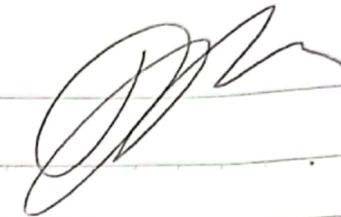


Ahmad Azka Al Husaini
09021282227096

Date

95



1. Pasifkan print screen seurai dengan program yang digunakan.
2. Gunakan link ke website original
3. Tambahkan di daftar pustaka website trip[?] database
4. Primary key tidak perlu index

DOKUMENTASI TUGAS BESAR
PERANCANGAN BASIS DATA
SEMESTER GANJIL 2023/2024

**PERANCANGAN SISTEM PERPUSTAKAAN
FAKULTAS ILMU KOMPUTER UNIVERSITAS
SRIWIJAYA BERBASIS WEBSITE**



Oleh :

Ahmad Azka Al Husaini

NIM : 09021282227096

TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
2024

ABSTRAK

Perpustakaan Fakultas Ilmu Komputer Universitas Sriwijaya berperan penting dalam mendukung kegiatan akademik dan penelitian bagi mahasiswa dan dosen. Seiring dengan meningkatnya kebutuhan akan akses informasi yang cepat dan fleksibel, pengelolaan perpustakaan yang selama ini dilakukan secara manual dirasakan kurang memadai. Proses pencatatan dan peminjaman buku yang berbasis kertas kerap menimbulkan kendala, seperti kesulitan dalam mencari buku, keterlambatan pembaruan data ketersediaan, serta minimnya aksesibilitas layanan perpustakaan secara online.

Proyek ini bertujuan untuk merancang dan mengembangkan sistem perpustakaan berbasis website yang memungkinkan pengguna (mahasiswa dan dosen) untuk melakukan pencarian, peminjaman, dan pengembalian buku secara online. Sistem ini diharapkan mampu meningkatkan efisiensi pengelolaan perpustakaan, mempermudah akses informasi, dan memberikan pengalaman yang lebih modern bagi pengguna. Dengan dukungan teknologi web dan basis data yang dikelola secara terpusat, admin dan super admin dapat mengelola data buku dan statistik peminjaman serta memastikan akurasi informasi ketersediaan buku.

Sistem perpustakaan berbasis website ini memiliki tiga kategori pengguna: admin, super admin, dan user, dengan hak akses yang berbeda sesuai dengan peran masing-masing. Admin bertanggung jawab atas pengelolaan penuh sistem, super user memiliki akses manajemen data tanpa hak hapus, dan user dapat mengakses informasi dasar perpustakaan. Dengan penerapan sistem ini, diharapkan proses pengelolaan perpustakaan fakultas Ilmu Komputer Universitas Sriwijaya dapat berjalan lebih efektif, efisien, dan sejalan dengan visi fakultas untuk unggul dalam bidang teknologi informasi pada tahun 2025.

Kata Kunci : Perancangan, Sistem Perpustakaan, *Website, Database*

DAFTAR ISI

ABSTRAK.....	ii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Informasi Perpustakaan.....	2
1.3 Tujuan.....	3
1.4 Rumusan Masalah	4
1.5 Batasan Masalah	4
BAB II.....	6
TINJAUAN PUSTAKA.....	6
2.1 Landasan teori.....	6
2.1.1 Transaksi ACID	6
2.1.2 Indexing.....	6
2.1.3 B-Tree	7
2.2 Identifikasi Pengguna	8
2.2.1 Kategori Pengguna.....	8
BAB III	10
PERANCANGAN SISTEM	10
3.1 Perancangan Arsitektur Hardware.....	10
3.2 Perancangan Basis Data	10
3.2.1 Database Management	10
3.2.2 Diagram ER.....	11
3.2.3 Tabel Relasi	11
3.2.4 Jumlah Kolom dan baris Tabel.....	11
3.2.5 Ukuran Tabel.....	12
3.3 Perancangan Perangkat Lunak	14
3.3.1 Bahasa Pemrograman	14
3.4 Perancangan Security	14
3.4.1 Hashing	14
3.4.2 Kontrol Akses Berdasarkan Peran.....	15
3.4.2.1 Graph	15
3.4.2.2 User Access Matrix.....	15
3.3.3 Replikasi Database	16
BAB IV	17

IMPLEMENTASI SISTEM	17
4.1 Query	17
4.2 User	23
4.3 Wireframe	24
BAB V	27
ANALISIS	27
5.1 Query Processing	27
5.2 Ukuran Data & Blok yang Dipakai	34
5.3 Indexing B-Tree.....	36
5.4 Durability.....	36
5.4.1 Replikasi Database	36
5.5 Recovery Plan.....	38
5.5.1 Recovery Sebelum Direset	38
5.5.2 Pencegahan Crash	38
5.6 Monitoring Databases	39
5.7 Maintenance	41
5.8 Evaluasi Performa Sistem	42
DAFTAR PUSTAKA.....	44

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era digital saat ini, kebutuhan akan akses informasi yang cepat dan mudah semakin meningkat, terutama di lingkungan akademik yang terus berkembang. Perpustakaan menjadi salah satu fasilitas penting di lingkungan akademik yang mendukung kegiatan belajar-mengajar, penelitian, serta pengembangan ilmu pengetahuan. Berbagai institusi pendidikan perlu beradaptasi dengan mengoptimalkan teknologi informasi.

Fakultas Ilmu Komputer sebagai salah satu fakultas yang ada di Universitas Sriwijaya memiliki visi menjadi institusi yang unggul di bidang ICT pada tahun 2025 dan misi menyelenggarakan dan mengembangkan pendidikan tinggi dalam upaya menghasilkan manusia terdidik yang dapat menerapkan dan mengembangkan ilmu pengetahuan bidang informasi, komunikasi dan teknologi. Fakultas ini juga dilengkapi dengan fasilitas perpustakaan, Namun, layanan perpustakaan yang selama ini digunakan pada fakultas ini bersifat manual. pencatatan, peminjaman, dan pengembalian buku secara fisik dilakukan dalam bentuk fisik melalui kertas.

Perpustakaan fakultas Ilmu Komputer Universitas Sriwijaya dirasa kurang memadai untuk memenuhi kebutuhan user (mahasiswa dan dosen) yang terus berkembang. Berbagai kendala sering kali ditimbulkan, seperti kesulitan dalam pencarian buku, keterlambatan dalam memperbarui data ketersediaan buku, serta minimnya aksesibilitas layanan perpustakaan secara online. Hal ini mengakibatkan anggota perpustakaan harus datang langsung ke lokasi untuk mendapatkan informasi atau layanan tertentu, yang tentunya mengurangi efektivitas penggunaan waktu dan tenaga. Dengan pengembangan sistem perpustakaan berbasis website, sistem ini memungkinkan mahasiswa dan dosen untuk mengakses layanan perpustakaan secara online, seperti mencari buku yang tersedia, meminjam, dan mengembalikan buku dengan lebih efisien. Selain itu, sistem ini akan membantu

admin dan super user dalam mengelola data buku, anggota, serta melakukan pemantauan dan pelaporan statistik peminjaman dan pengembalian secara lebih efektif. Sistem ini juga memberikan keuntungan dari sisi efisiensi dan penghematan waktu bagi pengguna dan pengelola perpustakaan. Oleh karena itu, diperlukan suatu sistem yang mampu mengotomatisasi proses manajemen perpustakaan dan memberikan akses lebih luas kepada pengguna melalui teknologi berbasis website.

1.2 Informasi Perpustakaan

Pada tahun 2024, Jumlah rata-rata peminjaman dan pengembalian per-harinya pada perpustakaan fakultas Ilmu Komputer Universitas Sriwijaya ialah 50 kali, dan dalam 3 tahun ke depan, perkiraan total peminjamannya sebanyak 54.750. Dengan adanya sistem ini, admin dan super user dapat mengelola transaksi peminjaman dan pengembalian lebih efisien, serta mengurangi waktu yang dibutuhkan untuk melayani peminjaman dan pengembalian buku secara manual. Admin dan super user juga dapat memperbarui data buku dan anggota secara real-time tanpa harus melakukan pekerjaan manual yang memakan waktu.

Jumlah buku di perpustakaan Fakultas Ilmu Komputer ialah 30.000, jumlah rata-rata peminjaman dan pengembalian buku per-harinya ialah 50 kali, dan dalam 3 tahun ke depan, perkiraan total peminjamannya sebanyak 54.750. jumlah kategori buku ialah 50, dan jumlah penerbit ialah 700.

Perancangan sistem ini juga menambahkan keuntungan dibandingkan sistem manual. Keuntungan sistem diperoleh melalui denda keterlambatan pengembalian buku, diperkirakan total denda sekitar Rp 500.000 per bulan, serta diperoleh dari penghematan biaya tenaga kerja dari yang sebelumnya Rp 3.500.000 dengan biaya tenaga kerja Rp 10.000 per jam dan 350 jam kerja per bulan, menjadi Rp 750.000 dengan biaya tenaga kerja Rp 10.000 per jam dan 75 jam kerja per bulan. Pengeluaran per bulan sistem ini (biaya listrik, pemeliharaan sistem, dll) diperkirakan sekitar Rp 1.000.000 per bulan. Keuntungan bersih dari sistem ini yang didapat dari pendapatan dikurang pengeluaran mencapai Rp 2.050.000 per bulan.

Dalam sistem perpustakaan fakultas Ilmu Komputer Universitas Sriwijaya, terdiri dari tiga kategori pengguna : admin, super user, dan user dengan hak akses yang berbeda sesuai dengan peran masing-masing. Admin ialah dekan, wakil dekan, dan kepala perpustakaan fakultas Ilmu Komputer Universitas Sriwijaya yang bertanggung jawab atas pengelolaan sistem secara keseluruhan. Admin memiliki akses penuh untuk mengelola data, termasuk membaca, menambahkan, menghapus, memperbarui data buku, peminjaman, pengembalian, data user / anggota (mahasiswa, dan dosen). Admin juga memastikan bahwa semua informasi di perpustakaan terupdate dan tersedia secara real-time. Super user terdiri dari 2 IT support dan 3 manajer perpustakaan pada Fakultas Ilmu Komputer Universitas Sriwijaya, dengan total 5 orang yang memiliki peran yang sama seperti admin, namun mereka tidak memiliki hak untuk menghapus data. Super user juga bertugas untuk melakukan audit, memastikan integritas sistem, dan melakukan pemeliharaan rutin seperti backup data. User atau anggota ialah mahasiswa dari berbagai jurusan di Fakultas Ilmu Komputer yang masih aktif, terdiri dari mahasiswa semester 1 hingga mahasiswa semester lanjut yang belum lulus, dengan total 2.500 mahasiswa, serta 78 dosen. Melalui sistem perpustakaan berbasis website, diharapkan seluruh proses pengelolaan perpustakaan dapat berjalan lebih efektif, efisien, dan modern sesuai dengan perkembangan teknologi saat ini, sehingga dapat mendukung kegiatan akademik secara optimal.

1.3 Tujuan

Adapun tujuan dari perancangan sistem perpustakaan berbasis website ini adalah sebagai berikut:

1. Mempermudah mahasiswa dan dosen dalam mencari, melihat peminjaman dan pengembalian buku secara online tanpa harus datang langsung ke perpustakaan.
2. Mempermudah admin dalam mencari, menambahkan, menghapus, memperbarui data buku, peminjaman, pengembalian, data pengguna, serta memastikan bahwa semua informasi di perpustakaan terupdate dan tersedia secara real-time.
3. Mempermudah super user dalam mencari, menambahkan memperbarui data

- buku, peminjaman, pengembalian, data pengguna, serta melakukan audit, memastikan integritas sistem, dan melakukan pemeliharaan rutin seperti backup data.
4. Meningkatkan aksesibilitas layanan perpustakaan dengan menyediakan informasi yang selalu diperbarui dan dapat diakses kapan saja melalui internet.
 5. Mengurangi potensi kesalahan manual bagi admin dan super user dalam mengelola data perpustakaan, sehingga meminimalkan ketidakakuratan informasi terkait ketersediaan buku.

1.4 Rumusan Masalah

Adapun rumusan masalah dari perancangan sistem perpustakaan berbasis website ini adalah sebagai berikut:

1. Bagaimana cara merancang dan mengimplementasikan sistem perpustakaan fakultas Ilmu Komputer Universitas Sriwijaya dengan baik hingga tiga tahun ke depan?
2. Bagaimana merancang dan mengelola basis data sistem perpustakaan fakultas Ilmu Komputer Universitas Sriwijaya dengan efisien dan skalibilitas hingga tiga tahun ke depan?
3. Bagaimana mendesain perangkat keras, jaringan yang handal untuk menunjang operasional, algoritma, dan query yang optimal untuk sistem perpustakaan fakultas Ilmu Komputer Universitas Sriwijaya hingga tiga tahun ke depan?
4. Bagaimana memastikan keandalan data, pencegahan, pemantauan, dan strategi pemulihan yang efektif untuk menghindari kerusakan atau kehilangan data?
5. Bagaimana memastikan keamanan data pengguna dan transaksi di dalam sistem perpustakaan berbasis website ini?

1.5 Batasan Masalah

Batasan masalah dari perancangan sistem perpustakaan berbasis website ini adalah sebagai berikut:

1. Sistem perpustakaan ini dibangun berbasis website dengan menggunakan database MySQL dan bahasa pemrograman PHP.
2. Pengunjung harus mendaftar ke perpustakaan terlebih dahulu ke perpustakaan fakultas Ilmu Komputer untuk menjadi anggota perpustakaan sehingga dapat meminjam buku. Anggota perpustakaan hanya dapat melakukan peminjaman buku ketika berada di perpustakaan Fakultas Ilmu Komputer.
3. Sistem ini memiliki tiga kategori pengguna: admin, super user, dan user, dengan hak akses yang berbeda sesuai dengan peran masing-masing.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan teori

2.1.1 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 tipe.
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.

2.1.2 Indexing

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 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*.

2.1.3 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.

2.2 Identifikasi Pengguna

2.2.1 Kategori Pengguna

Dalam sistem perpustakaan berbasis website yang dirancang, terdapat tiga kategori utama pengguna yang memiliki peran dan hak akses berbeda, yaitu Admin, Super User, dan User. Setiap kategori memiliki hak akses dan tanggung jawab yang berbeda sesuai dengan kebutuhan operasional sistem.

1. Admin

Admin merupakan pengguna yang bertanggung jawab atas pengelolaan sistem secara keseluruhan. Pada implementasi sistem ini, peran Admin dipegang oleh pihak dengan otoritas lebih tinggi dalam lingkungan fakultas, diantaranya ialah dekan, wakil dekan, dan kepala perpustakaan fakultas Ilmu Komputer Universitas Sriwijaya. Admin memiliki akses penuh untuk mengelola data, termasuk membaca,

menambahkan, menghapus, memperbarui data buku, peminjaman, pengembalian, dan data pengguna (mahasiswa, dan dosen). Selain itu, admin juga memastikan bahwa semua informasi di perpustakaan terupdate dan tersedia secara real-time. Admin juga bertugas memonitor aktivitas pengguna dan menyelesaikan permasalahan yang mungkin terjadi selama penggunaan sistem.

2. Super User

Super User merupakan pengguna dengan hak akses yang lebih tinggi dibandingkan User. Pada implementasi sistem ini, peran Super User ialah sebagai staf yang berhubungan dengan pengelolaan teknis dan strategis perpustakaan, diantaranya ialah IT support atau manajer perpustakaan pada Fakultas Ilmu Komputer Universitas Sriwijaya. Super User memiliki akses untuk mengelola data, termasuk membaca, menambahkan, memperbarui data buku, peminjaman, pengembalian, dan data pengguna (mahasiswa dan dosen), namun tidak memiliki hak untuk menghapus data. Selain itu, Super User bertugas untuk melakukan audit, memastikan integritas sistem, dan melakukan pemeliharaan rutin seperti backup data.

3. User

User merupakan pengguna umum yang hanya memiliki hak untuk membaca informasi yang tersedia di sistem. Pada implementasi sistem ini, peran user ialah sebagai mahasiswa atau dosen pada Fakultas Ilmu Komputer Universitas Sriwijaya yang menggunakan fasilitas perpustakaan. User memiliki akses untuk melihat kategori buku, ketersediaan buku, riwayat peminjaman dan pengembalian, serta data pribadi user, tetapi tidak memiliki akses untuk membuat, memperbarui, atau menghapus data apa pun.

BAB III

PERANCANGAN SISTEM

3.1 Perancangan Arsitektur Hardware

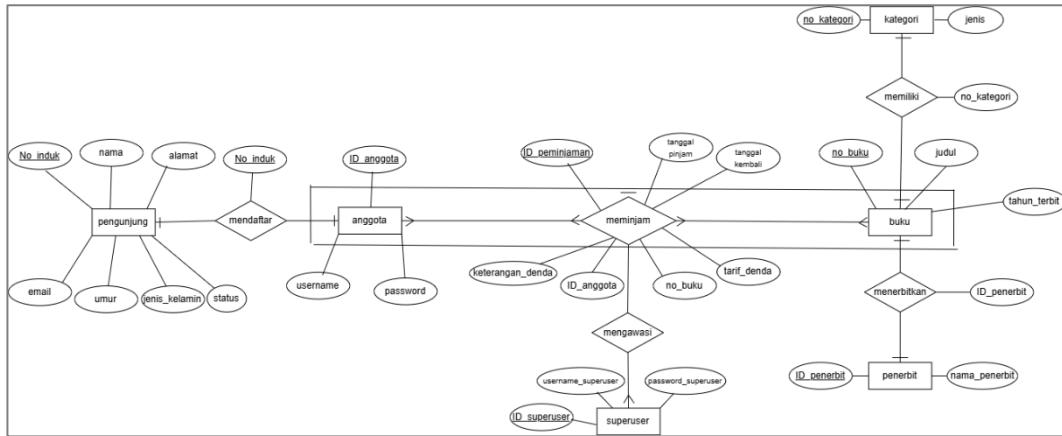
Komponen	Nama Produk	Jumlah Unit	Total Harga per Unit
Server	Dell PowerEdge T150	1	20.600.000
NAS (Network Attached Storage)	Synology DiskStation DS920+	1	8.500.000
Hard Drive NAS	Seagate IronWolf 4TB (RAID 5 Setup)	3	5.400.000
Sistem Operasi	Ubuntu Server 22.04 LTS	1	-
RAM	Crucial 32GB DDR4 RAM	1	2.800.000
Router	TP-Link Archer C7 Gigabit Router	1	1.500.000
Ethernet Switch	Netgear 8-Port Gigabit Ethernet Switch	1	600.000
UPS (Cadangan Daya)	APC UPS 1200VA	1	3.000.000
Cadangan Daya	Genset Honda EU22i 2200 Watt Silent	1	9.500.000
Power Supply	Corsair 450Watt PSU	1	600.000

3.2 Perancangan Basis Data

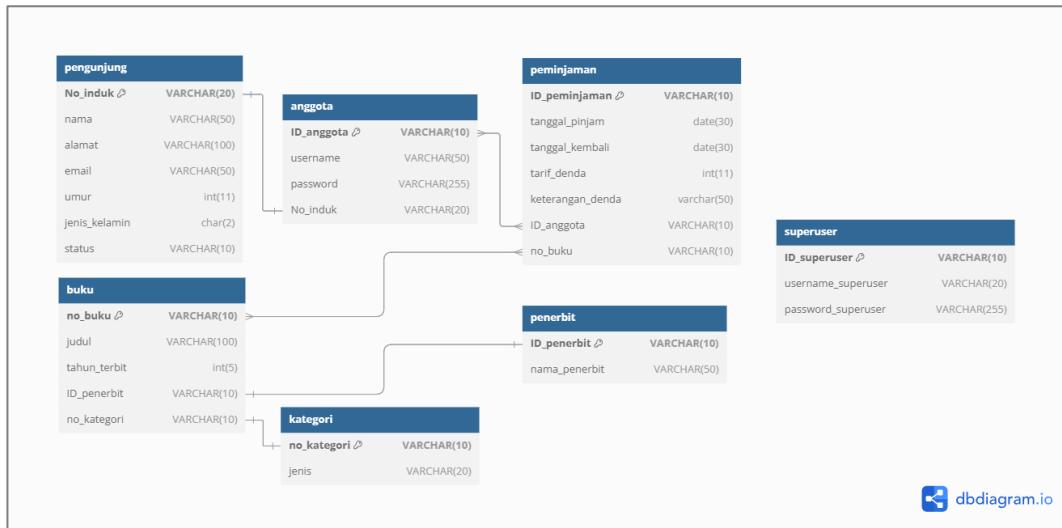
3.2.1 Database Management

Sistem basis data yang digunakan untuk menyimpan informasi pengguna, buku, peminjaman, dan pengembalian ialah **MySQL**.

3.2.2 Diagram ER



3.2.3 Tabel Relasi



3.2.4 Jumlah Kolom dan baris Tabel

Tabel-tabel dibawah ini dirancang berdasarkan jumlah rata-rata data yang mencerminkan keadaan operasional perpustakaan Fakultas Ilmu Komputer Universitas Sriwijaya setelah 3 tahun. Jumlah anggota perpustakaan atau user terdiri dari mahasiswa dari berbagai jurusan di Fakultas Ilmu Komputer yang masih aktif, terdiri dari mahasiswa semester 1 hingga mahasiswa semester lanjut yang belum lulus, dengan total 2.500 mahasiswa, serta 78 dosen, jumlah super user terdiri dari 2 IT support dan 3 manajer perpustakaan pada Fakultas Ilmu Komputer Universitas Sriwijaya, dengan total 5 orang. Jumlah buku di perpustakaan Fakultas Ilmu Komputer ialah 30.000, jumlah rata-rata peminjaman dan pengembalian buku

per-harinya ialah 50 kali, dan dalam 3 tahun ke depan, perkiraan total peminjamannya sebanyak 54.750. jumlah kategori buku ialah 50, dan jumlah penerbit ialah 700.

No	Nama Tabel	Jumlah Kolom	Jumlah Data
1	Pengunjung	6	2578
2	Anggota	9	2578
3	Buku	5	30000
4	Kategori	2	50
5	Peminjaman	7	54750
6	Penerbit	2	700
7	Superuser	4	5

3.2.5 Ukuran Tabel

Ukuran tabel merupakan estimasi kapasitas penyimpanan yang dibutuhkan untuk menyimpan data dalam tabel, termasuk ukuran setiap atribut, byte overhead, dan jumlah record dalam tabel. Perhitungan ini penting untuk memastikan efisiensi dalam pengelolaan basis data, terutama dalam hal optimasi performa, alokasi penyimpanan, dan perencanaan skalabilitas untuk kebutuhan data di masa depan.

1. Pengunjung

Atribut	Tipe Data	Ukuran (bytes)	Keterangan
No_induk	Varchar(20)	22	20 karakter + 2 byte overhead
Nama	Varchar(50)	52	50 karakter + 2 byte overhead
Alamat	Varchar(100)	102	100 karakter + 2 byte overhead
Email	Varchar(50)	52	50 karakter + 2 byte overhead
Umur	Int(3)	3	
Jenis_kelamin	Char(2)	2	
Status	Varchar(10)	12	10 karakter + 2 byte overhead

2. Anggota

Atribut	Tipe Data	Ukuran (bytes)	Keterangan
ID_anggota	Varchar(10)	12	10 karakter + 2 byte overhead
Username	Varchar(50)	52	50 karakter + 2 byte overhead
Password	Varchar(255)	257	255 karakter + 2 byte overhead
No_induk	Varchar(20)	22	20 karakter + 2 byte overhead

3. Buku

Atribut	Tipe Data	Ukuran (bytes)	Keterangan
No_buku	Varchar(10)	12	10 karakter + 2 byte overhead
Judul	Varchar(100)	102	100 karakter + 2 byte overhead
Tahun_terbit	Int(5)	5	
ID_penerbit	Varchar(10)	12	10 karakter + 2 byte overhead
No_kategori	Varchar(10)	12	10 karakter + 2 byte overhead

4. Kategori

Atribut	Tipe Data	Ukuran (bytes)	Keterangan
No_kategori	Varchar(10)	12	10 karakter + 2 byte overhead
Jenis	Varchar(20)	22	20 karakter + 2 byte overhead

5. Peminjaman

Atribut	Tipe Data	Ukuran (bytes)	Keterangan
ID_peminjaman	Varchar(10)	12	10 karakter + 2 byte overhead
Tanggal_pinjam	Date(30)	30	
Tanggal_kembali	Date(30)	30	
Tarif_denda	Int(11)	11	
Keterangan_denda	Varchar(50)	52	50 karakter + 2 byte overhead
ID_anggota	Varchar(10)	12	10 karakter + 2 byte overhead
NO_buku	Varchar(10)	12	10 karakter + 2 byte overhead

6. Penerbit

Atribut	Tipe Data	Ukuran (bytes)	Keterangan
ID_penerbit	Varchar(10)	12	10 karakter + 2 byte overhead
Nama_penerbit	Varchar(50)	52	50 karakter + 2 byte overhead

7. Superuser

Atribut	Tipe Data	Ukuran (bytes)	Keterangan
ID_superuser	Varchar(10)	12	10 karakter + 2 byte overhead
Username_superuser	Varchar(20)	22	20 karakter + 2 byte overhead
Password_superuser	Varchar(255)	257	255 karakter + 2 byte overhead

3.3 Perancangan Perangkat Lunak

3.3.1 Bahasa Pemrograman

Bahasa pemrograman yang digunakan untuk mengembangkan tampilan ialah HTML, CSS, JavaScript, dan framework Bootstrap 5. Sedangkan bahasa pemrograman yang digunakan untuk menangani proses server-side, berkomunikasi dengan database, dan memastikan keamanan serta kelancaran alur data ialah PHP.

3.4 Perancangan Security

Dalam merancang sistem, penerapan yang telah dilakukan ialah :

3.4.1 Hashing

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. Fungsi ini juga memiliki berbagai algoritma, seperti MD4, SHA, RIMPEND, WHIRPOL, dan TIGER.

Dalam merancang sistem ini, Algoritma hashing yang digunakan ialah SHA-2, karena algoritma ini dianggap sebagai algoritma hash yang paling aman dari serangan umum seperti brute force yang dapat memakan waktu bertahun-tahun untuk memecahkan intisari hash. SHA-2 juga lebih cepat dari BCrypt dan implementasinya lebih murah. Fungsi hashing dalam sistem perpusakaan ini digunakan pada kolom password di table super user, dan user / anggota. Hal ini digunakan saat sesi login, ketika membutuhkan kolom password, sehingga dapat memperkuat keamanan sistem.

Berikut adalah implementasi hashing :

```
INSERT into superuser values
('SU001', 'Ahmadfariz', SHA2('fariz123', 256));|
```

```

SELECT * FROM `superuser` WHERE 1;

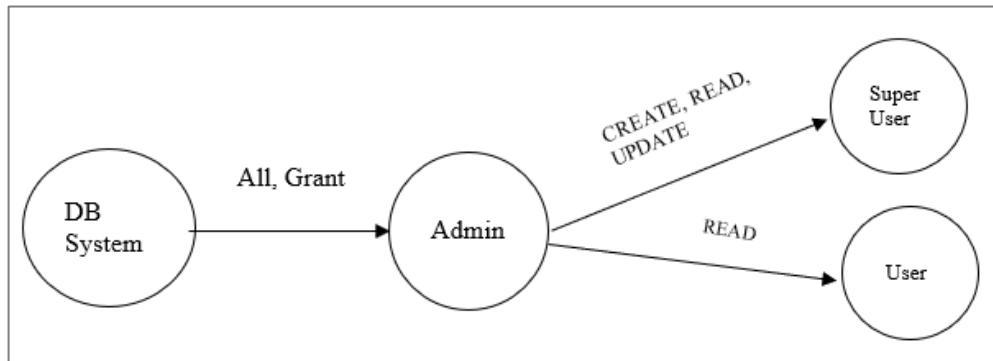
```

ID_superuser	username_superuser	password_superuser
SU001	Ahmadfariz	8a221bb139510e5cbdaca09b82930346d6b27748387ab10ba9...

3.4.2 Kontrol Akses Berdasarkan Peran

Sistem menerapkan kontrol akses berbasis peran untuk membatasi akses pengguna ke fitur tertentu. Admin memiliki akses penuh, Super User dan User diberikan hak akses terbatas sesuai peran yang telah dibahas pada bagian user access matrix. Hal ini mencegah pengguna dari mengakses data yang tidak diperlukan atau berpotensi disalahgunakan.

3.4.2.1 Graph



3.4.2.2 User Access Matrix

	Super User	Anggota	Pengunjung	Kategori	Buku	Peminjaman	Penerbit
Admin	CREATE, READ, UPDATE, DELETE	CREATE, READ, UPDATE, DELETE	CREATE, READ, UPDATE, DELETE	CREATE, READ, UPDATE, DELETE	CREATE, READ, UPDATE, DELETE	CREATE, READ, UPDATE, DELETE	CREATE, READ, UPDATE, DELETE
Super User	READ (id_superuser, username_ superuser)	CREATE, READ, UPDATE	CREATE, READ, UPDATE	CREATE, READ, UPDATE	CREATE, READ, UPDATE	CREATE, READ, UPDATE	CREATE, READ, UPDATE
User	-	READ (id_anggota, username, no_induk)	READ (nama, umur, jenis_kelamin, no_induk, status)	READ	READ	READ	READ

1. Admin

Admin diberikan akses penuh (Create, Read, Update, Delete) untuk semua tabel Anggota, Pengunjung, Kategori, Buku, Peminjaman, dan Penerbit. Admin juga diberikan akses grant.

2. Super User

Super User diberikan akses (Create, Read, Update) untuk tabel Anggota, Pengunjung, Kategori, Buku, Peminjaman, dan Penerbit. Super User hanya dapat melihat id_superuser dan username_superuser pada table Super User. Super User tidak dapat mengakses selain (Read) pada tabel super user dikarenakan role admin lebih tinggi.

3. User

User diberikan akses (Read) untuk tabel Kategori, Buku, Peminjaman, dan Penerbit. User juga dapat melihat tabel anggota dan pengunjung, namun tidak dapat melihat kolom seperti email, alamat, password dikarenakan data sensitif. User tidak dapat mengakses tabel super user dikarenakan role super user lebih tinggi.

3.3.3 Replikasi 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. Untuk implementasinya akan dibahas pada bab 5.

BAB IV

IMPLEMENTASI SISTEM

4.1 Query

A. Data Definition Language (DDL)

1. Database Perpustakaan

```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0029 seconds.)  
create database perpustakaan;  
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

2. Table Pengunjung

```
CREATE TABLE pengunjung (  
    No_induk VARCHAR(20) PRIMARY KEY,  
    nama VARCHAR(50),  
    alamat VARCHAR(100),  
    email VARCHAR(50),  
    umur INT(11),  
    jenis_kelamin CHAR(2),  
    status VARCHAR(10)  
) ;
```

3. Table Anggota (User)

```
CREATE TABLE anggota (  
    ID_anggota VARCHAR(10) PRIMARY KEY,  
    username VARCHAR(50),  
    password VARCHAR(255),  
    No_induk VARCHAR(20),  
    FOREIGN KEY (No_induk) REFERENCES pengunjung(No_induk)  
) ;
```

4. Table Super User

```
CREATE TABLE superuser (
    ID_superuser VARCHAR(10) PRIMARY KEY,
    username_superuser VARCHAR(20),
    password_superuser VARCHAR(255)
);
```

5. Table Kategori

```
CREATE TABLE kategori (
    no_kategori VARCHAR(10) PRIMARY KEY,
    jenis VARCHAR(20)
);
```

6. Table Penerbit

```
CREATE TABLE penerbit (
    ID_penerbit VARCHAR(10) PRIMARY KEY,
    nama_penerbit VARCHAR(50)
);
```

7. Table Buku

```
CREATE TABLE buku (
    no_buku VARCHAR(10) PRIMARY KEY,
    judul VARCHAR(100),
    tahun_terbit INT(5),
    ID_penerbit VARCHAR(10),
    no_kategori VARCHAR(10),
    FOREIGN KEY (ID_penerbit) REFERENCES penerbit(ID_penerbit),
    FOREIGN KEY (no_kategori) REFERENCES kategori(no_kategori)
);
```

8. Table Peminjaman

```
CREATE TABLE peminjaman (
    ID_peminjaman VARCHAR(10) PRIMARY KEY,
    tanggal_pinjam DATE,
    tanggal_kembali DATE,
    tarif_denda INT(11),
    keterangan_denda VARCHAR(50),
    ID_anggota VARCHAR(10),
    no_buku VARCHAR(10),
    FOREIGN KEY (ID_anggota) REFERENCES anggota(ID_anggota),
    FOREIGN KEY (no_buku) REFERENCES buku(no_buku)
);
```

B. Data Manipulation Language (DML)

1. Table Pengunjung

The screenshot shows the MySQL Workbench interface with the following details:

- Query results header: "Showing rows 0 - 10 (11 total, Query took 0.0021 seconds.)"
- SQL query: "select * from pengunjung;"
- Table structure:

	No_inuk	nama	alamat	email	umur	jenis_kelamin	status
<input type="checkbox"/>	202001001	Lina Damayanti	Jl. Kenanga 23	lina.damayanti@email.com	22	P	Mahasiswa
<input type="checkbox"/>	202001002	Agus Putra	Jl. Demang 12	agus.putra@email.com	22	L	Mahasiswa
<input type="checkbox"/>	202001003	Andi Setiawan	Jl. Merdeka No. 123	andi.setiawan@email.com	22	L	Mahasiswa
<input type="checkbox"/>	202001004	Rini Sari	Jl. Harapan Indah 45	rini.sari@email.com	22	P	Mahasiswa
<input type="checkbox"/>	202001005	Budi Pratama	Jl. Pahlawan 67	budi.pratama@email.com	23	L	Mahasiswa
<input type="checkbox"/>	202001006	Maya Indah	Jl. Cempaka 89	maya.indah@email.com	22	P	Mahasiswa
<input type="checkbox"/>	202001007	Dika Wijaya	Jl. Kencana 34	dika.wijaya@email.com	22	L	Mahasiswa
<input type="checkbox"/>	202001008	Dewi Cahaya	Jl. Surya 56	dewi.cahaya@email.com	22	P	Mahasiswa
<input type="checkbox"/>	202001009	Fajar Prasetyo	Jl. Mawar 78	fajar.prasetyo@email.com	21	L	Mahasiswa
<input type="checkbox"/>	202001010	Siti Nurul	Jl. Anggrek 90	siti.nurul@email.com	22	P	Mahasiswa
<input type="checkbox"/>	202011001	Yoga Santoso	Jl. Dahlia 12	yoga.santoso@email.com	34	L	Dosen
- Filter and sort options: "Show all", "Number of rows: 25", "Filter rows: Search this table", "Sort by key: None".

2. Table Anggota

The screenshot shows the MySQL Workbench interface with the following details:

- Query results header: "Showing rows 0 - 10 (11 total, Query took 0.0007 seconds.)"
- SQL query: "SELECT * FROM `anggota`"
- Table structure:

	ID_anggota	username	password	No_inuk
<input type="checkbox"/>	A001	lina_damayanti	9a1b2be94827d1e47f11fcae5e07fa796461315bd3d47a17f0...	202001001
<input type="checkbox"/>	A002	agus_putra	31dc47f23ec2d7ad58b676c12d0f095852c294e25f1e9e00f3...	202001002
<input type="checkbox"/>	A003	andi_setiawan	e6d0396d0dcfda0f6d98f6e4f4a7cdd10a7b1fdd09c2c547f1...	202001003
<input type="checkbox"/>	A004	rini_sari	441abe176fd485c28c5d56dd6ae76d14cf47f4ed2039261258...	202001004
<input type="checkbox"/>	A005	budi_pratama	5c3d1aa3c4127b4c61dc7755f79d369cfda003e234dcdbd3d5d...	202001005
<input type="checkbox"/>	A006	maya_indah	2a04c7a4b1b5ca7c81e9e6019b441d04387d3af8583b26568b...	202001006
<input type="checkbox"/>	A007	dika_wijaya	a1e6cda93d3f1ac46270a77c5bbf9eafa12f8e0e95025473a2...	202001007
<input type="checkbox"/>	A008	dewi_cahaya	7a6c6a449b08d529df42a5b2d26c66eeac13bfdc603c7b3b35...	202001008
<input type="checkbox"/>	A009	fajar_prasetyo	40f9d4512c9af417f7fe028f37532406f9c20a3e1e0c889022...	202001009
<input type="checkbox"/>	A010	siti_nurul	5d6e8562eec9ff5b34ea601618c766a7b2222b02080b7ace4e...	202001010
<input type="checkbox"/>	A011	yoga_santoso	b8f42135d8c6a9b47f0f16307c6bfbe80b8e7bb2434471a1ff...	202011001
- Filter and sort options: "Show all", "Number of rows: 25", "Filter rows: Search this table", "Sort by key: None".

3. Table Super User

Showing rows 0 - 4 (5 total, Query took 0.0005 seconds.)

```
SELECT * FROM `superuser`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	ID_superuser	username_superuser	password_superuser
<input type="checkbox"/>	Edit Copy Delete SU001	Ahmadfariz	8a221bb139510e5cbdaca09b82930346d6b27748387ab10ba9...
<input type="checkbox"/>	Edit Copy Delete SU002	fitriani	b6305322bce74fdf912e0101ce2e6464135b86c85a584f062c...
<input type="checkbox"/>	Edit Copy Delete SU003	madian	b297fd788e59cbae9debdb6b5795ca4fa94c5215310f215951...
<input type="checkbox"/>	Edit Copy Delete SU004	nabila	edd99d048235c3c0fc4ad913bb3ee81587314f9e93e85b6fb...
<input type="checkbox"/>	Edit Copy Delete SU005	rendy	a3f15df651e11b8566bad0150c1caa3ee02a0666c150c6689a...

4. Table Kategori

Showing rows 0 - 9 (10 total, Query took 0.0006 seconds.)

```
SELECT * FROM `kategori` WHERE 1;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]

Show all | Number of rows: 25 Filter row

Extra options

	no_kategori	jenis
<input type="checkbox"/>	K01	Teknologi
<input type="checkbox"/>	K02	Sejarah
<input type="checkbox"/>	K03	Fiksi
<input type="checkbox"/>	K04	Agama
<input type="checkbox"/>	K05	Sains
<input type="checkbox"/>	K06	Politik
<input type="checkbox"/>	K07	Olahraga
<input type="checkbox"/>	K08	Bahasa
<input type="checkbox"/>	K09	Seni
<input type="checkbox"/>	K10	Ekonomi

5. Table Penerbit

The screenshot shows the MySQL Workbench interface with the following details:

- Query results header: Showing rows 0 - 9 (10 total, Query took 0.0004 seconds.)
- SQL query: `SELECT * FROM `penerbit` WHERE 1;`
- Table structure: ID_penerbit, nama_penerbit
- Data rows (10 total):
 - PB01 Erlangga
 - PB02 Gramedia
 - PB03 Mizan
 - PB04 Kompas
 - PB05 A3
 - PB06 Naya Jaya
 - PB07 Ertiga
 - PB08 Matahari
 - PB09 Pustaka
 - PB10 Dahwa Jawa

6. Table Buku

The screenshot shows the MySQL Workbench interface with the following details:

- Query results header: Showing rows 0 - 9 (10 total, Query took 0.0006 seconds.)
- SQL query: `SELECT * FROM `buku` WHERE 1;`
- Table structure: no_buku, judul, tahun_terbit, ID_penerbit, no_kategori
- Data rows (10 total):
 - B0001 Pemrograman Dasar 2005 PB01 K01
 - B0002 Batavia 1930 PB02 K02
 - B0003 Gadis Kretek 2014 PB03 K03
 - B0004 Bismillah 2004 PB04 K04
 - B0005 IPA Dasar 2020 PB05 K05
 - B0006 Demokrasi 1999 PB06 K06
 - B0007 Raket Naya 2019 PB07 K07
 - B0008 KBBI 2000 PB08 K08
 - B0009 Cipta Syair 2002 PB09 K09
 - B0010 Akuntansi 2022 PB10 K10

7. Tabel Peminjaman

Showing rows 0 - 9 (10 total, Query took 0.0006 seconds.)

```
SELECT * FROM `peminjaman` WHERE 1;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	ID_peminjaman	tanggal_pinjam	tanggal_kembali	tarif_denda	keterangan_denda	ID_anggota	no_buku
<input type="checkbox"/>	PJM001	2023-01-05	2023-01-15	0	-	A001	B0001
<input type="checkbox"/>	PJM002	2023-02-10	2023-02-20	0	-	A002	B0002
<input type="checkbox"/>	PJM003	2023-03-15	2023-03-25	20000	Rusak	A003	B0003
<input type="checkbox"/>	PJM004	2023-04-20	2023-04-30	50000	Hilang	A004	B0004
<input type="checkbox"/>	PJM005	2023-05-25	2023-06-04	0	-	A005	B0005
<input type="checkbox"/>	PJM006	2023-06-30	2023-07-10	2000	Terlambat	A006	B0006
<input type="checkbox"/>	PJM007	2023-08-05	2023-08-15	0	-	A007	B0007
<input type="checkbox"/>	PJM008	2023-09-10	2023-09-20	0	-	A008	B0008
<input type="checkbox"/>	PJM009	2023-10-15	2023-10-25	5000	Rusak	A009	B0009
<input type="checkbox"/>	PJM010	2023-11-20	2023-11-30	0	-	A010	B0010

4.2 User

1. Admin

```
CREATE USER 'admin'@'%' IDENTIFIED BY 'admin';
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'%' WITH GRANT OPTION;
```

2. Super User

```
create user 'superuser'@'%' identified by 'superuser';

CREATE VIEW superuser_superuser AS
SELECT id_superuser, username_superuser from perpustakaan.superuser;

GRANT SELECT ON perpustakaan.superuser_superuser TO 'superuser'@'%';
GRANT SELECT, INSERT, UPDATE ON perpustakaan.pengunjung TO 'superuser'@'%';
GRANT SELECT, INSERT, UPDATE ON perpustakaan.anggota TO 'superuser'@'%';
GRANT SELECT, INSERT, UPDATE ON perpustakaan.kategori TO 'superuser'@'%';
GRANT SELECT, INSERT, UPDATE ON perpustakaan.buku TO 'superuser'@'%';
GRANT SELECT, INSERT, UPDATE ON perpustakaan.peminjaman TO 'superuser'@'%';
GRANT SELECT, INSERT, UPDATE ON perpustakaan.penerbit TO 'superuser'@'%';
```

3. User

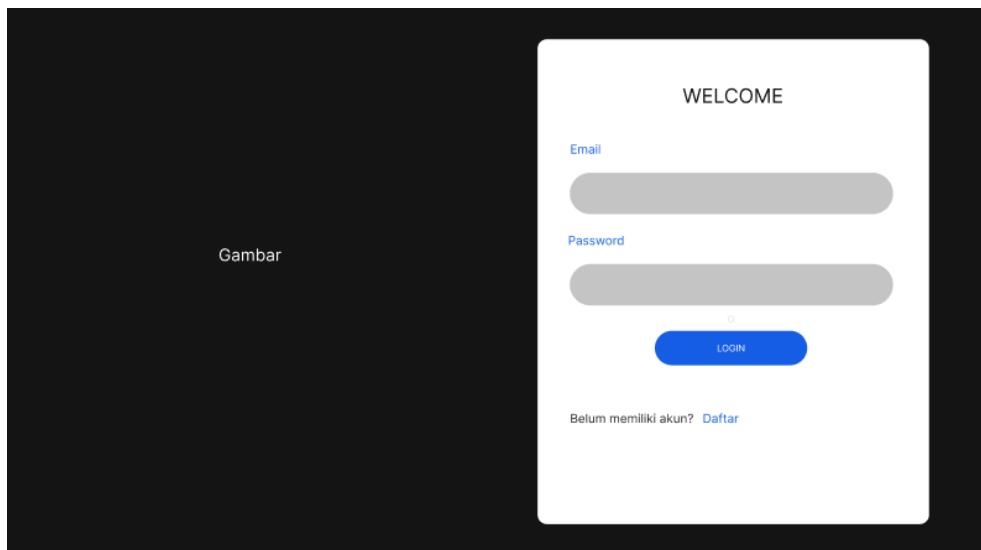
```
create user 'user'@'%' identified by 'user';

CREATE VIEW pengunjung_user AS
SELECT nama, umur, jenis_kelamin, no_induk, status from perpustakaan.pengunjung;
CREATE VIEW anggota_user AS
SELECT nama, id_anggota, username, no_induk from perpustakaan.anggota;

GRANT SELECT ON perpustakaan.pengunjung_user TO 'user'@'%';
GRANT SELECT ON perpustakaan.anggota_user TO 'user'@'%';
GRANT SELECT ON perpustakaan.buku TO 'user'@'%';
GRANT SELECT ON perpustakaan.kategori TO 'user'@'%';
GRANT SELECT ON perpustakaan.penerbit TO 'user'@'%';
GRANT SELECT ON perpustakaan.peminjaman TO 'user'@'%';
```

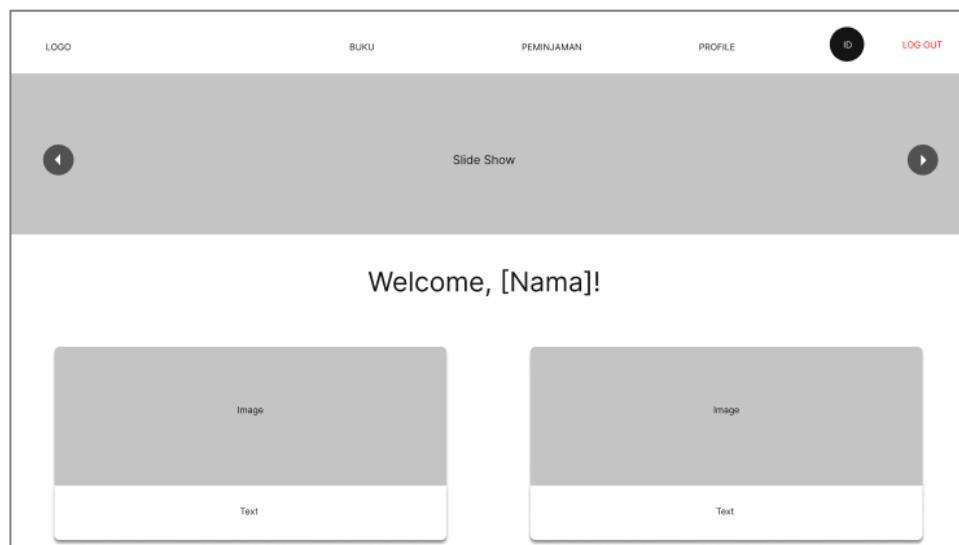
4.3 Wireframe

1. Wireframe Halaman Login



The wireframe shows a dark-themed login screen. At the top center, it says "WELCOME". Below that are two input fields: "Email" and "Password", each with a grey placeholder bar. Underneath the password field is a small circular icon with a question mark. To the right of the fields is a blue "LOGIN" button. At the bottom of the screen, there is a link "Belum memiliki akun? [Daftar](#)". The word "Gambar" is written in the middle-left area of the dark background.

2. Wireframe Halaman Dashboard



The wireframe displays a dashboard layout. At the top, there is a navigation bar with "LOGO", "BUKU", "PEMINJAMAN", "PROFILE", a user icon labeled "ID", and a red "LOG OUT" button. Below the navigation is a grey header section with a left arrow, a right arrow, and the text "Slide Show" in the center. The main content area starts with a welcome message "Welcome, [Nama]!". Below this are two large rectangular boxes, each containing a grey "Image" placeholder at the top and a white "Text" placeholder at the bottom. The entire dashboard has a light grey background.

3. Wireframe Halaman Profile

The wireframe for the Profile page features a header with navigation links: LOGO, BUKU, PEMINJAMAN, PROFILE, and a user ID icon with a LOG OUT link. The main content area contains five input fields for personal information: ID, Nama, Umur, No Induk, and Status.

ID :	
Nama :	
Umur :	
No Induk :	
Status :	

4. Wireframe Halaman Buku

The wireframe for the Book page includes a header with the same navigation links: LOGO, BUKU, PEMINJAMAN, PROFILE, and a user ID icon with a LOG OUT link. It features a search bar with a magnifying glass icon and the placeholder text "Cari Buku...". Below the search bar is a table listing books, with columns for No Buku, Judul, Penerbit, Tahun Terbit, and Kategori. The table has five rows of placeholder data.

No Buku	Judul	Penerbit	Tahun Terbit	Kategori

5. Wireframe Halaman Peminjaman

The wireframe for the Lending Page (Peminjaman) features a header with navigation links: LOGO, BUKU, PEMINJAMAN, PROFILE, and a user icon with 'ID' and 'LOG OUT'. Below the header is a search bar with a magnifying glass icon and the placeholder text 'Cari Peminjaman...'. The main content area contains a table with columns: ID Peminjaman, Tanggal Pinjam, Tanggal Kembali, Tarif Denda, Keterangan Denda, and Judul Buku. There are six rows of data in the table.

6. Wireframe Halaman Anggota

The wireframe for the Member Page (Anggota) features a header with navigation links: LOGO, BUKU, PEMINJAMAN, PROFILE, and a user icon with 'ID' and 'LOG OUT'. Below the header is a search bar with a magnifying glass icon and the placeholder text 'Cari Anggota...'. The main content area contains a table with columns: ID, Nama, No Induk, Jenis Kelamin, Umur, and Status. There are six rows of data in the table.

BAB V

ANALISIS

5.1 Query Processing

1. Halaman Login

Pada halaman login, diperlukan table super user, dan user atau anggota dimana setiap table tersebut menyimpan username dan password. Pada table anggota, perlu dibuat indeks dikarenakan data yang berada dalam table ini cukup banyak sehingga dapat mempercepat waktu pencarian.

- Mencari Username dan Password Yang Sesuai

A. Query

1. `SELECT username, password FROM anggota WHERE username = '$username' AND password = '$password';`
2. `SELECT username, password FROM (SELECT username, password FROM anggota) AS temp WHERE username = '$username' AND password = '$password';`

B. Relational Algebra

1. $\Pi_{\text{username}, \text{password}}(\sigma_{\text{username} = \$username} \wedge \sigma_{\text{password} = \$password}(\text{anggota}))$
2. $\sigma_{\text{username} = \$username} \wedge \sigma_{\text{password} = \$password}(\Pi_{\text{username}, \text{password}}(\text{anggota}))$

Berdasarkan 3 tahun kedepan, query yang paling optimal ialah query pertama.

2. Halaman Dashboard

Pada halaman dashboard, diperlukan table pengunjung untuk menampilkan nama. halaman ini juga berisi image dan text mengenai perpustakaan.

- Menampilkan Nama Anggota

A. Query

1. SELECT nama FROM pengunjung WHERE no_induk= '\$no_induk'
2. SELECT nama FROM (SELECT nama, no_induk FROM pengunjung) AS temp WHERE no_induk = '\$no_induk';

B. Relational Algebra

1. $\Pi_{\text{nama}}(\sigma_{\text{no_induk} = ' \$no_induk'}(\text{pengunjung}))$
2. $\sigma_{\text{no_induk} = ' \$no_induk'}(\Pi_{\text{nama}}, \text{no_induk}(\text{pengunjung}))$

Berdasarkan 3 tahun kedepan, query yang paling optimal ialah query pertama.

3. Halaman Profile

Pada halaman profile, diperlukan table anggota dimana table tersebut menampilkan informasi pribadi seperti nama, umur, jenis kelamin, dan data lainnya pada anggota itu sendiri. Pada table ini, perlu dibuat indeks dikarenakan data yang berada dalam table ini cukup banyak sehingga dapat mempercepat waktu pencarian.

- Menampilkan Informasi Anggota

A. Query

1. SELECT anggota.ID_anggota, pengunjung.nama, pengunjung.no_induk, pengunjung.umur, pengunjung.status, pengunjung.jenis_kelamin FROM anggota JOIN pengunjung ON anggota.no_induk = pengunjung.no_induk where anggota.ID_anggota = ' \\$ID_anggota';
2. SELECT ID_anggota, nama, no_induk, umur, status, jenis_kelamin FROM (SELECT anggota.ID_anggota, pengunjung.nama, pengunjung.no_induk, pengunjung.umur, pengunjung.status, pengunjung.jenis_kelamin FROM anggota JOIN pengunjung ON anggota.no_induk = pengunjung.no_induk) AS temp WHERE temp.ID_anggota = ' \\$ID_anggota';

B. Relational Algebra

1. $\pi_{\text{ID_anggota}, \text{nama}, \text{no_induk}, \text{umur}, \text{status}, \text{jenis_kelamin}}(\sigma_{\text{ID_anggota} = ' \$ID_anggota'}(\text{anggota} \bowtie \text{anggota.no_induk} = \text{pengunjung.no_induk}(\text{pengunjung})))$
2. $\sigma_{\text{ID_anggota} = ' \$ID_anggota'}(\pi_{\text{ID_anggota}, \text{nama}, \text{no_induk}, \text{umur}, \text{status}, \text{jenis_kelamin}}(\text{anggota} \bowtie \text{anggota.no_induk} = \text{pengunjung.no_induk}(\text{pengunjung})))$

Berdasarkan 3 tahun kedepan, query yang paling optimal ialah query pertama.

4. Halaman Buku

Pada halaman profile, diperlukan table buku dimana table tersebut menampilkan semua informasi tentang buku seperti no_buku, judul, id_penerbit, tahun_terbit, kategori. Jika ingin mencari buku yang lain, maka dapat melakukan pencarian pada kolom pencarian.

- Menampilkan Buku

A. Query

1. SELECT buku.no_buku, buku.judul, buku.tahun_terbit, penerbit.nama_penerbit, kategori.jenis FROM Buku JOIN Penerbit on buku.id_penerbit = penerbit.id_penerbit JOIN Kategori on buku.no_kategori = kategori.no_kategori;
2. SELECT buku.no_buku, buku.judul, buku.tahun_terbit, penerbit.nama_penerbit, kategori.jenis FROM buku, penerbit, kategori WHERE buku.id_penerbit = penerbit.id_penerbit AND buku.no_kategori = kategori.no_kategori;

B. Relational Algebra

1. $\pi_{no_buku, judul, tahun_terbit, nama_penerbit, jenis}((buku \bowtie b\{u\}ku.id_penerbit = p\{e\}nerbit.id_penerbit) \bowtie b\{u\}ku.no_kategori = kategori.no_kategori(kategori))$
2. $\pi_{no_buku, judul, tahun_terbit, nama_penerbit, jenis}(\sigma_{b\{u\}ku.id_penerbit = p\{e\}nerbit.i_d_penerbit} \wedge b\{u\}ku.no_kategori = kategori.no_kategori(b\{u\}ku \times p\{e\}nerbit \times kategori))$

Berdasarkan 3 tahun kedepan, query yang paling optimal ialah query pertama.

- Mencari Buku

A. Query

1. SELECT judul FROM buku WHERE judul = '\$judul';
2. SELECT buku.judul FROM buku, (SELECT DISTINCT judul FROM buku) AS

temp WHERE buku.judul = temp.judul AND buku.judul = '\$judul';

B. Relational Algebra

1. $\pi_{\text{judul}}(\sigma_{\text{judul}='\$judul'}(\text{buku}))$
2. $\pi_{\text{judul}}(\sigma_{\text{buku.judul}=\text{temp.judul} \wedge \text{buku.judul}='\$judul'}(\text{buku} \times \text{temp}))$

Berdasarkan 3 tahun kedepan, query yang paling optimal ialah query pertama.

5. Halaman Peminjaman

Pada pengguna, halaman ini menampilkan buku peminjaman yang dilakukan oleh pengguna. Pada super user, halaman ini menampilkan buku peminjaman yang dilakukan oleh pengguna serta aksi untuk create dan update. Table yang ditampilkan ialah table peminjaman yang berisi peminjaman.ID_peminjaman, peminjaman.tanggal_pinjam,peminjaman.tanggal_kembali,peminjaman.tarif_denda, peminjaman.keterangan_denda dan table buku yang berisi judul_buku

- Menampilkan Peminjaman

A. Query

1. $\text{SELECT peminjaman.ID_peminjaman, peminjaman.tanggal_pinjam, peminjaman.tanggal_kembali, peminjaman.tarif_denda, peminjaman.keterangan_denda, buku.judul_buku FROM peminjaman JOIN buku ON peminjaman.no_buku = buku.no_buku WHERE peminjaman.ID_peminjaman = '\$ID_peminjaman';}$
2. $\text{SELECT peminjaman.ID_peminjaman, peminjaman.tanggal_pinjam, peminjaman.tanggal_kembali, peminjaman.tarif_denda, peminjaman.keterangan_denda, buku.judul_buku FROM peminjaman, buku WHERE peminjaman.no_buku = buku.no_buku AND peminjaman.ID_peminjaman = '\$ID_peminjaman';}$

B. Relational Algebra

1. $\pi_{\text{ID_peminjaman}, \text{tanggal_pinjam}, \text{tanggal_kembali}, \text{tarif_denda}, \text{keterangan_denda}, \text{judul_buku}}(\sigma_{\text{ID_peminjaman}='\$ID_peminjaman'}(\text{peminjaman} \bowtie \text{peminjaman.no_buku} = \text{buku.no_buku}(\text{buku})))$

2. $\pi_{ID_peminjaman, tanggal_pinjam, tanggal_kembali, tarif_denda, keterangan_denda, judul_buku}(\sigma_{peminjaman.no_buku=buku.no_buku \wedge ID_peminjaman='\$ID_peminjaman'}(peminjaman \times buku))$

Berdasarkan 3 tahun kedepan, query yang paling optimal ialah query pertama.

- Mencari Peminjaman

A. Query

1. `SELECT ID_peminjaman FROM peminjaman WHERE tanggal_pinjam = '$tanggal_pinjam' AND tanggal_kembali = '$tanggal_kembali';`
2. `SELECT SELECT peminjaman.ID_peminjaman FROM peminjaman, (SELECT DISTINCT tanggal_pinjam, tanggal_kembali FROM peminjaman) AS temp WHERE peminjaman.tanggal_pinjam = temp.tanggal_pinjam AND peminjaman.tanggal_kembali = temp.tanggal_kembali AND peminjaman.tanggal_pinjam = '$tanggal_pinjam' AND peminjaman.tanggal_kembali = '$tanggal_kembali';`

B. Relational Algebra

1. $\pi_{ID_peminjaman}(\sigma_{tanggal_pinjam='\$tanggal_pinjam' \wedge tanggal_kembali='\$tanggal_kembali'}(peminjaman)) \pi_{judul}(\sigma_{buku.judul=temp.judul \wedge buku.judul='\$judul'}(buku \times temp))$
2. $\pi_{ID_peminjaman}(\sigma_{peminjaman.tanggal_pinjam=temp.tanggal_pinjam \wedge peminjaman.tanggal_kembali=temp.tanggal_kembali \wedge peminjaman.tanggal_pinjam='\$tanggal_pinjam' \wedge peminjaman.tanggal_kembali='\$tanggal_kembali'}(peminjaman \times temp))$

Berdasarkan 3 tahun kedepan, query yang paling optimal ialah query pertama.

- Mengupdate Peminjaman

A. Query dengan Transaksi

`START TRANSACTION;`

`UPDATE peminjaman SET tanggal_pinjam = ' \$tanggal_ pinjam ',`

```
tanggal_kembali = '$tanggal_kembali ', WHERE ID_peminjaman =
'$ID_peminjaman';
COMMIT;
```

B. Relational Algebra

1. $R1 \leftarrow \sigma_{ID_peminjaman='\$ID_peminjaman'}(peminjaman)$
 $R2 \leftarrow \tau_{tanggal_pinjam='\$tanggal_pinjam'}(R1)$
 $R3 \leftarrow \tau_{tanggal_kembali='\$tanggal_kembali'}(R2)$
 $peminjaman \leftarrow R3$
2. $R1 \leftarrow \tau_{tanggal_pinjam='\$tanggal_pinjam', tanggal_kembali='\$tanggal_kembali'}(\sigma_{ID_peminjaman='\$ID_peminjaman'}(peminjaman))$
 $R2 \leftarrow \Pi_{ID_peminjaman, tanggal_pinjam, tanggal_kembali}(R1)$
 $peminjaman \leftarrow R2$

Berdasarkan 3 tahun kedepan, query yang paling optimal ialah query kedua.

6. Halaman Anggota

Pada halaman anggota, diperlukan table anggota untuk menampilkan semua informasi seperti nama, umur, jenis kelamin, alamat, dan data lainnya pada seluruh anggota. Halaman ini digunakan super user untuk monitoring data anggota perpustakaan. Halaman ini hanya menampilkan 50 buku teratas untuk optimal dan efisiensi. Jika ingin mencari buku yang lain, maka dapat melakukan pencarian pada kolom pencarian.

- Menampilkan Anggota

A. Query

1.

```
SELECT anggota.ID_anggota, anggota.no_induk, anggota.username,
anggota.password, pengunjung.status, pengunjung.jenis_kelamin,
pengunjung.umur, pengunjung.nama FROM anggota JOIN pengunjung ON
```

- anggota.no_induk = pengunjung.no_induk;
2. SELECT anggota.ID_anggota, anggota.no_induk, anggota.username, anggota.password, pengunjung.status, pengunjung.jenis_kelamin, pengunjung.umur, pengunjung.nama FROM anggota, pengunjung WHERE anggota.no_induk = pengunjung.no_induk;

B. Relational Algebra

1. $\pi_{ID_anggota,no_induk,username,password,status,jenis_kelamin,umur,nama}(anggota \bowtie_{anggota.no_induk=pengunjung.no_induk} pengunjung)$
2. $\pi_{ID_anggota,no_induk,username,password,status,jenis_kelamin,umur,nama}(\sigma_{anggota.no_induk=pengunjung.no_induk}(anggota \times pengunjung))$

Berdasarkan 3 tahun kedepan, query yang paling optimal ialah query pertama.

- Mencari Anggota

A. Query

1. SELECT anggota.ID_anggota, pengunjung.nama FROM anggota JOIN pengunjung ON anggota.no_induk = pengunjung.no_induk WHERE ID_anggota = '\$ID_anggota';
2. SELECT anggota.ID_anggota, pengunjung.nama FROM anggota, pengunjung WHERE anggota.no_induk = pengunjung.no_induk AND anggota.ID_anggota = '\$ID_anggota';

B. Relational Algebra

1. $\pi_{ID_anggota,nama}(\sigma_{ID_anggota='\$ID_anggota'}(anggota \bowtie_{anggota.no_induk=pengunjung.no_induk} pengunjung))$
2. $\pi_{ID_anggota,nama}(\sigma_{anggota.no_induk=pengunjung.no_induk \wedge anggota.ID_anggota='\$ID_anggota'}(anggota \times pengunjung))$

Berdasarkan 3 tahun kedepan, query yang paling optimal ialah query pertama.

5.2 Ukuran Data & Blok yang Dipakai

Dalam menghitung ukuran data dan blok yang digunakan pada setiap tabel, langkah-langkah yang digunakan ialah :

- Hitung ukuran per record (row): Jumlahkan ukuran semua atribut dalam satu record.
- Hitung total ukuran data: Kalikan ukuran per record dengan jumlah data (record).
- Hitung jumlah blok yang digunakan: Bagi total ukuran data dengan ukuran blok (default ukuran blok di MySQL adalah 16 KB atau 16,384 bytes).

Berikut ialah tabel-tabel jumlah rata-rata data yang mencerminkan keadaan operasional perpustakaan Fakultas Ilmu Komputer Universitas Sriwijaya pada 3 tahun ke depan:

No	Nama Tabel	Ukuran Data	Blok Digunakan
1	Pengunjung	617 KB	39 blok
2	Anggota	864 KB	55 blok
3	Buku	4.09 MB	262 blok
4	Kategori	1.7 KB	1 blok
5	Peminjaman	8.3 MB	531 blok
6	Penerbit	44 KB	3 blok
7	Superuser	1.4 KB	1 blok

1. Tabel Pengunjung

- Ukuran per record: $22+52+102+52+3+2+12=245$ bytes
- Total ukuran data: $245 \text{ bytes} \times 2578 \text{ records}=631,610 \text{ bytes} (\approx 617 \text{ KB})$
- Jumlah blok: $[631,610/16,384]=39$ blok

2. Tabel Anggota

- Ukuran per record: $12+52+257+22=343$ bytes
- Total ukuran data: $343 \text{ bytes} \times 2578 \text{ records}=884,654$ bytes
- Jumlah blok: $[884,654/16,384]=55$ blok

3. Tabel Buku

- Ukuran per record: $12+102+5+12+12=143$ bytes
- Total ukuran data: $143 \text{ bytes} \times 30,000 \text{ records}=4,290,000$ bytes (≈ 4.09 MB)
- Jumlah blok: $[4,290,000/16,384]=262$ blok

4. Tabel Kategori

- Ukuran per record: $12+22=34$ bytes
- Total ukuran data: $34 \text{ bytes} \times 50 \text{ records}=1,700$ bytes (≈ 1.7 KB)
- Jumlah blok: $[1,700/16,384]=1$ blok

5. Tabel Peminjaman

- Ukuran per record: $12+30+30+11+52+12+12=159$ bytes
- Total ukuran data: $159 \text{ bytes} \times 54,750 \text{ records}=8,702,250$ bytes (≈ 8.3 MB)
- Jumlah blok: $[8,702,250/16,384]= 531$ blok

6. Tabel Penerbit

- Ukuran per record: $12+52=64$ bytes
- Total ukuran data: $64 \text{ bytes} \times 700 \text{ records}=44,800$ bytes (≈ 44 KB)
- Jumlah blok: $[44,800/16,384]=3$ blok

7. Tabel Superuser

- Ukuran per record: $12+22+257=291$ bytes
- Total ukuran data: $291 \text{ bytes} \times 5 \text{ records}=1,455$ bytes (≈ 1.4 KB)
- Jumlah blok: $[1,455/16,384]=1$ blok

Tabel dengan jumlah record yang besar, seperti Buku dan Peminjaman, memerlukan blok lebih banyak.

5.3 Indexing B-Tree

1. Index untuk Pencarian Buku Berdasarkan Judul

Index ini berfungsi untuk mempercepat pencarian buku berdasarkan judulnya. Table yang digunakan ialah buku.

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0252 seconds.)

```
CREATE INDEX idx_cari_buku ON buku(judul) USING BTREE;
```

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

2. Index untuk Pencarian Peminjaman Berdasarkan Tanggal Pinjam dan Tanggal Kembali

Index ini berfungsi untuk mempercepat pencarian peminjaman berdasarkan tanggal pinjam dan tanggal kembali. Table yang digunakan ialah peminjaman.

✓ 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]

5.4 Durability

Durability adalah salah satu dari ACID properties (Atomicity, Consistency, Isolation, Durability) yang merupakan prinsip dasar dalam sistem manajemen basis data (DBMS) untuk memastikan integritas dan konsistensi data. Durability mengacu pada kemampuan sistem untuk memastikan bahwa data yang telah diubah tetap aman dan persisten meskipun terjadi gangguan sistem, seperti kegagalan perangkat keras, kerusakan perangkat, atau bahkan pemadaman listrik

5.4.1 Replikasi Database

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 2 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.

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

```
CHANGE MASTER TO
MASTER_HOST = '192.186.72.128',
MASTER_LOG_FILE = 'mysql-bin.000001',
MASTER_LOG_POS = 714,
MASTER_SSL = 1;

START SLAVE;
```

Replikasi pun berhasil dijalankan.

5.5 Recovery Plan

Recovery Plan adalah langkah-langkah strategis yang dirancang untuk memulihkan sistem basis data dari kegagalan atau gangguan, baik yang disebabkan oleh kerusakan perangkat keras, kesalahan perangkat lunak, maupun bencana lainnya.

5.5.1 Recovery Sebelum Direset

- Backup seluruh database ke hard drive, kemudian cek dan pastikan file back up dapat dipulihkan
- Pastikan semua disk dalam kondisi sehat dan RAID tidak mengalami degradasi
- Hitung pemakaian daya server MySQL dan UPS agar daya mencukupi selama pemadaman listrik sementara.
- Pastikan genset sudah dikonfigurasi dan dapat menyala
- Persiapkan rencana rollback, agar jika setelah reset terjadi masalah maka ada mekanisme untuk mengembalikan sistem ke kondisi semula.
- Memastikan tidak ada transaksi penting yang sedang berlangsung, dan tidak ada proses input data yang terganggu selama proses reset.
- Dokumentasikan semua perubahan yang dilakukan sebelum reset.

5.5.2 Pencegahan Crash

- Pemeliharaan berkala pada server dan jaringan setiap 6 bulan.
- Lakukan backup rutin setiap 3 bulan dan simpan di hard drive eksternal serta cloud (Google Drive atau AWS S3).
- Menggunakan RAID 5 untuk memastikan tidak ada kehilangan data jika salah satu disk rusak.
- Sewa cloud storage untuk backup di luar lokasi
- Menggunakan jaringan internet cepat dan stabil untuk akses remote dan backup cloud. Dapat dilakukan dengan sewa internet fiber

5.6 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.

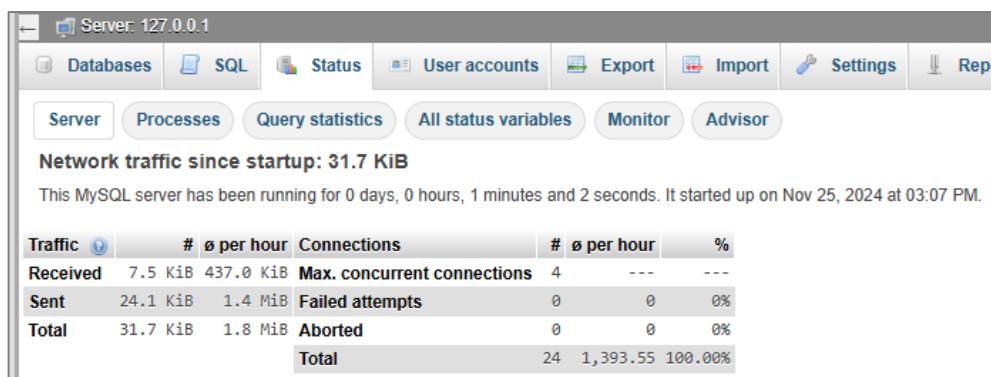
Monitoring databases memiliki beberapa manfaat utama, antara lain:

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.

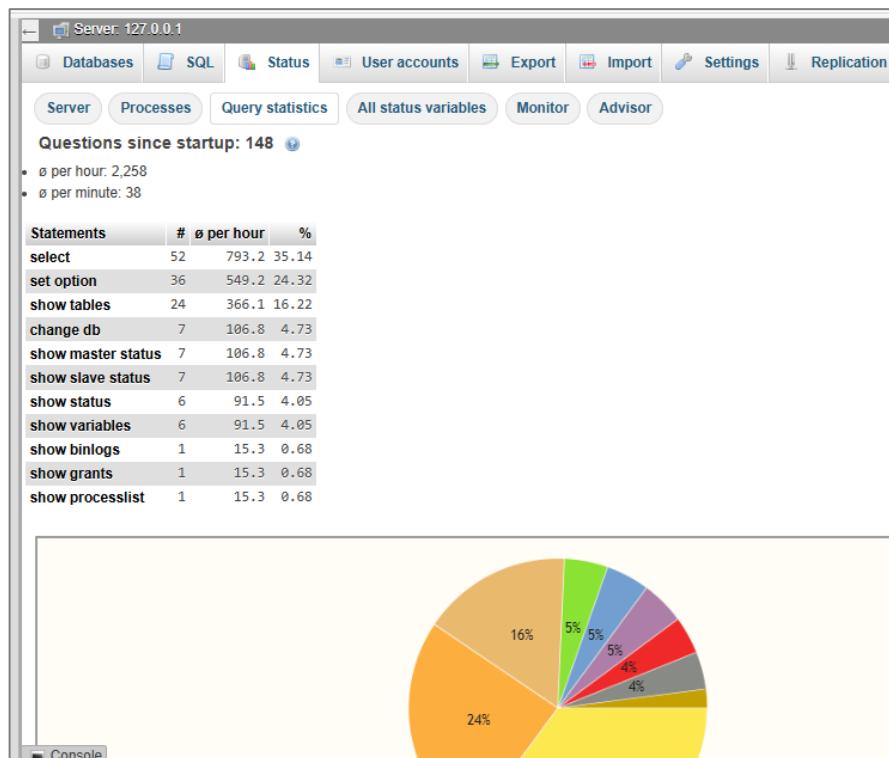
7. Memfasilitasi perbaikan berkelanjutan: Menciptakan siklus pembaruan dan peningkatan rutin untuk mengoptimalkan kesehatan basis data dari waktu ke waktu.

Berikut adalah implementasi monitoring database pada sistem ini :

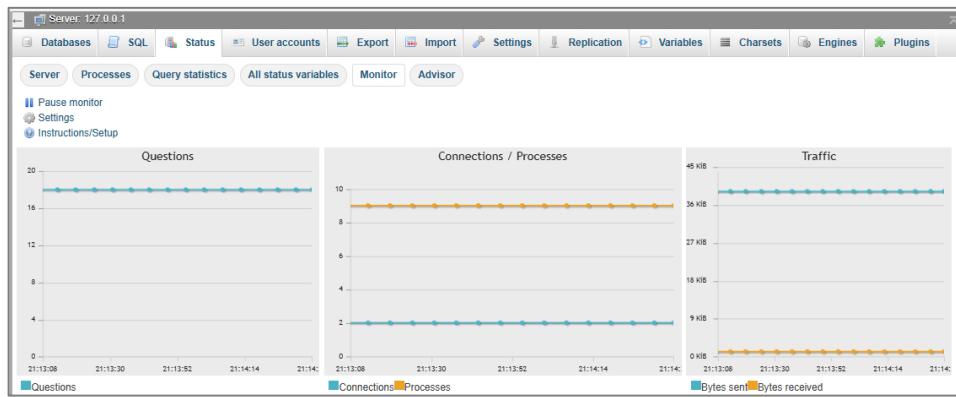
1. Buka Phpmyadmin
2. Buka tab status



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



4. Buka Monitor untuk melihat tabel yang paling seringdiakses atau dimodifikasi.



5.7 Maintenance

Maintenance atau pemeliharaan sistem dilakukan untuk memastikan bahwa sistem perpustakaan berbasis web tetap berfungsi optimal dan mampu memenuhi kebutuhan pengguna. Pemeliharaan ini mencakup kegiatan rutin yang bertujuan untuk memperpanjang usia layanan sistem, mengurangi downtime, dan meminimalkan risiko kerusakan data. Berikut adalah beberapa aspek penting yang dapat diimplementasikan dalam pemeliharaan sistem perpustakaan:

1. Pemantauan Kinerja Sistem

Pemantauan kinerja sistem secara berkala dilakukan untuk mengidentifikasi masalah performa seperti lambatnya respon query. Ini mencakup pengawasan pada penggunaan CPU, memori, dan kapasitas penyimpanan.

2. Pembersihan dan Pengelolaan Database

Data di dalam sistem perpustakaan perlu dibersihkan secara berkala untuk menghindari redundansi dan menjaga integritas data. Hal ini termasuk menghapus data tidak relevan, melakukan indexing ulang jika diperlukan, dan melakukan backup data secara berkala.

3. Backup dan Recovery

Backup rutin sangat penting untuk menjaga data agar tidak hilang akibat kesalahan sistem atau serangan siber. Recovery plan juga disiapkan agar dapat dengan cepat mengembalikan data dan sistem ke kondisi semula jika terjadi kegagalan atau bencana.

4. Peningkatan Sistem Berkala

Teknologi dan kebutuhan pengguna dapat berubah seiring waktu, sehingga sistem perlu diperbarui. Hal ini meliputi peningkatan fitur, perbaikan bug, dan peningkatan keamanan.

5. Dokumentasi dan Evaluasi

Setiap tindakan pemeliharaan harus terdokumentasi dengan baik untuk memudahkan tim dalam melakukan evaluasi terhadap efektivitas pemeliharaan. Evaluasi ini juga membantu menentukan kebutuhan upgrade atau modifikasi pada sistem di masa depan.

5.8 Evaluasi Performa Sistem

Evaluasi performa sistem dilakukan untuk memastikan efisiensi dan keandalan sistem perpustakaan berbasis website dalam menghadapi beban kerja saat ini dan proyeksi pertumbuhan data selama 3 tahun ke depan. Evaluasi mencakup analisis waktu eksekusi query, penggunaan indeks, kapasitas hardware, dan kemampuan sistem dalam menangani transaksi harian.

Penggunaan indeks pada atribut kunci seperti ID_peminjaman dan judul terbukti meningkatkan performa query secara signifikan. Berdasarkan pengujian dengan EXPLAIN, query pencarian buku berdasarkan judul yang menggunakan indeks menunjukkan waktu eksekusi rata-rata yang lebih cepat dibandingkan tanpa indeks pada tabel berisi 30.000 data. Hal ini menegaskan pentingnya penerapan indeks untuk query dengan filter yang sering digunakan.

Berdasarkan proyeksi data selama 3 tahun ke depan, dengan total 54.750 transaksi peminjaman dan pengembalian, serta pertumbuhan data anggota dan buku, kapasitas hardware pada sistem ini sudah cukup baik, khususnya RAM dan penyimpanan. Dengan RAM 32GB saat ini, sistem mampu menangani hingga 50 transaksi per hari tanpa penurunan performa yang signifikan.

Pengujian beban transaksi menunjukkan bahwa sistem mampu menangani hingga 50 transaksi bersamaan tanpa mengalami bottleneck, berkat penggunaan indeks dan transaksi berbasis ACID. sistem dapat menangani pertumbuhan data tanpa memengaruhi waktu respon.

Sistem perpustakaan berbasis website dinilai mampu mendukung kegiatan operasional selama 3 tahun ke depan tanpa peningkatan hardware tambahan. Namun, untuk mendukung pertumbuhan data dan pengguna yang lebih besar setelah 3 tahun, diperlukan optimasi query lebih lanjut.

DAFTAR PUSTAKA

- 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>.
- Oracle Corporation. (n.d.). *MySQL Documentation*. Diakses pada 18 November 2024, dari <https://dev.mysql.com/doc/>
- Hafiz. (2022). Pentingnya Sistem Keamanan Database untuk Melindungi Data Pribadi. *JISKA: Jurnal Sistem Informasi dan Informatika*, 1(2), 44–47.
- Wibisono, P. (2011). Cara Kerja B-Tree dan Aplikasinya. *Makalah IF2091*.
- Saputro, M. W., Basir, A., & Permatasari, E. K. (2022). Perancangan Sistem Informasi Perpustakaan Berbasis Web di SMA Negeri 1 Sirampog. *Jurnal Teknik Informatika dan Sistem Informasi (JURTISI)*, 2(2), 36-43

Fitriatun, E., & Aprilyani, F. (n.d.). Perancangan Sistem Informasi Perpustakaan Berbasis Web Pada MAN 10 Jakarta Untuk Meningkatkan Minat Belajar Siswa. *Jurnal Sistem Informasi STMIK Antar Bangsa*, X(1), 18-26.

Harumiati, N., Hariyanti, E., & Taufik. (n.d.). Perencanaan Arsitektur Teknologi Informasi: Studi Kasus Pada Perpustakaan Universitas Airlangga. *Jurnal Sistem Informasi*, 1(1), 16-20.

Mardiani, G. T., & Irmayanti, H. (Tahun). Analisis optimasi query SQL menggunakan teknik heuristic pada kasus data transaksi pelanggan yang layak mendapatkan rekomendasi produk. *Majalah Ilmiah UNIKOM*, 16(2), 133-143.

Darmanto, Eko. Analisa Optimalisasi Bahasa Sql Berdasarkan Relational Algebra Pada Kasus Rekapitulasi Mahasiswa Layak Wisuda. *Jurnal SIMETRIS*, 6(2) 405-414