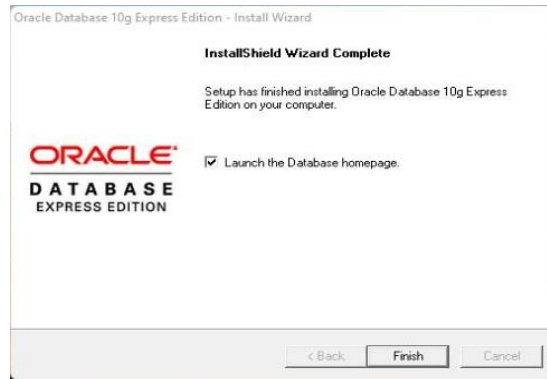
Pertama ialah klik download pada link berikut :

<https://www.oracle.com/id/database/technologies/oracle-database-software-downloads.html>

Kemudian lakukan proses instalasi oracle dengan melakukan run



Setelah itu, maka instalasi berhasil.



Uji coba melalui CMD

```
Command Prompt - sqlplus
Microsoft Windows [Version 10.0.22000.2176]
(c) Microsoft Corporation. All rights reserved.

C:\Users\gamer>SQL+
'SQL+' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\gamer>SQL>
The syntax of the command is incorrect.

C:\Users\gamer>sqlplus

SQL*Plus: Release 10.2.0.1.0 - Production on Thu Aug 31 11:06:31 2023
Copyright (c) 1982, 2005, Oracle. All rights reserved.

Enter user-name: SYSTEM
Enter password:
ERROR:
ORA-01017: invalid username/password; logon denied

Enter user-name: SYSTEM
Enter password:

Connected to:
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production
SQL> _
```

2.4 Instalasi SQLite

SQLite adalah library bahasa C yang mengimplementasikan SQL database engine yang kecil, cepat, mandiri, andal, dan berfitur lengkap. SQLite terpasang di semua ponsel dan sebagian besar komputer dan disertakan dalam banyak aplikasi lain yang digunakan orang setiap hari.

Langkah-langkah instalasi :

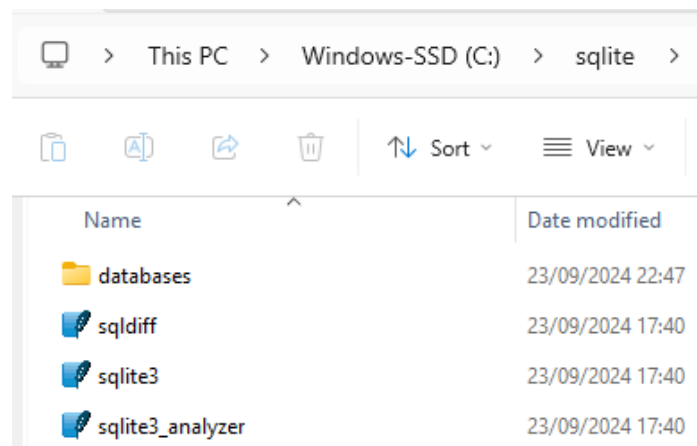
Pertama ialah klik download pada link berikut

<https://www.sqlite.org/index.html>

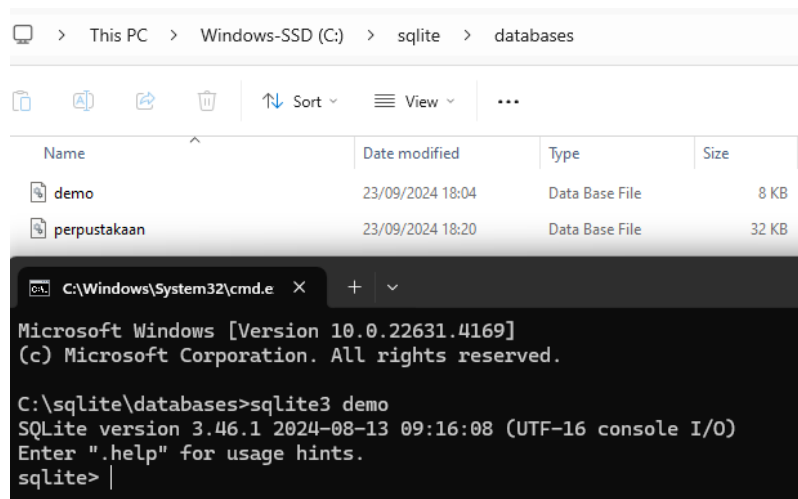
Setelah berhasil diinstall, atur direktori sesuai yang kita inginkan, contohnya seperti ini

C:\sqlite

Kemudian buat folder databases



Uji coba pada database demo



2.5 Instalasi PostgreSQL

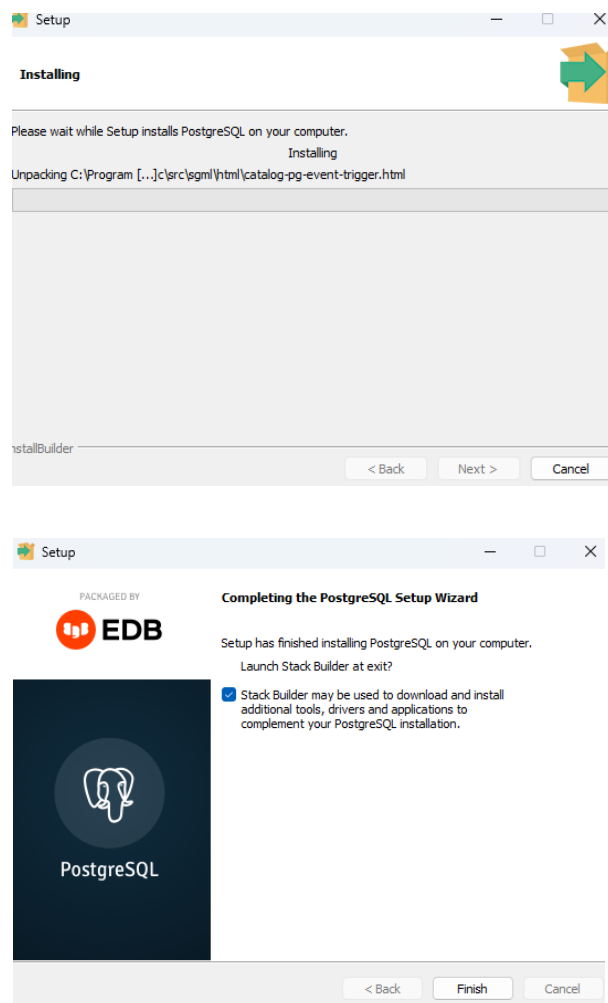
PostgreSQL adalah sistem manajemen database open-source tingkat perusahaan. DBMS ini mendukung SQL untuk kueri relasional, dan JSON untuk kueri non-relasional.

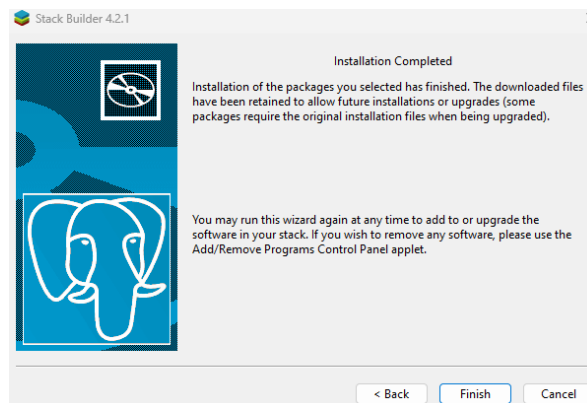
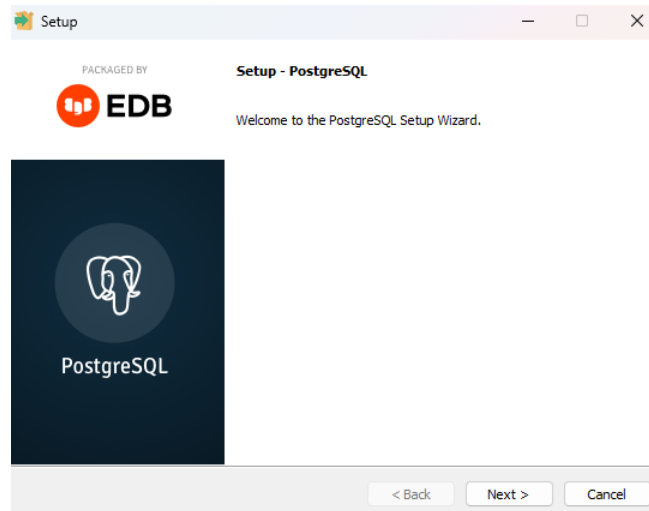
Langkah-langkah instalasi :

Pertama ialah klik download pada link berikut

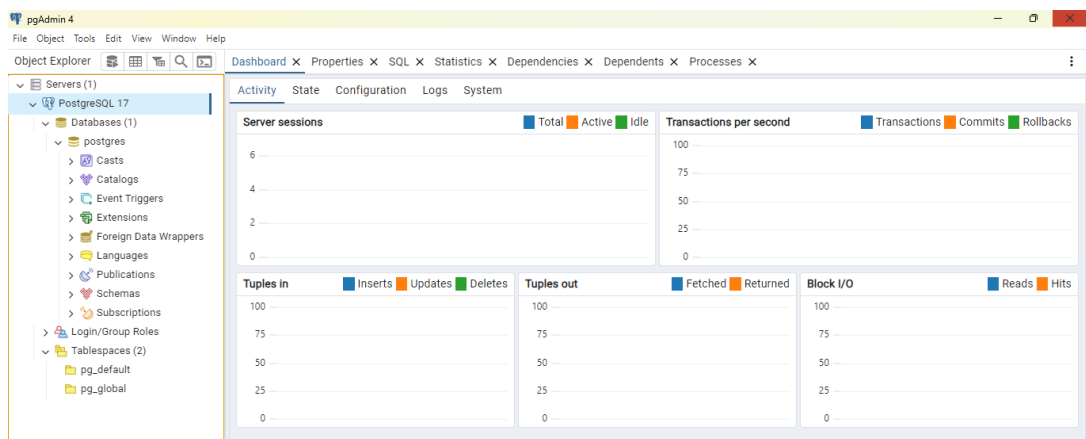
<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

Proses Instalasi





Uji coba melalui pgAdmin4



BAB II

MIGRASI DATABASE

Migrasi database adalah proses memindahkan data dari satu atau beberapa basis data sumber ke satu atau beberapa basis data target.

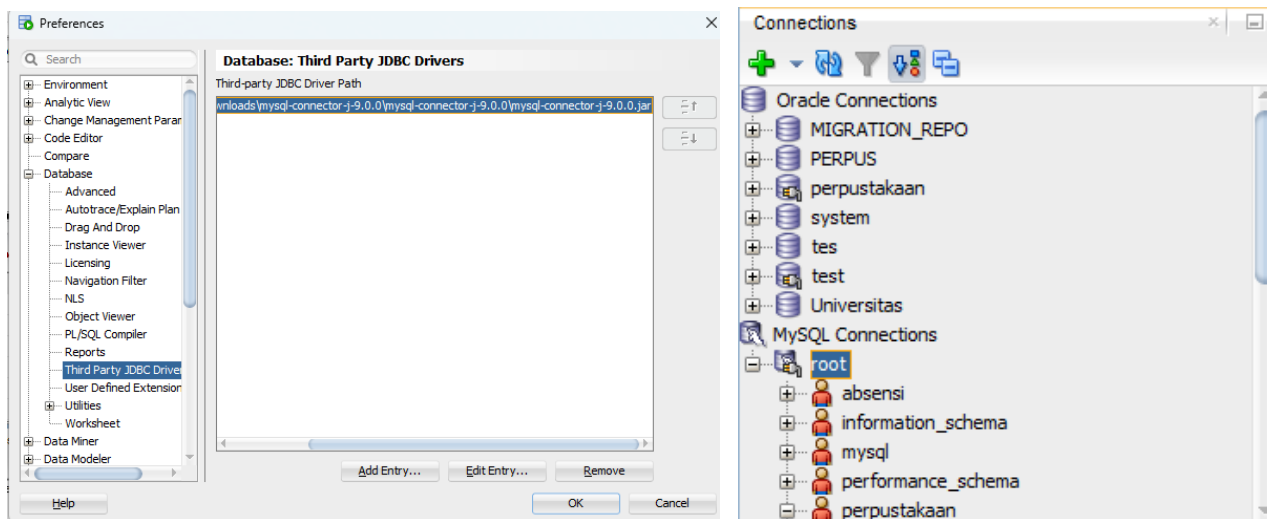
2.1 MySQL to ORACLE

Tools yang digunakan ialah ORACLE SQL Developer.

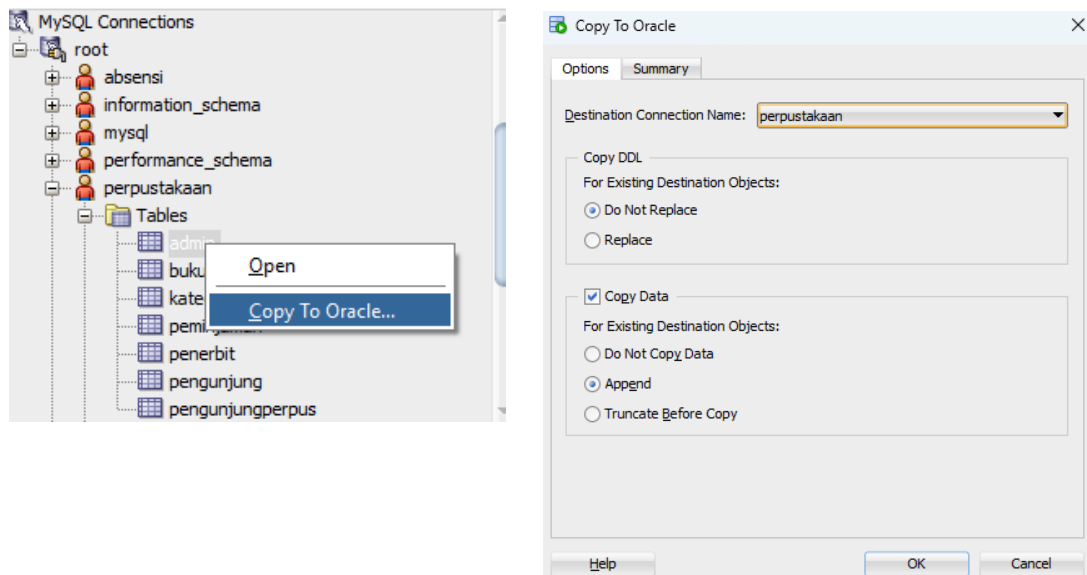
Ini merupakan database dari Mysql yang akan saya migrasikan ke Oracle



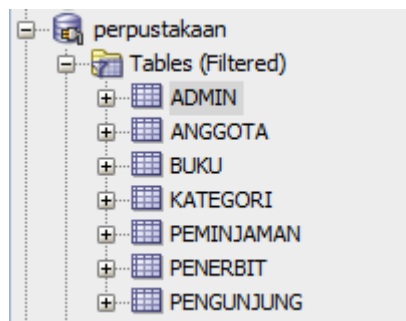
Langkah yang pertama saya lakukan ialah saya menghubungkan MYSQL ke ORACLE SQL Developer dengan **JDBC driver for MySQL connection** yang saya download di internet.



Setelah itu, saya menyalin tabel ke perpustakaan dari MySQL Connections ke Oracle Connections



Setelah itu, tabel berhasil dimigrasi

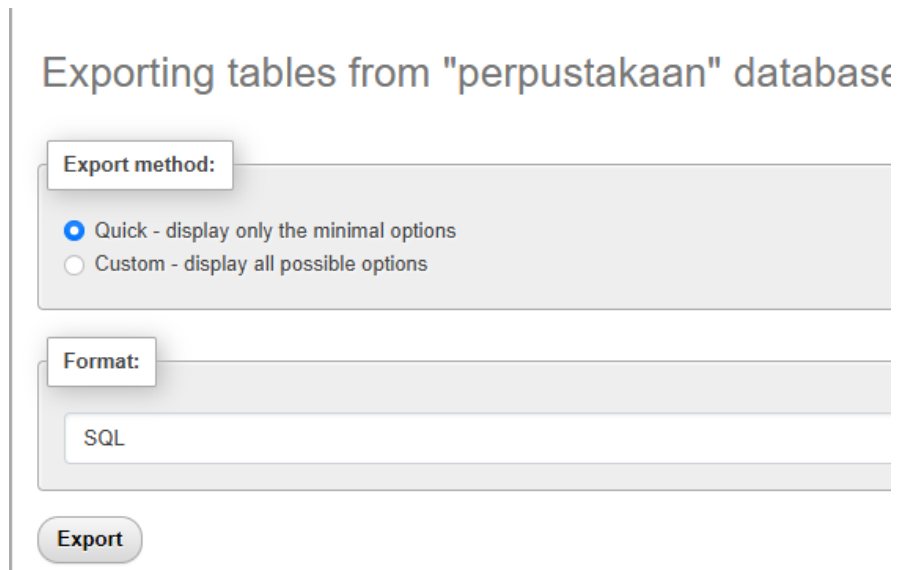


```
ALTER TABLE admin ADD PRIMARY KEY (ID_admin);
ALTER TABLE anggota ADD PRIMARY KEY (id_anggota);
ALTER TABLE buku ADD PRIMARY KEY (no_buku);
ALTER TABLE kategori ADD PRIMARY KEY (no_kategori);
ALTER TABLE peminjaman ADD PRIMARY KEY (ID_peminjaman);
ALTER TABLE penerbit ADD PRIMARY KEY (ID_penerbit);
ALTER TABLE pengunjung ADD PRIMARY KEY (NIK);

ALTER TABLE admin ADD CONSTRAINT fkadmin FOREIGN KEY (ID_peminjaman) REFERENCES peminjaman (ID_peminjaman) ON DELETE CASCADE;
ALTER TABLE anggota ADD CONSTRAINT fkanggota FOREIGN KEY (NIK) REFERENCES pengunjung (NIK) ON DELETE CASCADE;
ALTER TABLE buku ADD CONSTRAINT fk2_buku FOREIGN KEY (ID_penerbit) REFERENCES penerbit (ID_penerbit);
ALTER TABLE buku ADD CONSTRAINT fk_buku FOREIGN KEY (no_kategori) REFERENCES kategori (no_kategori) ON DELETE CASCADE;
ALTER TABLE peminjaman ADD CONSTRAINT fkpeminjaman FOREIGN KEY (no_buku) REFERENCES buku (no_buku) ON DELETE CASCADE;
```

2.2 MySQL to SQLite

Yang pertama saya lakukan ialah ekspor database dari MySQL



Exporting tables from "perpustakaan" database

Export method:

☒ Quick - display only the minimal options
☐ Custom - display all possible options

Format:

SQL

Export

Kemudian buka teks editor, dan perbaiki kodenya.

Setelah itu saya memasukkan querynya melalui cmd

```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\sqlite\databases>sqlite3 perpustakaan.db
SQLite version 3.46.1 2024-08-13 09:16:08 (UTF-16 console I/O)
Enter ".help" for usage hints.
sqlite> .databases
main: C:\sqlite\databases\perpustakaan.db r/w
sqlite> CREATE TABLE 'kategori' (
(x1...> 'no_kategori' varchar(10) NOT NULL PRIMARY KEY,
(x1...> 'Jenis' varchar(20)
(x1...> );
sqlite> CREATE TABLE 'penerbit' (
(x1...> 'ID_penerbit' varchar(10) NOT NULL PRIMARY KEY,
(x1...> 'Nama_penerbit' varchar(50)
(x1...> );
sqlite> CREATE TABLE 'buku' (
(x1...> 'no_buku' varchar(10) NOT NULL PRIMARY KEY,
(x1...> 'judul' varchar(100),
(x1...> 'ID_penerbit' varchar(10),
(x1...> 'tahun_terbit' int(11),
(x1...> 'no_kategori' varchar(10),
(x1...> FOREIGN KEY ('ID_penerbit') REFERENCES 'penerbit' ('ID_penerbit'),
(x1...> FOREIGN KEY ('no_kategori') REFERENCES 'kategori' ('no_kategori')
(x1...> );
sqlite> CREATE TABLE 'pengunjung' (
(x1...> 'NIK' varchar(20) NOT NULL PRIMARY KEY,
(x1...> 'nama' varchar(50),
(x1...> 'Alamat' varchar(100),
(x1...> 'email' varchar(50),
(x1...> 'Umur' int(11),
(x1...> 'jenis_kelamin' char(2)
(x1...> );
```

```

sqlite> CREATE TABLE `anggota` (
(x1...> `id_anggota` integer NOT NULL PRIMARY KEY AUTOINCREMENT,
(x1...> `email` varchar(50),
(x1...> `password_hash` varchar(255),
(x1...> `nama` varchar(50),
(x1...> `NIK` varchar(20),
(x1...> `Alamat` varchar(100),
(x1...> `Umur` int(11),
(x1...> `jenis_kelamin` char(2),
(x1...> FOREIGN KEY (`NIK`) REFERENCES `pengunjung` (`NIK`)
(x1...> );

```

```

sqlite> CREATE TABLE `peminjaman` (
(x1...> `ID_peminjaman` varchar(10) NOT NULL PRIMARY KEY,
(x1...> `Tanggal_pinjam` varchar(30),
(x1...> `Tanggal_kembali` varchar(30),
(x1...> `Tarif_denda` int(11),
(x1...> `Keterangan_denda` varchar(20),
(x1...> `id_anggota` integer NOT NULL,
(x1...> `no_buku` varchar(10),
(x1...> FOREIGN KEY (`id_anggota`) REFERENCES `anggota` (`id_anggota`),
(x1...> FOREIGN KEY (`no_buku`) REFERENCES `buku` (`no_buku`)
(x1...> );

```

```

sqlite> CREATE TABLE `admin` (
(x1...> `ID_admin` varchar(10) NOT NULL,
(x1...> `Username_admin` varchar(20) DEFAULT NULL,
(x1...> `ID_peminjaman` varchar(10) DEFAULT NULL,
(x1...> `password` varchar(255) NOT NULL,
(x1...> FOREIGN KEY (`ID_peminjaman`) REFERENCES `peminjaman` (`ID_peminjaman`)
(x1...> );

```

```

sqlite> INSERT INTO `kategori` (`no_kategori`, `Jenis`) VALUES
...> ('001', 'Teknologi'),
...> ('002', 'Sejarah'),
...> ('003', 'Fiksi'),
...> ('004', 'Agama'),
...> ('005', 'Sains'),
...> ('006', 'Politik'),
...> ('007', 'Olahraga'),
...> ('008', 'Bahasa'),
...> ('009', 'Seni'),
...> ('010', 'Ekonomi');
sqlite>
sqlite> INSERT INTO `penerbit` (`ID_penerbit`, `Nama_penerbit`) VALUES
...> ('PB01', 'Erlangga'),
...> ('PB02', 'Gramedia'),
...> ('PB03', 'Mizan'),
...> ('PB04', 'Kompas'),
...> ('PB05', 'A3'),
...> ('PB06', 'Maya Jaya'),
...> ('PB07', 'Ertiga'),
...> ('PB08', 'Matahari'),
...> ('PB09', 'Pustaka'),
...> ('PB10', 'Dahwa Jawa');
sqlite>
sqlite> INSERT INTO `buku` (`no_buku`, `judul`, `ID_penerbit`, `tahun_terbit`, `no_kategori`) VALUES
...> ('NB001', 'Pemrograman Dasar', 'PB01', 2005, '001'),
...> ('NB002', 'Batavia', 'PB02', 1930, '002'),
...> ('NB003', 'Gadis Kretek', 'PB03', 2014, '003'),
...> ('NB004', 'Bismillah', 'PB04', 2004, '004'),
...> ('NB005', 'IPA Dasar', 'PB05', 2020, '005'),
...> ('NB006', 'Demokrasi', 'PB06', 1999, '006'),
...> ('NB007', 'Raket Naya', 'PB07', 2019, '007'),
...> ('NB008', 'KBBI', 'PB08', 2000, '008'),
...> ('NB009', 'Cipta Syair', 'PB09', 2002, '009'),
...> ('NB010', 'Akuntansi', 'PB10', 2022, '010');
sqlite>
sqlite> INSERT INTO `pengunjung` (`NIK`, `nama`, `Alamat`, `email`, `Umur`, `jenis_kelamin`) VALUES
...> ('123450', 'Lina Damayanti', 'Jl. Kenanga 23', 'lina.damayanti@email.com', 31, 'P'),
...> ('123451', 'Andi Setiawan', 'Jl. Merdeka No. 123', 'andi.setiawan@email.com', 28, 'L'),
...> ('123452', 'Rini Sari', 'Jl. Harapan Indah 45', 'rini.sari@email.com', 35, 'P'),
...> ('123453', 'Budi Pratama', 'Jl. Pahlawan 67', 'budi.pratama@email.com', 42, 'L'),
...> ('123454', 'Maya Indah', 'Jl. Cempaka 89', 'maya.indah@email.com', 25, 'P'),
...> ('123455', 'Dika Wijaya', 'Jl. Kencana 34', 'dika.wijaya@email.com', 30, 'L'),
...> ('123456', 'Dewi Cahaya', 'Jl. Surya 56', 'dewi.cahaya@email.com', 29, 'P'),
...> ('123457', 'Fajar Prasetyo', 'Jl. Mawar 78', 'fajar.prasetyo@email.com', 33, 'L'),
...> ('123458', 'Siti Nurul', 'Jl. Anggrek 90', 'siti.nurul@email.com', 40, 'P'),
...> ('123459', 'Yoga Santoso', 'Jl. Dahlia 12', 'yoga.santoso@email.com', 27, 'L');
sqlite>

```



```

sqlite> INSERT INTO `anggota` (`id_anggota`, `email`, `password_hash`, `nama`, `NIK`, `Alamat`, `Umur`, `jenis_kelamin`) VALUES
...> (1, 'andi.setiawan@email.com', '$2y$10$pmBX0jm'nss.z7vp9mrzNu7D2GXE07We2NwJrCm8uCVlt4y5L12/a', 'Andi Setiawan', '123451', 'Jl. Merdeka No. 123', 28, 'L'),
...> (2, 'rini.sari@email.com', '$2y$10$jD31KG7qz/M4Rb.1F/ibrOPsJ90neERUPhD4sNSbKbYcgTRPPYMQu', 'Rini Sari', '123452', 'Jl. Harapan Indah 45', 35, 'P'),
...> (3, 'budi.pratama@email.com', '$2y$10$5j4w.udT3mbZ6rNQpeJPFh.B.tTRqD2FQFIva3CLdxv5GMOaNd2aIe', 'Budi Pratama', '123453', 'Jl. Pahlawan 67', 42, 'L'),
...> (4, 'maya.indah@email.com', '$2y$10$0T12wsQ9IBB23xkcFDC3kuM.paJJeNiMFOcH19GQKMTSGN9eQ5hy', 'Maya Indah', '123454', 'Jl. Cempaka 89', 25, 'P'),
...> (5, 'dika.wijaya@email.com', '$2y$10$usvjuItACztFTWtd4PnXeuSDXAwoGUR1985gc0Uhx/CzX86.8zoZ6', 'Dika Wijaya', '123455', 'Jl. Mencana 34', 30, 'L'),
...> (6, 'dewi.cahaya@email.com', '$2y$10$ay0aHh3TRBY5J3S3mBBeKtJ0RbWQ4JowAGydxNkZ/nAuccSLa52', 'Dewi Cahaya', '123456', 'Jl. Surya 56', 29, 'P'),
...> (7, 'fajar.prasetyo@email.com', '$2y$10$hyMDezgmZ0/onJiBRGGfuhN.DC9vdPvc8wA/aGcgeuY0uWXSUhpY', 'Fajar Prasetyo', '123457', 'Jl. Mawar 78', 33, 'L'),
...> (8, 'siti.nurul@email.com', '$2y$10$5A9h/Xn3zKdc/9FmNBpEe3DU4JHVI.Fbn4q5c0mtxKznH81EskE', 'Siti Nurul', '123458', 'Jl. Anggrek 90', 40, 'P'),
...> (9, 'yoga.santoso@email.com', '$2y$10$uFL9vL2Yq8BbOD9Mjren.0LdND8IulZbbS8eFZM6uqXLV9Q39Mux6', 'Yoga Santoso', '123459', 'Jl. Dahlia 12', 27, 'L'),
...> (10, 'lina.damayanti@email.com', '$2y$10$ZtdBNqAsbvDgfgAAvh5Xt046Y4VPT7ikiG88xfFwFv74Us/xk0/Ki', 'Lina Damayanti', '123450', 'Jl. Kenanga 23', 31, 'P');
sqlite>
sqlite> INSERT INTO `peminjaman` (`ID_peminjaman`, `Tanggal_pinjam`, `Tanggal_kembali`, `Tarif_denda`, `Keterangan_denda`, `ID_anggota`, `no_buku`) VALUES
...> ('PJM001', '05 January 2023', '15 January 2023', 0, '-', '1', 'NB001'),
...> ('PJM002', '10 February 2023', '20 February 2023', 0, '-', '2', 'NB002'),
...> ('PJM003', '15 March 2023', '25 March 2023', 20000, 'Rusak', '3', 'NB003'),
...> ('PJM004', '20 April 2023', '30 April 2023', 50000, 'Hilang', '4', 'NB004'),
...> ('PJM005', '25 May 2023', '04 June 2023', 0, '-', '5', 'NB005'),
...> ('PJM006', '30 June 2023', '10 July 2023', 2000, 'Terlambat', '6', 'NB006'),
...> ('PJM007', '05 August 2023', '15 August 2023', 0, '-', '7', 'NB007'),
...> ('PJM008', '10 September 2023', '20 September 2023', 0, '-', '8', 'NB008'),
...> ('PJM009', '15 October 2023', '25 October 2023', 5000, 'Rusak', '9', 'NB009'),
...> ('PJM010', '20 November 2023', '30 November 2023', 0, '-', '10', 'NB010');
sqlite>

```

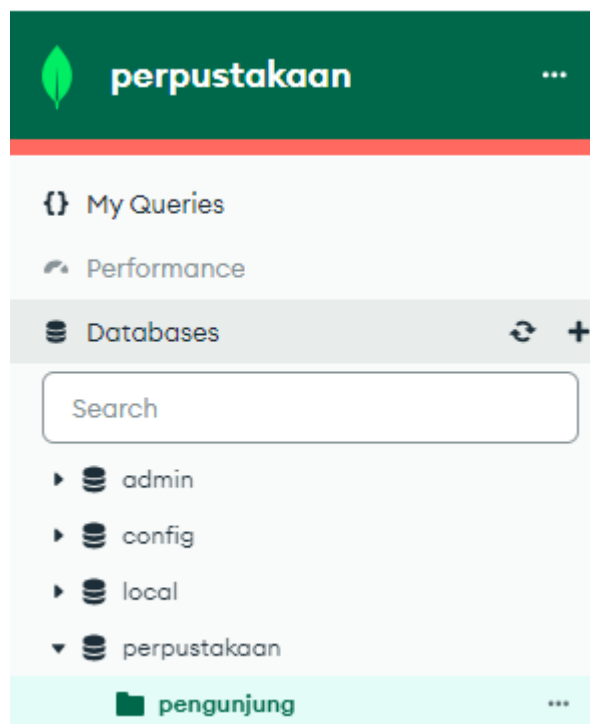
```

sqlite> INSERT INTO `admin` (`ID_admin`, `Username_admin`, `ID_peminjaman`, `password`) VALUES
...> ('ADM11', 'Upin', 'PJM001', '5e9c1555075c1c7a830634225dae087b93fb4d8b0d4b86eab0b1ae744d1c998c'),
...> ('ADM12', 'Badrul', 'PJM002', 'eac78f2a38e837c4cdd37d3656924ad2574b7b85cd81c1ced9903ac36d338cc1'),
...> ('ADM13', 'Ipin', 'PJM003', '028b46aae188657c274ef7aac4948c2bd4174aeb576ee9651039e783d505ce4'),
...> ('ADM14', 'Jarjit', 'PJM004', 'f6f749ed0f7ad330005f925ceda6987ef3d54195d40ea4b67e5a3d52d30a3b'),
...> ('ADM15', 'Susanti', 'PJM005', 'd3428b30a201e41663f1785c0c0d71c0e6aeed08dbbdc4cc407252eaf056beb3'),
...> ('ADM16', 'Mail', 'PJM006', '07ed6ccf6b6f393d18684d1d4f774639d44c7d2d2895fd30491ccc50614ed4ea'),
...> ('ADM17', 'Ehsan', 'PJM007', '0cf9b414aadfaec49915f3037b44937474a7d9573433eb0fc2ba48aec9fd87ea'),
...> ('ADM18', 'Fizi', 'PJM008', 'c694e2a5e4cdcd3a32ced90f7cd8e9163604be94d94e246e0f8e16cd64ddd9c0'),
...> ('ADM19', 'Ijat', 'PJM009', '6d9f827a6c3a35a07140dc9798bb069978560532225d9889e2c05e1df8b6787b'),
...> ('ADM20', 'Meimei', 'PJM010', '46b780b3502b047e5222a75b86187ee8a00bf3c51b101adfc479cfff0541947e4');
sqlite>

```

2.3 MySQL to MongoDB

Yang pertama saya lakukan ialah buka MongoDB Compass, kemudian buat database



Kemudian saya import database dari mysql kedalam bentuk table yang diolah menggunakan SQLIE.exe, sehingga hasilnya dalam bentuk CSV seperti ini

	A	B	C	D	E	F	G	H
1	NIK,nama,Alamat,email,Umur,jenis_Kelamin							
2	123450,	"Lina Damayanti",	"Jl. Kenanga 23",	"lina.damayanti@email.com",	"31",	"P"		
3	123451,	"Andi Setiawan",	"Jl. Merdeka No. 123",	"andi.setiawan@email.com",	"28",	"L"		
4	123452,	"Rini Sari",	"Jl. Harapan Indah 45",	"rini.sari@email.com",	"35",	"P"		
5	123453,	"Budi Pratama",	"Jl. Pahlawan 67",	"budi.pratama@email.com",	"42",	"L"		
6	123454,	"Maya Indah",	"Jl. Cempaka 89",	"maya.indah@email.com",	"25",	"P"		
7	123455,	"Dika Wijaya",	"Jl. Kencana 34",	"dika.wijaya@email.com",	"30",	"L"		
8	123456,	"Dewi Cahaya",	"Jl. Surya 56",	"dewi.cahaya@email.com",	"29",	"P"		
9	123457,	"Fajar Prasetyo",	"Jl. Mawar 78",	"fajar.prasetyo@email.com",	"33",	"L"		
10	123459,	"Yoga Santoso",	"Jl. Dahlia 12",	"yoga.santoso@email.com",	"27",	"L"		

Setelah itu, import masing-masing table pada database perpustakaan

Setelah klik import data, kemudian klik import

Import

To collection perpustakaan.anggota

☒ Ignore empty strings

☐ Stop on errors

Specify Fields and Types [Learn more about data types](#)

<input checked="" type="checkbox"/> id_anggota	<input checked="" type="checkbox"/> email	<input checked="" type="checkbox"/> password_hash	<input checked="" type="checkbox"/> nama	<input checked="" type="checkbox"/> NIK
Int32	String	String	String	String
1	andi.setiawan@email.com	\$2y\$10\$pmbXOjYnss.z7vp9mrzNu7D2G...	Andi Setiawan	123451
2	rini.sari@email.com	\$2y\$10\$jD31KG7qz/MkRb.1F/1brOPsj9One...	Rini Sari	123452
3	budi.pratama@email.com	\$2y\$10\$j4w.udT3mbZ6rNQpeJPFh.B.tTRq...	Budi Pratama	123453
4	maya.indah@email.com	\$2y\$10\$0Tf2wsQ9IBB23xkcFDC3kuM.paJj...	Maya Indah	123454
5	dika.wijaya@email.com	\$2y\$10\$usvjuItACztFTWtdkHPnXeUSIXAw...	Dika Wijaya	123455

Cancel
Import

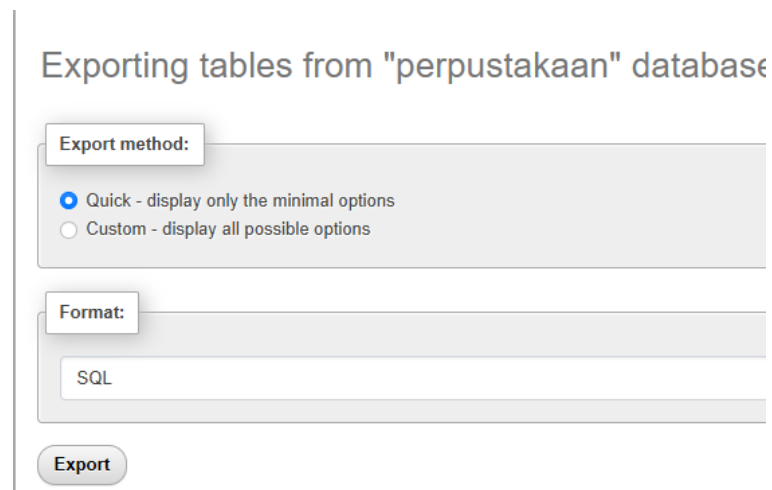
Database pun berhasil dimigrasi.



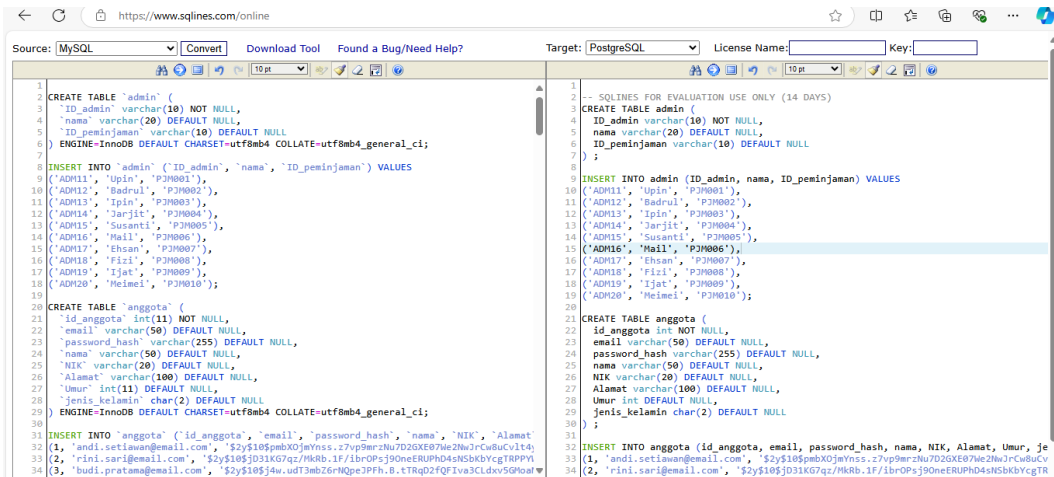
2.4 MySQL to PostgreSQL

Tools yang digunakan ialah SQLines

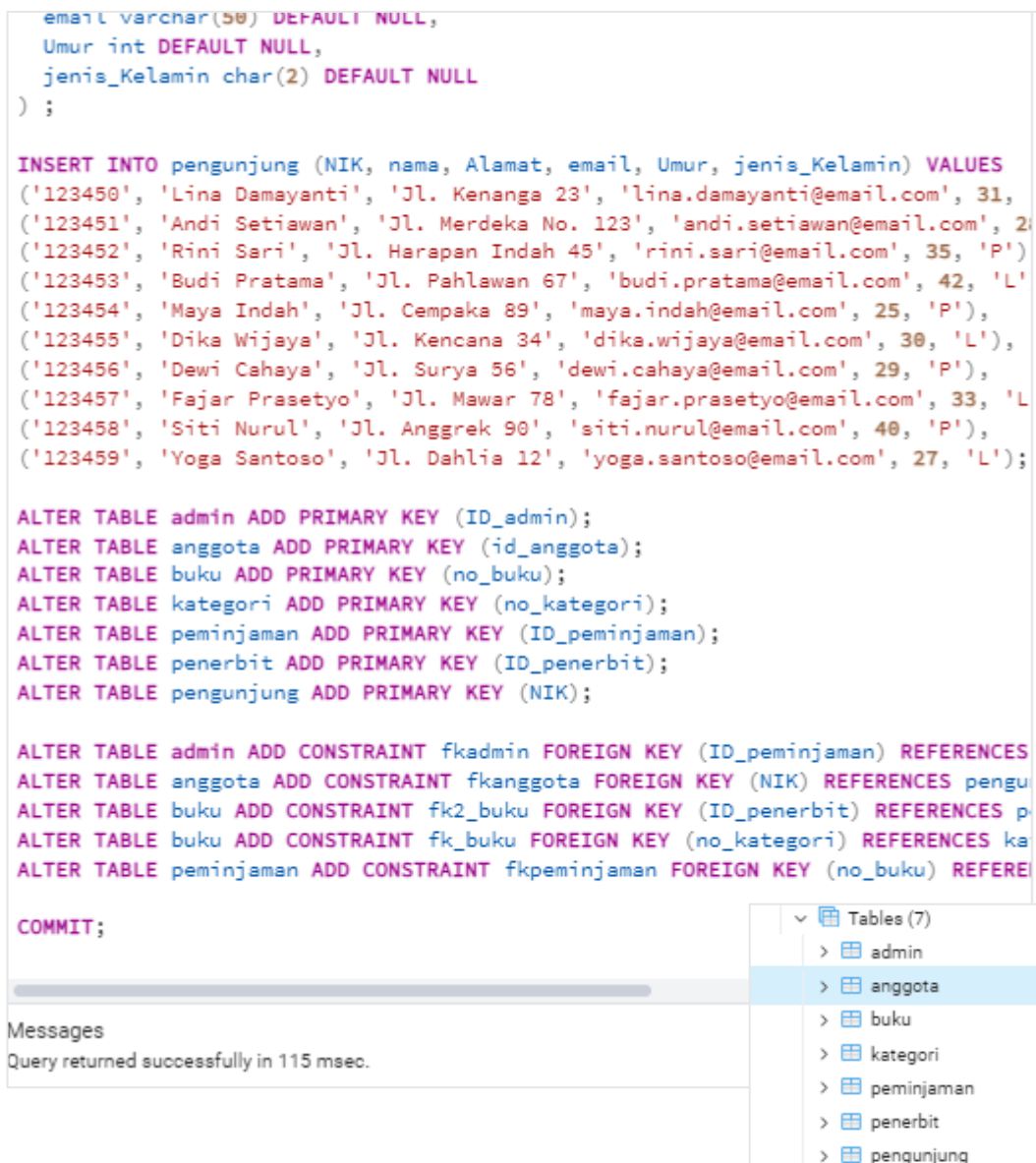
Yang pertama saya lakukan ialah ekspor terlebih dahulu database perpustakaan dari MySQL,



Setelah itu, buka SQLines dan pilih konversi Mysql ke PostgreSQL, kemudian dikonversi



Setelah dikonversi, masukkan query hasil konversi tersebut ke pgAdmin4 di database



Database pun berhasil dimigrasi.

Query

Query History

1

select * from anggota;

Scratch Pad

x

Data Output

Messages

Notifications

SQL

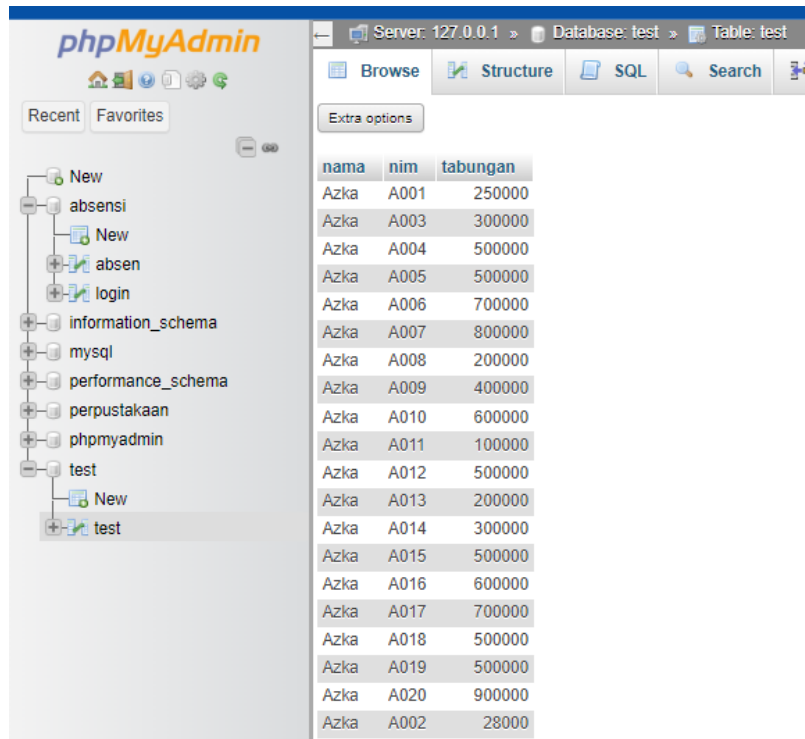
	id_anggota (PK) integer	email character varying (50)	password_hash character varying (255)	nama character varying (50)	nik character varying (20)	alamat character varying (100)	umur integer	jenis_kelamin character (2)
1	1	andi.setiawan@email.com	\$2y\$10\$pmBX0jmyYnss.z7vp9mrzNu7D2GXE07We2NwJrCw8uCVit4y...	Andi Setiawan	123451	Jl. Merdeka No. 123	28	L
2	2	rini.sari@email.com	\$2y\$10\$jD31K67qz/MkRb.1F/1brOPaj90neERUPhD4sNSbKbYcgTRPP...	Rini Sari	123452	Jl. Harapan Indah 45	35	P
3	3	budi.pratama@email.com	\$2y\$10\$j4w.udT3mbZ6rNQpeJFFh.B.tRqDZQFiva3CLdxv5GMoaNd...	Budi Pratama	123453	Jl. Pahlawan 67	42	L
4	4	maya.indah@email.com	\$2y\$10\$0T12wsQ9IBB23kkcFDC3kuM.paJJeNIWF0cH19XqQKMTSGN...	Maya Indah	123454	Jl. Cempaka 89	25	P
5	5	dika.wijaya@email.com	\$2y\$10\$usvjuItACztFTWTdKHPnXeUSIXAw0gURi985gc0UhXCzX86.8...	Dika Wijaya	123455	Jl. Kencana 34	30	L
6	6	dewi.cahaya@email.com	\$2y\$10\$ayOaM4h3TrBY5JSSNmB0ektj0RbNQkUoWAGydXNKZ/nAu...	Dewi Cahaya	123456	Jl. Surya 56	29	P
7	7	fajar.prasetyo@email.com	\$2y\$10\$HyNDtgNkZ0/onJIBREGFhuNr.DC9vdPxc8wA/sGCgeuYu8MX...	Fajar Prasetyo	123457	Jl. Mawar 78	33	L
8	8	siti.nurul@email.com	\$2y\$10\$A9h/Xn3zkdE/9FmNBqEe3DU4JHVlfbn4q5cqWmtxKznH8...	Siti Nurul	123458	Jl. Anggrek 90	40	P
9	9	yoga.santoso@email.com	\$2y\$10\$uFL9vL2Yg8BbQD9Mjren.OldND8lu1ZbbS8efZN6uqXLV9Q39...	Yoga Santoso	123459	Jl. Dahlia 12	27	L
10	10	lina.damayanti@email.com	\$2y\$10\$Ztd8NgAsbvDglfAAvH5XtO46Y4VPT7kiG88xjfwH74Us/vxO...	Lina Damayanti	123450	Jl. Kenanga 23	31	P

BAB III

USER

3.1 Membuat User

Table test

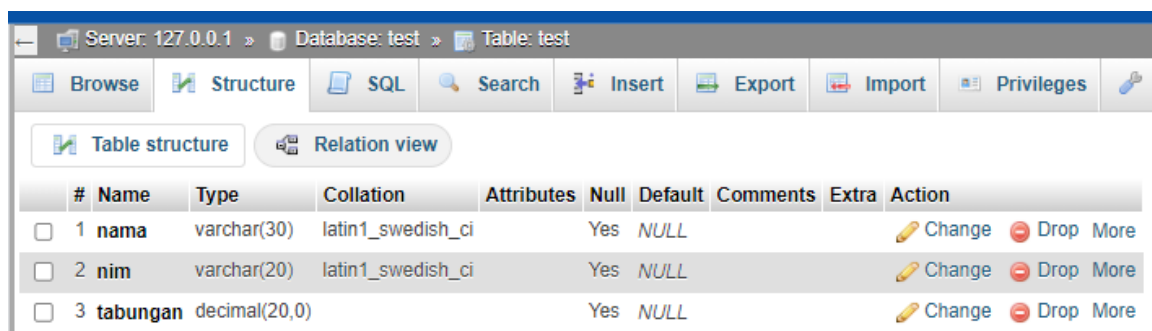


The screenshot shows the phpMyAdmin interface. On the left, a tree view shows the database structure with 'test' selected. The main panel displays the 'test' table with 20 rows of data. The table has three columns: 'nama', 'nim', and 'tabungan'.

nama	nim	tabungan
Azka	A001	250000
Azka	A003	300000
Azka	A004	500000
Azka	A005	500000
Azka	A006	700000
Azka	A007	800000
Azka	A008	200000
Azka	A009	400000
Azka	A010	600000
Azka	A011	100000
Azka	A012	500000
Azka	A013	200000
Azka	A014	300000
Azka	A015	500000
Azka	A016	600000
Azka	A017	700000
Azka	A018	500000
Azka	A019	500000
Azka	A020	900000
Azka	A002	280000

Table ini terdiri dari 20 nama, nim, serta tabungan.

Struktur Table



The screenshot shows the 'Table structure' view of the 'test' table. It displays the columns 'nama', 'nim', and 'tabungan' with their respective data types, collations, and attributes.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	nama	varchar(30)	latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 2	nim	varchar(20)	latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 3	tabungan	decimal(20,0)			Yes	NULL			Change Drop More

1) Query di komputer saya (dengan sistem operasi Windows 11)

Server: 127.0.0.1 » Database: test » Table: test

Column	Type	Function	Null	Value
nama	varchar(30)		<input type="checkbox"/>	Tamu
nim	varchar(20)		<input type="checkbox"/>	A021
tabungan	decimal(20,0)		<input checked="" type="checkbox"/>	780000

Go

✓ 1 row inserted.

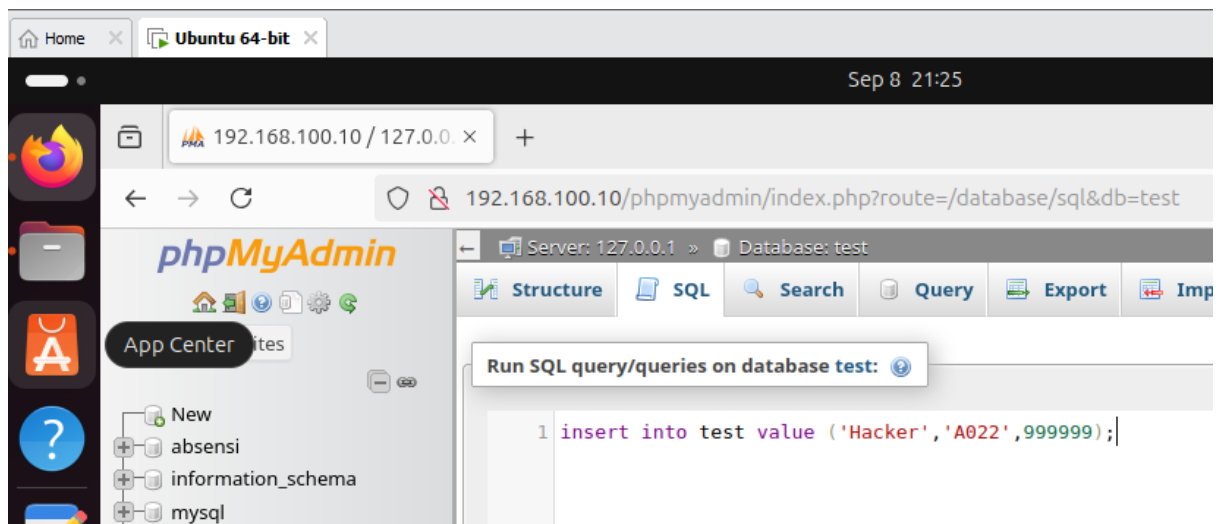
```
INSERT INTO `test` (`nama`, `nim`, `tabungan`) VALUES ('Tamu', 'A021', '780000');
```

[Edit inline] [Edit] [Create PHP code]

(Setelah diquery dengan insert)

nama	nim	tabungan
Azka	A001	250000
Azka	A003	300000
Azka	A004	500000
Azka	A005	500000
Azka	A006	700000
Azka	A007	800000
Azka	A008	200000
Azka	A009	400000
Azka	A010	600000
Azka	A011	100000
Azka	A012	500000
Azka	A013	200000
Azka	A014	300000
Azka	A015	500000
Azka	A016	600000
Azka	A017	700000
Azka	A018	500000
Azka	A019	500000
Azka	A020	900000
Azka	A002	280000
Tamu	A021	780000

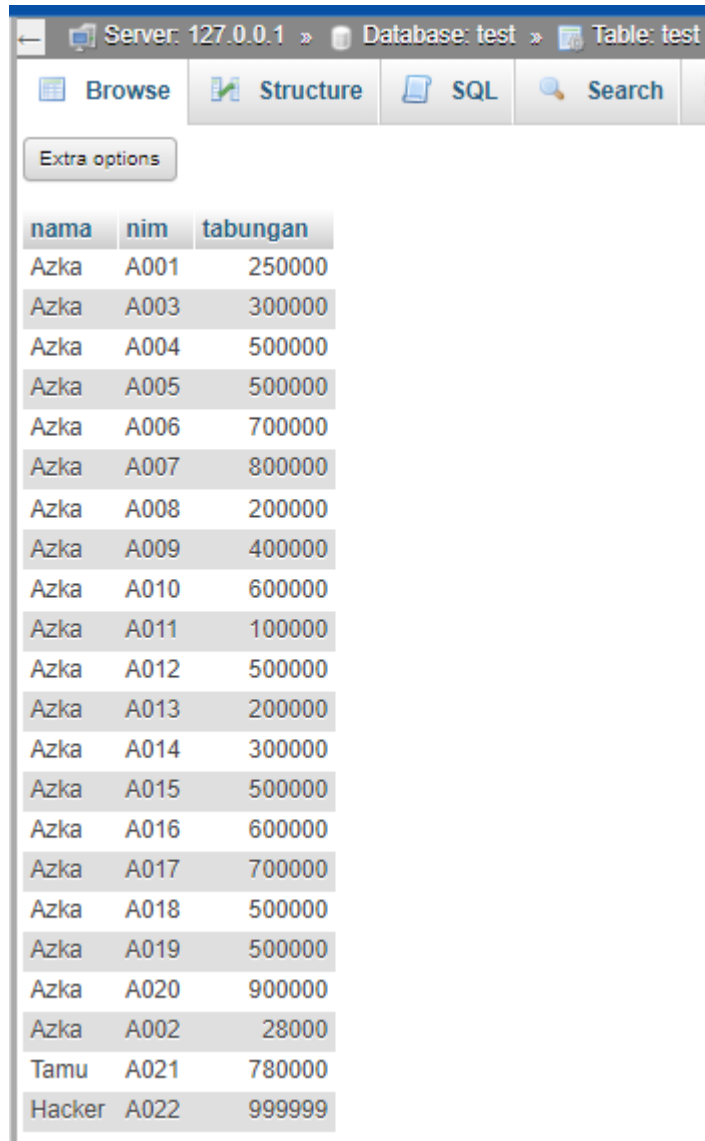
2) Query melalui VMware (dengan sistem operasi Linux) ke komputer saya



User dari linux dengan IP 192.168.72.128 mencoba memasuki Phpmyadmin dengan IP dari komputer saya, yaitu 192.168.100.10

✓ 1 row inserted. (Query took 0.0007 seconds.)

```
insert into test value ('Hacker', 'A022', 999999);
```



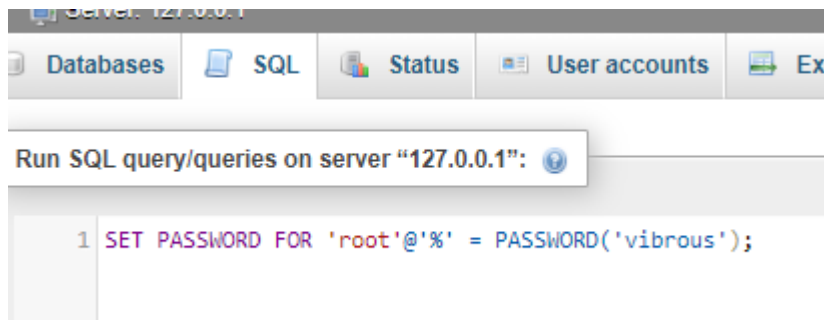
The screenshot shows the phpMyAdmin interface for a database named 'test' and a table named 'test'. The table structure is as follows:

nama	nim	tabungan
Azka	A001	250000
Azka	A003	300000
Azka	A004	500000
Azka	A005	500000
Azka	A006	700000
Azka	A007	800000
Azka	A008	200000
Azka	A009	400000
Azka	A010	600000
Azka	A011	100000
Azka	A012	500000
Azka	A013	200000
Azka	A014	300000
Azka	A015	500000
Azka	A016	600000
Azka	A017	700000
Azka	A018	500000
Azka	A019	500000
Azka	A020	900000
Azka	A002	28000
Tamu	A021	780000
Hacker	A022	999999

Pada komputer saya, table test bertambah 1 data, yaitu hacker. artinya user lain dapat memasuki ke database saya dan dapat melakukan query.

3) Kunci

Kunci Phpmyadmin



4) Buat User

```
Command Prompt - mysql -u X + v
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\gamer>cd C:/xampp/mysql/bin

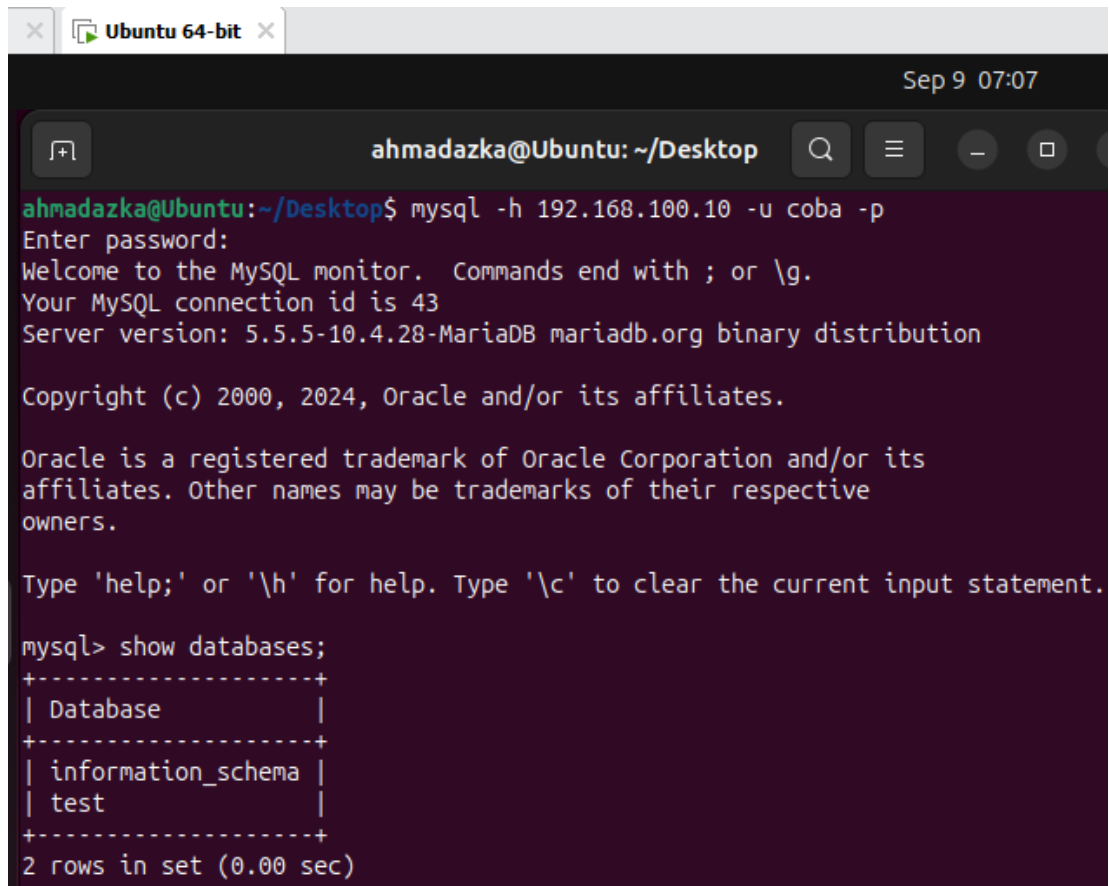
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.4.28-MariaDB mariadb.org binary distribution

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)]> create user 'coba'@'%' identified by 'coba';
Query OK, 0 rows affected (0.004 sec)
```

User dibuat di windows yang akan digunakan di VMware linux

A terminal window titled 'Ubuntu 64-bit' with a dark background. The user 'ahmadazka' is at the prompt 'ahmadazka@Ubuntu: ~/Desktop'. They have executed the command 'mysql -h 192.168.100.10 -u coba -p'. The terminal shows the MySQL login process, including a password prompt, a welcome message, connection ID 43, and server version 5.5.5-10.4.28-MariaDB. After typing 'show databases;', it displays a table with two rows: 'information_schema' and 'test'.

```
ahmadazka@Ubuntu:~/Desktop$ mysql -h 192.168.100.10 -u coba -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 5.5.5-10.4.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

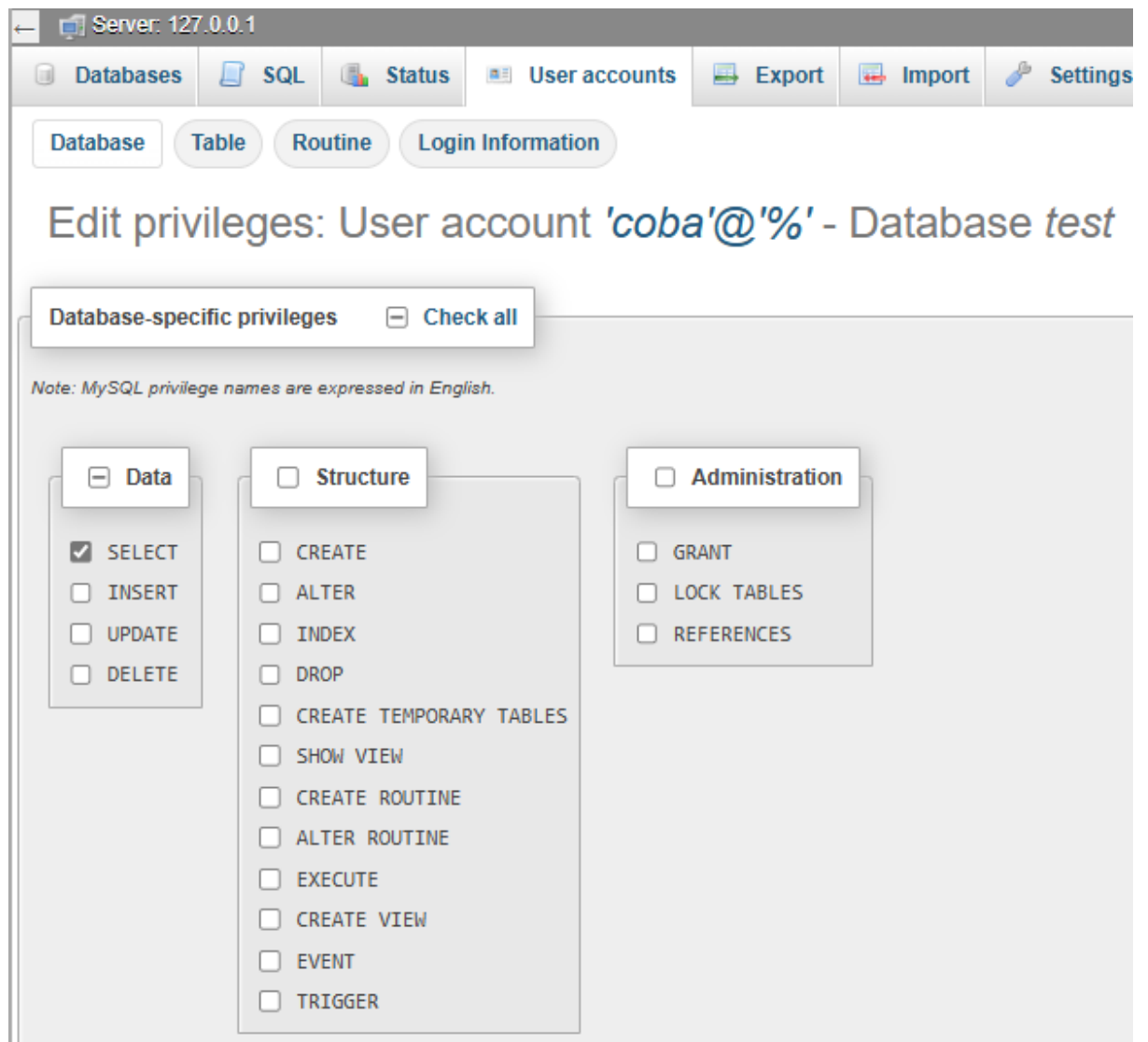
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

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

User dari linux dapat memasuki ke database yang ada di komputer, yaitu test.

Batasi User hanya dapat select saja pada database test



Uji coba

```
mysql> insert into test values ('Hacker','A027',777777);  
ERROR 1142 (42000): INSERT command denied to user 'coba'@'AhmadAzka83' for table 'test`.`test`
```

User dari linux tidak dapat melakukan insert

Batasi user hanya dapat select nama dan NIM pada table test, namun tidak dapat melihat tabungan.

Server: 127.0.0.1

Databases SQL Status User accounts Export Import Settings Replication

Table Login Information

Edit privileges: User account 'coba'@'%' - Database test - Table test

Table-specific privileges

Note: MySQL privilege names are expressed in English.

SELECT	INSERT	UPDATE	REFERENCES	DELETE
nama	nama	nama	nama	<input type="checkbox"/> DELETE
nim	nim	nim	nim	<input type="checkbox"/> CREATE
tabungan	tabungan	tabungan	tabungan	<input type="checkbox"/> DROP
				<input type="checkbox"/> GRANT
				<input type="checkbox"/> INDEX
				<input type="checkbox"/> ALTER
				<input type="checkbox"/> CREATE VIEW
				<input type="checkbox"/> SHOW VIEW
				<input type="checkbox"/> TRIGGER
				<input type="checkbox"/> DELETE HISTORY

Select all Or ☐ None Select all Or ☐ None Select all Or ☐ None Select all Or ☐ None

Go

Uji coba

```
mysql> select * from tabungan;  
ERROR 1146 (42S02): Table 'test.tabungan' doesn't exist
```

```
mysql> SELECT nim, nama FROM test.test;  
+-----+-----+  
| nim | nama |  
+-----+-----+  
| A001 | Azka |  
| A003 | Azka |  
| A004 | Azka |  
| A005 | Azka |  
| A006 | Azka |  
| A007 | Azka |  
| A008 | Azka |  
| A009 | Azka |  
| A010 | Azka |  
| A011 | Azka |  
| A012 | Azka |  
| A013 | Azka |  
| A014 | Azka |  
| A015 | Azka |  
| A016 | Azka |  
| A017 | Azka |  
| A018 | Azka |  
| A019 | Azka |  
| A020 | Azka |  
| A002 | Azka |
```

User dari linux tidak dapat melihat kolom tabungan

3.2 Penerapan Hashing Password

Pengelolaan pengguna dalam sistem basis data merupakan elemen krusial untuk memastikan data dapat diakses dengan aman dan sesuai dengan hak akses yang telah ditentukan. Setiap pengguna harus diautentikasi dan diotorisasi untuk mencegah akses yang tidak sah serta melindungi integritas data. Salah satu teknik yang sering digunakan untuk meningkatkan keamanan pada level pengguna adalah dengan menerapkan *hashing* pada data sensitif, seperti kata sandi.

Dalam penerapan pada laporan ini, Fungsi hashing digunakan di table yang memiliki kolom password di table. Hal ini digunakan saat sesi login, ketika membutuhkan kolom password, sehingga dapat memperkuat keamanan sistem. Berikut adalah kode dan implementasi hashing pada website login sederhana :

Connection.php

```
<?php
$servername = 'localhost';
$username = 'root';
$password = '';
$dbname = 'test2';

$conn = mysqli_connect($servername, $username, $password, $dbname);

if($conn->connect_error){
    die("Connection failed :".$conn->connect_error);
}
?>
```

Register.php (Sebelum dihashing)

```
<?php
include 'connection.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    $check_query = "SELECT * FROM login WHERE username = '$username'";
    $result = mysqli_query($conn, $check_query);
    $insert_query = "INSERT INTO login (username, password) VALUES ('$username', '$password')";

    if (mysqli_query($conn, $insert_query)) {
        echo "<h1>Registration successful!</h1>";
    } else {
        echo "<h1>Error: Registration failed.</h1>";
    }
}
```

Register.php

```
<?php
include 'connection.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    $hashed_password = password_hash($password, PASSWORD_DEFAULT);

    $stmt = $conn->prepare("SELECT * FROM login WHERE username = ?");
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        echo "<h1>Username already exists!</h1>";
    } else {
        $stmt = $conn->prepare("INSERT INTO login (username, password) VALUES (?, ?)");
        $stmt->bind_param("ss", $username, $hashed_password);

        if ($stmt->execute()) {
            echo "<h1>Registration successful!</h1>";
        } else {
            echo "<h1>Error: Registration failed.</h1>";
        }
    }
}
```

Index.php (untuk login)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
</head>
<body style="display: flex; align-items: center; justify-content: center; height: 100vh; flex-direction: column;">
    <form action="main.php" method="post" style="display: flex; flex-direction: column;">
        <h2 style="text-align: center;">Login</h2>
        <label name="username">Username</label>
        <input type="text" name="username"><br>
        <label name="password">Password</label>
        <input type="password" name="password"><br>
        <button type="submit">Submit</button>
    </form>
    <p><a href="register.php">Register</a></p>
</body>
</html>
```

main.php (Sebelum dihashing)

```

<?php
session_start();
include 'connection.php';

$username = $_POST['username'];
$password = $_POST['password'];

$login_query = "SELECT * FROM login WHERE username = '$username' AND password = '$password'";
$result = mysqli_query($conn, $login_query);

if (mysqli_num_rows($result) > 0) {
    $_SESSION['username'] = $username;
    echo "<h1>Login successful!</h1>";
} else {
    echo "<h1>Invalid Username or Password</h1>";
}
}
}

```

main.php

```

<?php
session_start();
include 'connection.php';

$username = $_POST['username'];
$password = $_POST['password'];

$stmt = $conn->prepare("SELECT * FROM login WHERE username = ?");
$stmt->bind_param("s", $username);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();

    // Verifikasi password hash
    if (password_verify($password, $row['password'])) {
        $_SESSION['username'] = $username;
    } else {
        echo "<h1>Invalid Username or Password</h1>";
        exit();
    }
} else {
    echo "<h1>Invalid Username or Password</h1>";
    exit();
}
}
}

```

```

<?php
session_start();
session_destroy();
header("Location: index.php");
exit();
}
}

```

logout.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
</head>

<body style="">
    <h1><?php echo "Welcome " . $_SESSION['username']; ?></h1>
    <form action="logout.php" method="post">
        <button type="submit" name="logout">Log out</button>
    </form>
</body>
</html>

```

Database

Server: 127.0.0.1 » Database: test2 » Table: login	
username	password
azka	\$2y\$10\$LpDL3EZYt0PBm0emZoFLHuPeeFhtvqSvR7ssH.zG0gc...

Note : Password sebelum di hashing: "azka123"

Uji coba login sebelum dihashing



A login form titled "Login". It contains a "Username" field with the text "azka" entered. Below it is a "Password" field with several dots indicating masked characters. At the bottom of the form is a "Submit" button and a link labeled "Register".

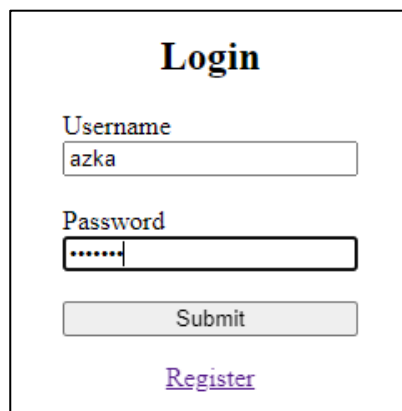
Berhasil



A message box displaying "Welcome azka" in a large, bold font. Below the message is a "Log out" button.

Note : Password yang dimasukkan : " ' OR '1'="

Uji coba login setelah dihashing



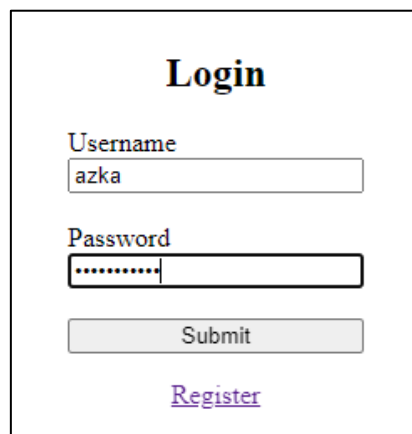
A login form titled "Login". It contains a "Username" field with the text "azka" entered. Below it is a "Password" field with several dots indicating masked characters. At the bottom of the form is a "Submit" button and a link labeled "Register".

Berhasil



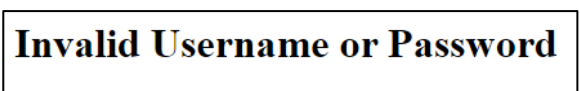
A message box displaying "Welcome azka" in a large, bold font. Below the message is a "Log out" button.

Note : Password yang dimasukkan : "azka123"



A login form titled "Login". It contains a "Username" field with the text "azka" entered. Below it is a "Password" field with several dots indicating masked characters. At the bottom of the form is a "Submit" button and a link labeled "Register".

Tidak berhasil



A message box displaying "Invalid Username or Password" in a bold font.

Note : Password yang dimasukkan : " ' OR '1'="

BAB IV

SECURITY

4.1 Pengertian dan Tahapan Security

Keamanan (*security*) dalam sistem basis data adalah aspek yang sangat penting untuk melindungi data dari akses yang tidak sah, penyalahgunaan, dan ancaman lainnya. Dalam pengelolaan basis data, keamanan mencakup berbagai mekanisme dan kebijakan yang dirancang untuk menjaga kerahasiaan, integritas, dan ketersediaan data (*confidentiality, integrity, availability*).

Dalam merancang keamanan, diperlukan berbagai tahapan untuk menjaga sistem salah satunya ialah *Role-Based Access Control* (RBAC). RBAC adalah metode memberikan hak akses kepada pengguna berdasarkan peran (*role*) yang mereka miliki dalam organisasi. Metode ini bertujuan untuk memastikan bahwa setiap pengguna hanya memiliki izin yang sesuai dengan tanggung jawab mereka, sehingga dapat mengurangi risiko akses yang tidak sah.

RBAC mengorganisasi hak akses dalam tiga komponen utama:

1. Pengguna (User): Individu yang mengakses sistem basis data.
2. Peran (Role): Kumpulan hak akses tertentu yang mewakili tugas atau fungsi tertentu.
3. Hak Akses (Permissions): Otorisasi untuk melakukan operasi tertentu pada data, seperti membaca (*read*), menulis (*write*), menghapus (*delete*), atau memperbarui (*update*).

Selain RBAC, diperlukan hashing password untuk memperkuat keamanan password, Hashing ialah fungsi yang mengkompresi pesan berukuran sembarang menjadi *string* yang berukuran *fixed*. Hashing digunakan untuk memastikan keaslian suatu data dan bahwa data tersebut belum dirusak. Hashing secara umum bersifat *irreversible* dan merupakan fungsi satu arah, artinya mudah untuk mengubah sebuah pesan menjadi hash, tetapi tidak dapat untuk membalikkan hash kembali ke pesan aslinya karena memerlukan daya komputasi yang sangat besar,

sehingga menjadikannya solusi yang andal untuk menyimpan data sensitif seperti password. Fungsi ini juga memiliki berbagai algoritma, seperti MD4, SHA, RIPEMD, WHIRPOL, dan TIGER.

Agar data tetap aman, terdapat juga replikasi, Replikasi database adalah sistem yang digunakan untuk menyalin, mendistribusikan data, serta melakukan sinkronisasi data antar database. Sistem menggunakan replikasi database untuk memastikan data tetap tersedia dan terlindungi. Dengan adanya replikasi, setiap perubahan yang terjadi pada basis data utama akan dicerminkan pada server cadangan. Hal ini memungkinkan pemulihan data secara cepat apabila terjadi kegagalan sistem atau kerusakan pada server utama.

Sistem audit dan logging juga diperlukan untuk memungkinkan administrator untuk melacak aktivitas pengguna dan mendeteksi tindakan mencurigakan. Dengan mencatat semua aktivitas, seperti login, query yang dijalankan, dan perubahan data, administrator dapat melakukan analisis forensik jika terjadi pelanggaran keamanan.

Pembatasan akses fisik ke server basis data juga merupakan elemen penting dalam strategi keamanan. Server harus ditempatkan di lokasi yang aman dengan akses terbatas hanya untuk personel yang berwenang. Hal ini termasuk pengamanan fisik seperti penggunaan kunci dan kamera pengawas untuk memantau area server.

Terakhir, pembaruan perangkat lunak secara rutin dan penerapan patch keamanan wajib perlu dilakukan untuk melindungi sistem dari kerentanan yang baru ditemukan. Basis data yang tidak diperbarui menjadi target utama bagi serangan siber. Oleh karena itu, administrator harus memantau perkembangan patch keamanan yang dirilis oleh vendor perangkat lunak dan menerapkannya secara berkala tanpa mengganggu operasi sistem.

BAB V

INDEXING

5.1 Index

Pada basis data, index merupakan sebuah struktur data yang berisi kumpulan *keys* beserta referensinya ke *actual data* di table. Tujuannya ialah untuk mempercepat proses penentuan lokasi data tanpa melakukan pencarian secara penuh ke seluruh data (full scan).

Manfaat dari menggunakan index ialah :

1. Jumlah data pada index jauh lebih kecil daripada data aslinya dan data-datanya tersimpan secara berurutan dalam bentuk struktur apapun sehingga pencarian pada *dataset* asli akan menjadi lebih singkat.
2. Index dapat disimpan di RAM dikarenakan ukuran *dataset* yang jauh lebih kecil. cost untuk baca atau tulis pada RAM jauh lebih kecil daripada pada Harddisk tidak bisa dipungkiri proses pencarian *keys* pada index sangat cepat.

Index dapat digunakan Ketika terdapat beberapa kondisi, yaitu :

1. Kolom sering digunakan dalam klausa *WHERE* atau dalam kondisi *join*.
2. Kolom berisi nilai dengan jangkauan yang luas.
3. Kolom berisi banyak nilai *null*.
4. Tabel berukuran besar dan sebagian besar *query* menampilkan data kurang dari 2-4%.

Namun, terdapat beberapa kondisi dimana index tidak diperlukan, yaitu :

1. Table berukuran kecil.
2. Kolom tidak sering digunakan sebagai kondisi dalam *query*.
3. Kebanyakan *query* menampilkan data lebih dari 2-4% dari seluruh data.
4. Table sering di-*update*.

5.2 B-Tree

B-Tree merupakan tipe standar dari *index* yang tersedia pada basis data yang berukuran besar, termasuk MySQL. B-tree adalah pohon yang menjaga data tetap terurut, dan dapat melakukan operasi akses, pencarian, penambahan, dan penghapusan dalam waktu logaritmik. Kelebihan lain dari *B-tree* adalah kemampuan untuk menangani operasi dalam jumlah yang tak terhingga. Secara struktur, B-tree merupakan generalisasi dari pohon pencarian biner. Bedanya, B-tree bersifat *multiway*, sehingga tiap simpul dapat mempunyai lebih dari dua anak. Biasanya sebuah B-tree mempunyai batas atas dan batas bawah jumlah simpul anak. Batasan-batasan tersebut berbeda-beda dan disesuaikan dengan kebutuhan implementasi tersebut. Misalnya dalam 2-3 tree, setiap simpul hanya boleh memiliki dua atau tiga anak. Simpul akar adalah pengecualian, karena simpul tersebut tidak mempunyai batas bawah jumlah simpul anak.

Setiap simpul dalam pada *B-tree* memiliki sejumlah kunci. Jumlah kunci adalah bilangan di antara d dan $2d$. Jika sebuah simpul dalam mempunyai $2d$ kunci, maka penambahan kunci pada simpul tersebut dapat dilakukan dengan membagi simpul tersebut menjadi dua simpul dengan d kunci, dan menambahkan satu kunci ke simpul orang tua. Proses ini akan menghasilkan dua simpul dengan jumlah kunci minimum. Sementara itu jika dua buah simpul dalam masing-masing memiliki d kunci, maka penghapusan kunci dari salah satu simpul tersebut dapat dilakukan dengan menggabungkan kedua simpul tersebut dan menambahkan satu kunci dari simpul orang tua. Sehingga terbentuklah sebuah simpul penuh dengan $2d$ kunci.

Sebuah *B-tree* menjaga dirinya sendiri agar tetap seimbang dengan mengharuskan tiap daun untuk berada di tingkat yang sama. Tinggi *B-tree* akan bertambah seiring bertambahnya jumlah kunci. Namun, struktur B-tree membuat penambahan tingkat pada *B-tree* jarang terjadi bila dibandingkan dengan jenis pohon yang lain. *B-tree* sendiri memiliki beberapa variansi, seperti *B+ tree*, *B*-tree*, *UB-tree*, *2-3 tree*, *2-3-4 tree*, dan lain-lain.

Pada tugas ini, penulis menggunakan index cari pinjaman yang berfungsi untuk mempercepat pencarian peminjaman berdasarkan tanggal pinjam dan tanggal

kembali. Table yang digunakan ialah peminjaman. Berikut adalah implementasi indexing B-tree :

```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0172 seconds.)  
  
CREATE INDEX idx_cari_peminjaman on peminjaman(tanggal_pinjam, tanggal_kembali) USING BTREE;  
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

5.3 Hash Index

Hash Index adalah algoritma pengindeksan populer yang digunakan untuk mempercepat query. Hash Indeks menggunakan fungsi hash untuk memetakan kunci ke posisi index. Algoritma pengindeksan ini paling berguna untuk query yang persis, seperti mencari rekaman tertentu berdasarkan nilai kunci utama.

Index hash bekerja dengan memetakan setiap rekaman dalam tabel ke unique bucket berdasarkan nilai hash-nya. Nilai hash dihitung menggunakan fungsi hash, yang merupakan fungsi matematika yang mengambil item data sebagai input dan mengembalikan nilai integer unik.

Untuk menemukan rekaman dalam indeks hash, basis data menghitung nilai hash dari kunci pencarian lalu mencari bucket yang sesuai. Jika rekaman ada di bucket, basis data akan mengembalikannya. Jika tidak, basis data akan melakukan pemindaian tabel secara penuh.

Indeks hash sangat cepat untuk pencarian, tetapi tidak dapat digunakan untuk mengkueri rentang data secara efisien. Hal ini karena fungsi hash tidak mempertahankan urutan apa pun di antara rekaman dalam tabel.

Untuk melakukan kueri menggunakan indeks hash :

1. Basis data menghitung nilai hash untuk kondisi kueri.
2. Mencari hash bucket yang sesuai di tabel hash.
3. Basis data kemudian mengambil petunjuk ke baris dalam tabel yang memiliki nilai hash yang sesuai.
4. Menggunakan petunjuk tersebut untuk mengambil baris sebenarnya dari tabel.

Berikut adalah implementasi dari index hash untuk pencarian judul buku :

```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0283 seconds.)  
  
CREATE INDEX buku_judul_hash ON buku (judul);  
  
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

```
✓ Showing rows 0 - 0 (1 total, Query took 0.0013 seconds.)  
  
SELECT * FROM buku WHERE judul = 'Pemrograman Dasar';  
  
☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]
```

BAB VI

MONITORING

6.1 Pengertian Monitoring Databases

Monitoring databases adalah proses pemantauan dan analisis berkelanjutan terhadap kinerja, keamanan, dan ketersediaan basis data untuk memastikan basis data beroperasi secara efisien dan andal. Dengan memantau metrik-metrik utama seperti waktu respons query, penggunaan CPU, dan kapasitas penyimpanan, organisasi dapat secara proaktif menangani masalah, mencegah downtime, meningkatkan integritas data, dan menjaga kepatuhan terhadap standar regulasi. Alat dan praktik monitoring basis data yang efektif sangat penting untuk mengoptimalkan manajemen basis data dan mendukung kelancaran operasi.

Monitoring basis data yang efektif sangat penting untuk menjaga kinerja yang optimal, mencegah downtime, dan melindungi data sensitif. Ini merujuk pada serangkaian tugas spesifik yang dilakukan secara rutin untuk memastikan basis data berfungsi sebagaimana mestinya, memungkinkan perusahaan untuk memantau kinerja jaringan dan basis data. Ini mencakup pemeliharaan perangkat keras maupun perangkat lunak.

6.2 Manfaat Monitoring Databases

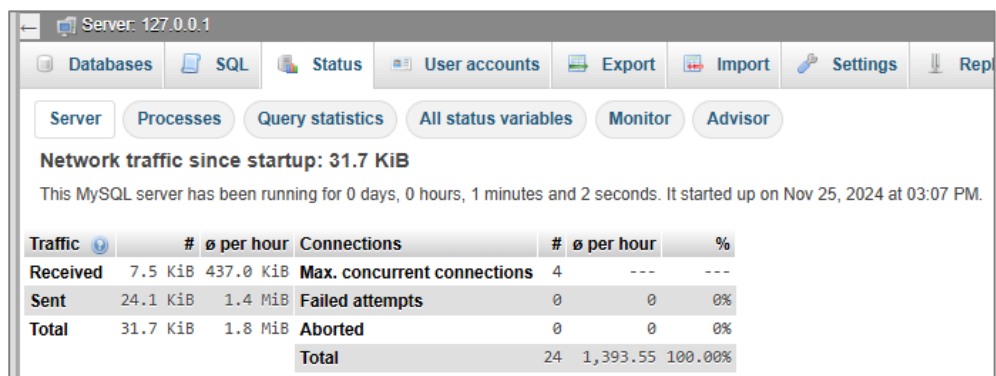
1. Menghindari kerugian pendapatan: Monitoring yang konsisten membantu mencegah downtime dan masalah kinerja yang dapat berdampak langsung pada pendapatan.
2. Melindungi data sensitif: Melindungi data berharga dari akses tidak sah dan pelanggaran memastikan kepercayaan dan kepatuhan.
3. Meningkatkan pengalaman pengguna: Kinerja basis data yang cepat dan andal menciptakan pengalaman pengguna yang lebih lancar.
4. Menyederhanakan analisis data: Mempermudah proses analisis data dan pengambilan wawasan.

5. Mengurangi biaya infrastruktur TI: Dengan mengotomasi tugas rutin, organisasi dapat menghemat upaya manual di bidang TI.
6. Meningkatkan efisiensi: Mengotomatiskan tugas-tugas yang memakan waktu, sehingga tim dapat fokus pada inisiatif strategis.
7. Memfasilitasi perbaikan berkelanjutan: Menciptakan siklus pembaruan dan peningkatan rutin untuk mengoptimalkan kesehatan basis data dari waktu ke waktu.

6.3 Langkah-langkah Monitoring Databases

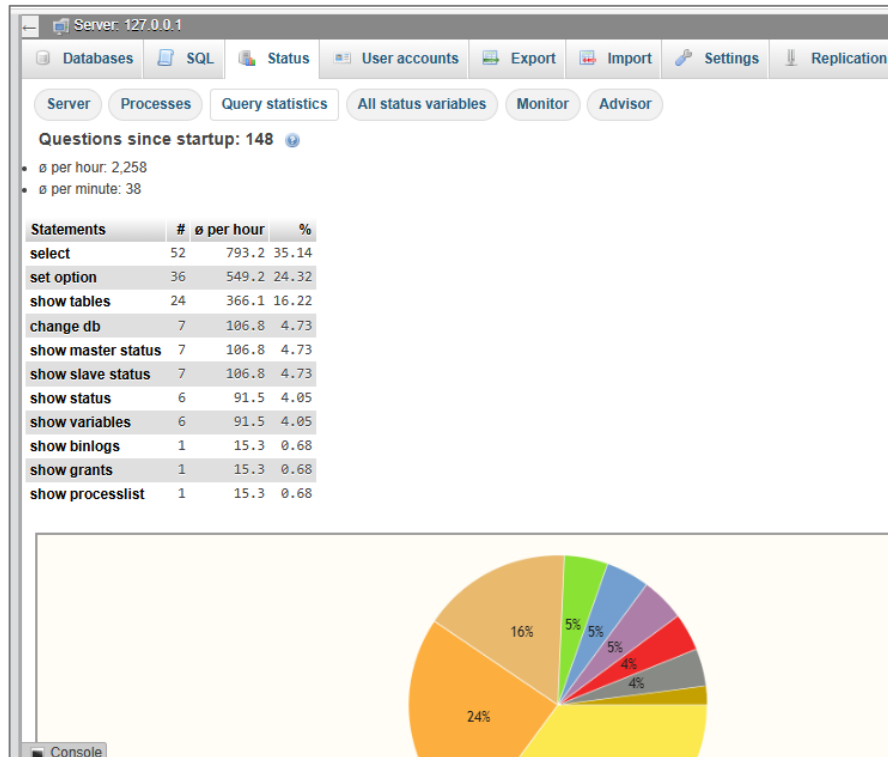
Berikut adalah implementasi monitoring database pada laporan ini :

1. Buka Phpmyadmin
2. Buka tab status

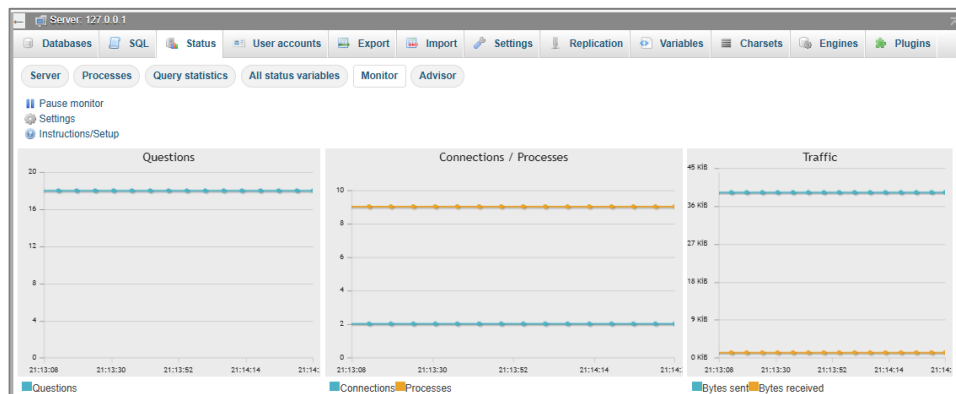


Traffic	#	Ø per hour	Connections	#	Ø per hour	%
Received	7.5 KiB	437.0 KiB	Max. concurrent connections	4	---	---
Sent	24.1 KiB	1.4 MiB	Failed attempts	0	0	0%
Total	31.7 KiB	1.8 MiB	Aborted	0	0	0%
			Total	24	1,393.55	100.00%

3. Buka query statistics untuk melihat jumlah query yang dilakukan di server



- Buka Monitor untuk melihat tabel yang paling sering diakses atau dimodifikasi.



BAB VII

REPLIKASI DATABASE

7.1 Pengertian Replikasi Databases

Replikasi database adalah sistem yang digunakan untuk menyalin, mendistribusikan data, serta melakukan sinkronisasi data antar database. Replikasi database juga dapat membantu meningkatkan kinerja database dengan mempercepat waktu akses dan meminimalkan waktu downtime yang mungkin terjadi jika hanya ada satu database tunggal. Sistem replikasi membutuhkan minimal dua buah server untuk digunakan sebagai master dan slave. Terdapat dua model replikasi database yang bisa kita lakukan, berikut keterangannya.

1. Master-Slave: Jika terjadi perubahan database pada Master, database pada Slave juga ikut berubah. Namun, jika terjadi perubahan pada Slave, Master tidak terpengaruh.
2. Master-Master: Semua server bertindak sebagai Master dan Slave. Karena jika terjadi perubahan database di salah satu server, database pada server lain juga ikut berubah.

7.2 Langkah-langkah Replikasi Databases

Pada implementasi replikasi ini, akan digunakan model replikasi Master-Slave :

Membuat User dan aksesnya melalui linux

```
mysql> create user 'repl'@'192.168.100.9' identified by 'repl';
Query OK, 0 rows affected (0.01 sec)

mysql> grant replication slave on *.* to 'repl'@'192.168.100.9';
Query OK, 0 rows affected (0.00 sec)
```

Melihat Status

```
mysql> show master status\g
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB	Executed_Gtid_Set
mysql-bin.000001	714			

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

Mengubah Master dan start slave

```
MariaDB [(none)]> CHANGE MASTER TO
-> MASTER_HOST = '192.168.72.128',
-> MASTER_LOG_FILE = 'mysql-bin.000001',
-> MASTER_LOG_POS = 174,
-> MASTER_SSL = 1;
Query OK, 0 rows affected (0.042 sec)

MariaDB [(none)]> START SLAVE;
Query OK, 0 rows affected (0.003 sec)
```

Replikasi pun berhasil dijalankan.

BAB VIII

TRANSAKSI (ACID)

8.1 Pengertian Transaksi (ACID)

Transaksi dalam basis data adalah sekumpulan operasi yang dikelompokkan bersama sehingga dieksekusi sebagai satu kesatuan. Jika salah satu operasi dalam transaksi gagal, maka semua operasi dalam transaksi tersebut harus dibatalkan (*rollback*) untuk menjaga konsistensi data. Konsep ini diatur oleh empat properti utama yang dikenal sebagai ACID:

1. Atomicity : Setiap pernyataan dalam transaksi (untuk membaca, menulis, memperbarui, atau menghapus data) harus diproses secara keseluruhan atau tidak sama sekali. Jika terjadi kegagalan di tengah proses, maka semua perubahan yang telah dilakukan oleh transaksi tersebut harus dibatalkan.
2. Consistency : Memastikan bahwa transaksi hanya membuat perubahan pada tabel dengan cara yang telah ditentukan sebelumnya dan dapat diprediksi. Consistency memastikan bahwa kerusakan atau kesalahan pada data tidak menimbulkan konsekuensi yang tidak diinginkan bagi integritas tabel.
3. Isolation : ketika beberapa pengguna membaca dan menulis dari tabel yang sama sekaligus, isolasi transaksi mereka memastikan bahwa transaksi yang bersamaan tidak saling mengganggu atau memengaruhi.
4. Durability : memastikan bahwa perubahan pada data yang dibuat oleh transaksi yang berhasil dieksekusi akan disimpan, bahkan jika terjadi kegagalan sistem.

8.2 Implementasi Transaksi (ACID)

Berikut adalah implementasi transaksi ACID pada laporan ini :

- Start Transaction, Commit, dan Rollback Statements

```
START TRANSACTION;
SELECT * from bank;
UPDATE bank SET tabungan=500000 WHERE tabungan<200000;
select * from bank;
ROLLBACK;
SELECT * from bank;
```

✓ MySQL returned an empty result set (0 rows).

```
START TRANSACTION;
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create new file \]](#)

✓ Showing rows 0 - 19 (20 rows total)

```
SELECT * from bank;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#)

Nama, nim, dan Tabungan pada table bank

nama	nim	tabungan
azka	a001	100000
azka2	a002	200000
azka3	a003	300000
azka4	a004	900000
azka5	a005	100000
azka6	a006	200000
azka7	a007	300000
azka8	a008	900000
azka9	a009	100000
azka10	a010	200000
azka11	a011	300000
azka12	a012	900000
azka13	a013	100000
azka14	a014	200000
azka15	a015	300000
azka16	a016	900000
azka17	a017	100000
azka18	a018	200000
azka19	a019	300000
azka20	a020	900000

Update tabungan

✓ 5 rows affected. (Query took 0.0006 seconds.)

```
UPDATE bank SET tabungan=500000 WHERE tabungan<200000;
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

✓ Showing rows 0 - 19 (20 rows total)

```
SELECT * from bank;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#)

Data di update

nama	nim	tabungan
azka	a001	500000
azka2	a002	200000
azka3	a003	300000
azka4	a004	900000
azka5	a005	500000
azka6	a006	200000
azka7	a007	300000
azka8	a008	900000
azka9	a009	500000
azka10	a010	200000
azka11	a011	300000
azka12	a012	900000
azka13	a013	500000
azka14	a014	200000
azka15	a015	300000
azka16	a016	900000
azka17	a017	500000
azka18	a018	200000
azka19	a019	300000
azka20	a020	900000

```
✓ MySQL returned an empty result set (0 rows)

ROLLBACK;

[ Edit inline ] [ Edit ] [ Create PHP code ]
```

```
✓ Showing rows 0 - 19 (20 rows)

SELECT * from bank;

.....

☐ Profiling [ Edit inline ] [ Edit ]
```

nama	nim	tabungan
azka	a001	100000
azka2	a002	200000
azka3	a003	300000
azka4	a004	900000
azka5	a005	100000
azka6	a006	200000
azka7	a007	300000
azka8	a008	900000
azka9	a009	100000
azka10	a010	200000
azka11	a011	300000
azka12	a012	900000
azka13	a013	100000
azka14	a014	200000
azka15	a015	300000
azka16	a016	900000
azka17	a017	100000
azka18	a018	200000
azka19	a019	300000
azka20	a020	900000

Data kembali ke keadaan sebelumnya dengan menggunakan rollback

- Statements yang tidak dapat di Rolled Back

```

1 START TRANSACTION;
2
3 CREATE TABLE bank2 (
4     id INT AUTO_INCREMENT PRIMARY KEY,
5     description VARCHAR(255)
6 );
7
8 ROLLBACK;

```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

```
START TRANSACTION;
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0007 seconds.)

```
CREATE TABLE bank2 ( id INT AUTO_INCREMENT PRIMARY KEY, description VARCHAR(255) );
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)

```
ROLLBACK;
```

[Edit inline] [Edit] [Create PHP code]



Meskipun sudah di rollback, namun table bank2 tetap ada

- Savepoint, Rollback to Savepoint, dan Release Savepoint Statements

```

START TRANSACTION;

INSERT INTO bank VALUES ('azka21', 'a021', 100000);
SAVEPOINT sp1;
INSERT INTO bank VALUES ('azka22', 'a022', 300000);
SELECT * from bank;

ROLLBACK TO SAVEPOINT sp1;
SELECT * from bank;

```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)

```
START TRANSACTION;
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted. (Query took 0.0080 seconds.)

```
INSERT INTO bank VALUES ('azka21', 'a021', 100000);
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
SAVEPOINT sp1;
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted. (Query took 0.0004 seconds.)

```
INSERT INTO bank VALUES ('azka22', 'a022', 300000);
```

[Edit inline] [Edit] [Create PHP code]

The screenshot shows the SQLyog interface after executing the command `ROLLBACK TO SAVEPOINT sp1;`. The status bar at the top indicates "MySQL returned an empty result set". The query window displays the executed command and the results of `SELECT * from bank;`, which shows 21 rows. The Profiling tab is also visible.

- Set Transaction Statement

```
SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE;  
SET SESSION TRANSACTION READ ONLY;
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query too slow to log)

```
SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE;
```

[Edit inline] [Edit] [Create PHP code]

✔ MySQL returned an empty result set (i.e. zero rows). (Query too slow to log)

```
SET SESSION TRANSACTION READ ONLY;
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

DAFTAR PUSTAKA

Oracle Corporation. (n.d.). *MySQL Documentation*. Diakses pada 18 November 2024, dari <https://dev.mysql.com/doc/>

Oracle Corporation. (n.d.). *Oracle Database Documentation*. Diakses pada 18 November 2024, dari <https://www.oracle.com/id/database/technologies/oracle-database-software-downloads.html>

Hipp, W. C. (n.d.). *SQLite Documentation*. Diakses pada 18 November 2024, dari <https://www.sqlite.org/docs.html>

The PostgreSQL Global Development Group. (n.d.). *PostgreSQL Documentation*. Diakses pada 18 November 2024, dari <https://www.postgresql.org/docs/>

MongoDB, Inc. (n.d.). *MongoDB Documentation*. Diakses pada 18 November 2024, dari <https://www.mongodb.com/docs/>

Apache Friends. (n.d.). *Download XAMPP*. Diakses pada 18 November 2024, dari <https://www.apachefriends.org/download.html>

Dirgantara, M. R., Syahputri, S., Hasibuan, A., & Nurbaiti. (2023). Pengenalan Database Management System (DBMS). *Madani: Jurnal Ilmiah Multidisiplin*, 1(6), 300–306.

Hafiz. (2022). Pentingnya Sistem Keamanan Database untuk Melindungi Data Pribadi. *JISKA: Jurnal Sistem Informasi dan Informatika*, 1(2), 44–47.

Munir, R. (2006). *Kriptografi*. Bandung: Penerbit Informatika.

Putra, A. E. (2009). Fungsi Hash pada Kriptografi. *Makalah IF209*.

Wibisono, P. (2011). Cara Kerja B-Tree dan Aplikasinya. *Makalah IF2091*.

DOT Indonesia. (2018). *Database Optimization #1: Indexing*. Diakses pada 18 November 2024, dari <https://blog.dot.co.id/database-optimization-1-indexing-649e7837a22a>.

Puskomedia. (n.d.). Cara Membuat Replikasi Database MariaDB/MySQL. Diakses pada 18 November 2024, dari <https://www.puskomedia.id/blog/cara-membuat-replikasi-database-mariadbmysql>.

Oracle Corporation. (n.d.). SQL Statements for Transaction Management. *MySQL 8.4 Reference Manual*. Diakses pada 18 November 2024, dari <https://dev.mysql.com/doc/refman/8.4/en/sql-transactional-statements.html>.

LogicMonitor. (n.d.). What Is Database Monitoring and Why Is It Still Important? Diakses pada 25 November 2024, dari <https://www.logicmonitor.com/blog/what-is-database-monitoring-and-why-is-it-still-important>.

Oracle Corporation. (n.d.). Password Hashing. *MySQL 5.7 Reference Manual*. Diakses pada 18 November 2024, dari <https://dev.mysql.com/doc/mysql-security-excerpt/5.7/en/password-hashing.html>.