



Analisis Performa Query SQL pada Basis Data Transaksi Peminjaman di Perpustakaan Menggunakan Indeks di MYSQL

Ridha Adjie Eryadi¹, Ninda Syaima Zain², Nazwa Nurhafidah³, Muhamad Hasbi⁴

¹²³⁴Program Studi Informatika, Universitas Teknologi Digital, Bandung, Indonesia

Email: ridhaadjie@digitechuniversity.ac.id, ninda20124076@digitechuniversity.ac.id,
nazwa20124105@digitechuniversity.ac.id, muhamad20124074@digitechuniversity.ac.id

Abstrak

Penelitian ini bertujuan untuk menganalisis dan membandingkan secara kuantitatif pengaruh penggunaan indeks terhadap kinerja eksekusi query Structured Query Language (SQL) pada basis data transaksi peminjaman di sistem informasi perpustakaan berbasis MySQL. Metode yang digunakan adalah pendekatan eksperimental dengan mengukur waktu eksekusi sejumlah query pencarian pada skema basis data perpustakaan, baik dalam kondisi tanpa indeks maupun dengan penerapan indeks pada kolom-kolom kunci yang sering digunakan sebagai parameter pencarian (seperti ID anggota, ID buku, dan tanggal peminjaman). Pengujian dilakukan menggunakan volume data yang representatif untuk mencerminkan kondisi nyata. Parameter utama yang dianalisis adalah waktu eksekusi (execution time) dari query SELECT yang bersifat kompleks. Hasil penelitian menunjukkan bahwa penggunaan indeks pada kolom yang relevan secara signifikan meningkatkan performa eksekusi query. Rata-rata waktu eksekusi pada basis data yang telah diindeks mengalami penurunan yang cukup besar dibandingkan dengan basis data tanpa indeks. Peningkatan kinerja ini sangat penting, terutama pada tabel transaksi peminjaman yang terus bertambah volumenya. Dengan demikian, penerapan indeks terbukti menjadi teknik optimasi basis data yang efektif dan penting dalam mempercepat proses pencarian data pada sistem informasi perpustakaan berbasis MySQL. Optimalisasi query melalui pengindeksan mampu mempercepat akses data transaksi peminjaman, meningkatkan efisiensi sistem, serta memberikan pengalaman pengguna yang lebih baik.

Kata Kunci : Indeks MySQL, Optimasi Query, Basis Data Perpustakaan, Performa SQL.

PENDAHULUAN

Sistem Informasi Perpustakaan (SIP) sangat bergantung pada kecepatan dan ketepatan dalam mengakses data transaksi peminjaman. Seiring bertambahnya jumlah anggota, koleksi buku, dan frekuensi peminjaman, tabel transaksi akan terus tumbuh dan secara perlahan dapat menurunkan performa basis data. Pada sistem yang menggunakan MySQL, penurunan performa biasanya terlihat ketika query pencarian atau pelaporan mulai memakan waktu lebih lama karena database harus melakukan full table scan untuk menemukan data yang dibutuhkan. Jika tidak diatasi, kondisi ini dapat memperlambat proses pelayanan dan mengganggu kenyamanan pengguna dalam mengakses informasi perpustakaan.

Pada era layanan perpustakaan yang semakin digital, kecepatan pemrosesan data peminjaman mulai dari pencatatan pinjam, pengembalian, hingga perpanjangan menjadi sangat penting. Hal ini mendorong perlunya mekanisme optimasi seperti indexing, terutama B-Tree Index yang digunakan secara default oleh MySQL. Dari latar belakang tersebut, muncul beberapa pertanyaan penting: bagaimana indexing dapat mempercepat proses pencarian pada tabel transaksi peminjaman, jenis indeks apa yang paling cocok untuk kolom penting seperti `member_id`, `book_id`, dan `loan_date`, serta seberapa besar pengaruhnya terhadap waktu respons query, baik pada operasi SELECT maupun JOIN.

Penelitian ini dilakukan untuk menjawab pertanyaan tersebut melalui evaluasi empiris terhadap performa eksekusi query SQL. Analisis difokuskan pada efektivitas berbagai jenis indeks (Single, Composite, dan Unique) dalam mendukung proses pencarian dan penggabungan data. Hasil yang diperoleh diharapkan

..

dapat memberikan gambaran yang lebih jelas mengenai strategi optimasi basis data MySQL yang dapat diterapkan oleh pengembang maupun administrator sistem perpustakaan digital.

Manfaat penelitian ini tidak hanya bersifat teknis, tetapi juga praktis. Bagi perpustakaan dan pustakawan, penelitian ini dapat menjadi panduan dalam meningkatkan kinerja sistem secara langsung mulai dari proses layanan hingga pelaporan rutin. Bagi akademisi dan peneliti, hasil penelitian ini dapat memperkaya kajian mengenai efektivitas B-Tree indexing pada skema basis data transaksi di lingkungan Library Information System. Selain itu, penelitian ini juga memberikan wawasan bagi praktisi TI, terutama dalam memilih kolom yang tepat untuk diindeks, membaca hasil analisis EXPLAIN, serta memahami trade-off antara kecepatan baca dan tulis pada sistem transaksi.

Dengan semakin meningkatnya jumlah pengguna dan digitalisasi koleksi, tabel transaksi peminjaman dapat bertambah hingga jutaan baris. Jika tidak dioptimalkan, operasi sederhana pun berpotensi berubah menjadi proses penuh yang lambat dan membebani server. Indexing di MySQL berfungsi seperti daftar isi yang memudahkan mesin database menemukan baris tertentu tanpa harus memeriksa seluruh tabel. Karena alasan inilah penelitian ini berfokus pada analisis mendalam mengenai bagaimana penerapan indeks yang tepat dapat meminimalkan masalah performa tersebut.

Secara keseluruhan, penelitian ini bertujuan untuk membuktikan secara kuantitatif sejauh mana pengindeksan pada kolom-kolom kunci mampu meningkatkan kecepatan eksekusi query pada basis data transaksi peminjaman. Perbandingan dilakukan antara kondisi tanpa indeks dan dengan indeks, terutama untuk operasi SELECT, sehingga hasilnya dapat menunjukkan seberapa besar peningkatan performa yang diperoleh serta memberikan dasar pertimbangan yang kuat dalam pengelolaan basis data perpustakaan berbasis MySQL.

METODE

Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kualitatif dan kuantitatif dengan dua metode utama, yaitu studi literatur dan eksperimen.

Studi literatur dilakukan untuk memperoleh pemahaman teoretis mengenai sistem perpustakaan berbasis PHP–MySQL, implementasi SLiMS 9.6.1 Bulian, serta konsep indexing pada MySQL. Sementara itu, metode eksperimen dilakukan untuk menguji secara langsung pengaruh penggunaan indeks terhadap kecepatan eksekusi query pada tabel peminjaman.

Pendekatan ini dipilih agar hasil penelitian tidak hanya bersifat konseptual, tetapi juga didukung oleh data empiris dari hasil pengujian sistem.

Sumber Data

- Sumber data yang digunakan dalam penelitian ini meliputi:
- Artikel jurnal yang membahas sistem perpustakaan SLiMS 9.6.1 Bulian.
- Jurnal terkait sistem perpustakaan berbasis PHP dan MySQL.
- Dataset transaksi peminjaman buku, baik berupa data riil maupun data simulasi yang disesuaikan dengan struktur sistem perpustakaan.

Dokumentasi resmi MySQL, khususnya yang berkaitan dengan mekanisme indexing dan penggunaan perintah EXPLAIN untuk analisis query.

Teknik Pengumpulan Data

Pengumpulan data dilakukan melalui dua teknik, yaitu:

- a. Studi dokumentasi, dengan menganalisis jurnal dan referensi untuk mengetahui desain sistem, struktur basis data, serta praktik optimasi query pada sistem perpustakaan.
- b. Pengujian sistem, yaitu dengan menjalankan sejumlah query pada database perpustakaan yang mencakup:
 - pencarian data peminjaman,
 - pengecekan status ketersediaan buku,
 - query JOIN antara tabel buku, anggota, dan peminjaman,
 - pembuatan laporan bulanan berdasarkan tanggal peminjaman. Seluruh query yang diuji diukur waktu eksekusinya sebelum dan sesudah penerapan indeks.

Tahapan Penelitian

Tahapan penelitian ini dilakukan secara sistematis sebagai berikut:

- Membuat atau menggunakan struktur tabel database perpustakaan yang mengacu pada jurnal sistem perpustakaan berbasis PHP–MySQL.
- Mengisi tabel peminjaman dengan data dalam jumlah besar untuk mensimulasikan kondisi penggunaan nyata.
- Menjalankan query tanpa menggunakan indeks dan mencatat waktu eksekusinya.
- Menambahkan indeks pada kolom-kolom yang sering digunakan dalam proses pencarian dan JOIN.
- Menjalankan kembali query yang sama setelah penerapan indeks dan mencatat waktu eksekusinya.
- Membandingkan hasil waktu eksekusi query sebelum dan sesudah penerapan indexing.
- Menarik kesimpulan mengenai efektivitas penggunaan indeks dalam meningkatkan performa query.

Teknik Analisis Data

Analisis data dilakukan dengan cara membandingkan hasil pengujian sebelum dan sesudah penerapan indeks.

Teknik analisis meliputi:

- Perbandingan waktu eksekusi query sebelum dan sesudah indexing.
- Analisis hasil perintah EXPLAIN untuk melihat perubahan metode akses data, seperti peralihan dari full table scan ke index scan.
- Evaluasi efektivitas setiap indeks yang diterapkan terhadap performa query.
- Penarikan kesimpulan mengenai tingkat efisiensi sistem database setelah optimasi dilakukan.

HASIL DAN PEMBAHASAN

Perancangan adalah langkah penting dalam membangun sebuah basis data, karena struktur yang dibuat harus mampu memenuhi kebutuhan saat ini sekaligus tetap relevan di masa mendatang. Untuk mencapai itu, diperlukan perancangan yang baik, baik dari sisi konsep maupun fisiknya. Pada tahap konseptual, perancang menentukan entitas apa saja yang terlibat beserta relasinya. Secara umum, proses perancangan basis data dibagi menjadi tiga tahap utama: perancangan konseptual, perancangan logikal, dan perancangan fisik.

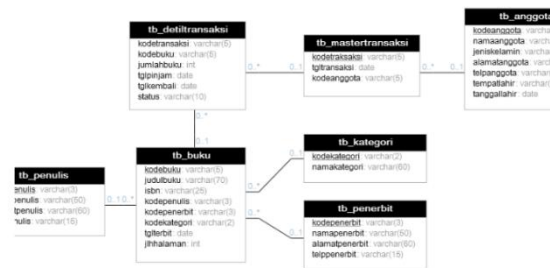
Conceptual Database Design

Conceptual database design adalah proses membangun sebuah model data yang akan digunakan di dalam suatu perusahaan, yang bersifat independent dari pertimbangan fisik. Tahap desain konseptual yang dimulai dengan membuat model data dari perusahaan dengan rincian implementasi seperti target DBMS (Database Management System), program aplikasi, bahasa pemrograman, hardware platform, performance dan segala pertimbangan fisik lainnya.

Tabel 1. Identifikasi Tipe Entitas

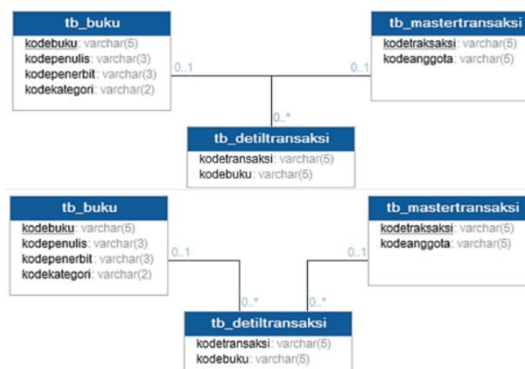
NO	Nama Entitas	Keterangan Entitas	Kegiatan
1	Kategori Buku	Merupakan Entitas yang berisi informasi mengenai data kategori dari suatu buku.	Pengelompokan data buku berdasarkan kategori dan satu kategori buku bisa terdiri dari beberapa buku.
2	Penerbit Buku	Merupakan Entitas yang berisi informasi mengenai data penerbit buku.	Pengelompokan data buku berdasarkan penerbit dan satu penerbit buku bisa terdiri dari beberapa buku.
3	Penulis Buku	Merupakan Entitas yang berisi informasi mengenai data penulis buku.	Pengelompokan data buku berdasarkan penulis dan satu penulis buku bisa terdiri dari beberapa buku.
4	Buku	Merupakan Entitas yang berisi informasi mengenai data buku.	Setiap buku dapat dipinjamkan kepada satu atau beberapa anggota.
5	Anggota	Merupakan Entitas yang berisi informasi mengenai data anggota.	Setiap buku dapat dipinjamkan kepada satu atau beberapa anggota.
6	Master Transaksi	Merupakan Entitas yang berisi informasi mengenai data transaksi peminjaman dan pengembalian.	Dapat menyimpan hanya satu jenis data transaksi.
7	Detail Transaksi	Merupakan Entitas yang berisi informasi mengenai data transaksi peminjaman dan pengembalian secara detail.	Satu transaksi bisa terdiri dari satu buku atau beberapa buku.

Gambar.1 Erd Konseptual



Logikal Database Design

Perancangan basis data logikal adalah suatu proses membangun sebuah model dari informasi yang digunakan di perusahaan berdasarkan model data yang spesifik. Tujuannya untuk membangun data logikal dari model konseptual dengan memvalidasi model perusahaan tersebut dengan benar dan memastikan mendukung transaksi yang diperlukan. Sistem basis data relasional yang akan digunakan, ada yang tidak bisa diimplementasikan oleh sebab karena ada yang perlu dimodifikasi yaitu menghilangkan bagian yang tidak kompatibel dari model data konseptual. Seperti menghilangkan relasi many to many. Untuk menghilangkan tipe hubungan yang mengandung many to many, dipecah dengan mengidentifikasi sebuah entitas baru dan mengganti hubungannya dengan one to many.



GAMBAR.2 HUBUNGAN BUKU DENGAN TRANSAKSI

Validasi relasi-relasi dengan menggunakan normalisasi untuk meminimalkan ada terjadinya data rangkap, menghindari data yang tidak konsisten terutama bila dilakukan penambahan atau penghapusan data. untuk menjamin bahwa identitas tabel secara tunggal sebagai determinan semua atribut. Berikut ini adalah normalisasi basis data perpustakaan:

Bentuk Unnormal

Pada tahap unnormal, seluruh data masih digabung dalam satu kelompok tanpa pemisahan entitas. Atribut-atribut yang terdapat dalam bentuk unnormalized ini meliputi:

{KodeBuku, JudulBuku, ISBN, TanggalTerbit, JumlahHalaman, NamaKategori, NamaPenerbit, AlamatPenerbit, TeleponPenerbit, NamaPenulis, AlamatPenulis, TeleponPenulis, NamaAnggota, JenisKelamin, AlamatAnggota, TeleponAnggota, TempatLahir, TanggalLahir, TanggalTransaksi, TanggalPinjam, TanggalKembali, JumlahBuku, Status}

Bentuk Normal Pertama

Langkah berikutnya adalah dengan cara memisahkan atribut-atribut yang nilainya sama akan ditulis hanya satu kali.

Tabel buku

{*Kodebuku, judulbuku, isbn, tglterbit,halaman,namakategori, namapenerbit,alamatpenerbit, telppenerbit, namapenulis, alamatpenulis, telppenulis, namaanggota, jeniskelamin, alamatanggota, telpanggota, tempatlahir, tanggallahir} Tabel transaksi {tgltransaksi, tglpinjam, tglkembali, jumlahbuku, status}

Bentuk Normal Kedua

Langkah selanjutnya adalah dengan cara menentukan ketergantungan fungsional.

Tabelbuku{*Kodebuku,judulbuku,isbn,tglterbit,jlhhalaman,m**kodekategori,**kodepenerbit, **kodepenulis}

Tabelkategori{*kodekategori, namakategori}

Tabelpenerbit{*kodepenerbit, namapenerbit,alamatpenerbit, telppenerbit}

Tabelpenulis{*kodepenulis,namapenulis, alamatpenulis, telppenulis}

Tabelanggota{*kodeanggota,namaanggota,jeniskelamin,alamatanggota,telpanggota,tempatlahir, tanggallahir}

Tabelmastertransaksi {kodetransaksi, tgltransaksi, kodeanggota}

Tabel detiltransaksi (kodetransaksi, kodebuku, tglpinjam, tglkembali, jumlahbuku,status}

A. Physical Database Design

Perancangan database secara fisik merupakan,tahapan untuk mengimplementasikan hasil dari database secara logis. Dapat disimpulkan bahwa proses perancangan fisik merupakan transformasi dari perancangan logis terhadap jenis DBMS yang digunakan sehingga dapat disimpan secara fisik pada media penyimpanan. My Structured Query Language (MySQL) merupakan pilihan DBMS yang tepat untuk mendukung aplikasi basis data yang dapat dilakukan sebelum menuju desain logical asalkan terdapat cukup informasi mengenai kebutuhan sistem.

Tabel 2. Kategori

No	Nama Field	Type	Size	Keterangan
1	<u>Kodekategori</u>	VARCHAR	2	Kode Kategori
2	<u>Namakategori</u>	VARCHAR	6	Nama Kategori

Tabel 3. Tabel Penerbit

No	Nama Field	Type	Size	Keterangan
1	<u>Kodepenerbit</u>	VARCHAR	2	Kode penerbit
2	<u>Namapenerbit</u>	VARCHAR	50	Nama penerbit
3	<u>Alamatpenerbit</u>	VARCHAR	60	Alamat penerbit
4	<u>Telppenerbit</u>	VARCHAR	15	Telp penerbit

Tabel 4. Tabel Penulis

No	Nama Field	Type	Size	Keterangan
1	<u>Kodepenulis</u>	VARCHAR	3	Kode penulis
2	<u>Namapenulis</u>	VARCHAR	50	Nama penulis
3	<u>Alamatpenulis</u>	VARCHAR	60	Alamat penulis
4	<u>Telppenulis</u>	VARCHAR	15	Telp penulis

Tabel 5. Tabel Buku

No	Nama Field	Type	Size	Keterangan
1	<u>Kodebuku</u>	VARCHAR	5	
2	<u>Jumlahbuku</u>	VARCHAR	70	
3	<u>Isbn</u>	VARCHAR	25	
4	<u>Kodepenulis</u>	VARCHAR	3	
5	<u>Kodepenerbit</u>	VARCHAR	3	
6	<u>Kodekategori</u>	VARCHAR	2	
7	<u>Tglterbit</u>	Date		
8	<u>Jlhhalaman</u>	Integer		

Tabel 6. Tabel Anggota

No	Nama Field	Type	Size	Keterangan
1	Kodeanggota	VARCHAR	5	
2	Namaanggota	VARCHAR	2	
3	Jemiskelamin	VARCHAR	5	
4	Alamatanggota	VARCHAR	70	
5	Telpanggota	VARCHAR	2	
6	Tempatlahir	DOUBLE		
7	Tanggalahir	DOUBLE		
8	Banyak	INT		
9	Kodesupplier	VARCHAR	4	

Tabel 7. Tabel Master Transaksi

No	Nama Field	Type	Size	Keterangan
1	Kode_transaksi	VARCHAR	5	
2	Tgl_transaksi	DATE		
3	Kodeanggota	VARCHAR	5	

Tabel 8 . Tabel Detail Transaksi

No	Nama Field	Type	Size	Keterangan
1	Kode_transaksi	VARCHAR	5	
2	Kodebuku	VARCHAR	5	
3	Tglpinjam	Date	10	
4	Tglkembali	Date	10	
5	Jumlahbuku	Int	2	
6	Status	VARCHAR	10	

Data Definiton Language Adalah Sebuah Bahasa Yang Digunakan Untuk Mendefinisikan Pendefinisian Data. Terdiri Dari Perintah-Perintah Untuk Membentuk, Mengubah Atau Menghapus Tabel Beserta Kolom-Kolom Dan Type Data Penyusunnya, Serta Perintah-Perintah Untuk Menetapkan Hubungan Dan Batasan-Batasan Data. DDL Juga Kumpulan Perintah SQL Yang Digunakan Untuk Membuat (Create), Mengubah (Alter) Dan Menghapus (Drop) Struktur Dan Definisi Tipe Data Dari Objek-Objek Database.

A. Sql Create Table Kategori

```
CREATE TABLE `perpustakaan_db`.`tb_kategori` ( `kodekategori` varchar(2) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL, `namakategori` varchar(60) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, PRIMARY KEY (`kodekategori`) ) ENGINE=InnoDB DEFAULT CHARACTER SET=latin1 COLLATE=latin1_swedish_ci ;
```

B. Sql Create Table Penerbit

```
CREATE TABLE `perpustakaan_db`.`tb_penerbit` ( `kodepenerbit` varchar(3) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL, `namapenerbit` varchar(50) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `alamatpenerbit` varchar(60) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `telpenerbit` varchar(15) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, PRIMARY KEY (`kodepenerbit`) ) ENGINE=InnoDB DEFAULT CHARACTER SET=latin1 COLLATE=latin1_swedish_ci ;
```

C. Sql Create Table Penulis

```
CREATE TABLE `perpustakaan_db`.`tb_penulis` ( `kodepenulis` varchar(3) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL, `namapenulis` varchar(50) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `alamatpenulis` varchar(60) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `telpenulis` varchar(15) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, PRIMARY KEY (`kodepenulis`) ) ENGINE=InnoDB DEFAULT CHARACTER SET=latin1 COLLATE=latin1_swedish_ci ;
```

D. Sql Create Table Buku

```
CREATE TABLE `perpustakaan_db`.`tb_buku` ( `kodebuku` varchar(5) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL, `judulbuku` varchar(70) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `isbn` varchar(25) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `kodepenulis` varchar(3) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `kodepenerbit` varchar(3) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `kodekategori` varchar(2) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `tglterbit` date NULL DEFAULT NULL, `jhlhelaman` int(3) NULL DEFAULT NULL, PRIMARY KEY (`kodebuku`), CONSTRAINT `tb_buku_ibfk_1` FOREIGN KEY (`kodepenulis`) REFERENCES `perpustakaan_db`.`tb_penulis` (`kodepenulis`), CONSTRAINT `tb_buku_ibfk_2` FOREIGN KEY (`kodepenerbit`) REFERENCES `perpustakaan_db`.`tb_penerbit` (`kodepenerbit`), CONSTRAINT `tb_buku_ibfk_3` FOREIGN KEY (`kodekategori`) REFERENCES `perpustakaan_db`.`tb_kategori` (`kodekategori`), INDEX `kodepenulis` (`kodepenulis`), INDEX `kodepenerbit` (`kodepenerbit`), INDEX `kodekategori` (`kodekategori`) ) ENGINE=InnoDB DEFAULT CHARACTER SET=latin1 COLLATE=latin1_swedish_ci ;
```

E. Sql Create Table Anggota

```
CREATE TABLE `perpustakaan_db`.`tb_anggota` ( `kodeanggota` varchar(5) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL, `namaanggota` varchar(50) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `jenis kelamin` varchar(15) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `alamatanggota` varchar(60) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `teleponanggota` varchar(15) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `tempatlahir` varchar(20) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `tanggalahir` date NULL DEFAULT NULL, PRIMARY KEY (`kodeanggota`) )ENGINE=InnoDB DEFAULT CHARACTER SET=latin1 COLLATE=latin1_swedish_ci;
```

F. Sql Create Detail Transaksi

```
CREATE TABLE `perpustakaan_db`.`tb_detailtransaksi` ( `kode_transaksi` varchar(5) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `kodebuku` varchar(5) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, `tglpinjam` date NULL DEFAULT NULL, `tglkembali` date NULL DEFAULT NULL, `jumlahbuku` int(2) NULL DEFAULT NULL, `status` varchar(10) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, CONSTRAINT `tb_detailtransaksi_ibfk_1` FOREIGN KEY (`kode_transaksi`) REFERENCES `perpustakaan_db`.`tb_mastertransaksi` (`kode_transaksi`), CONSTRAINT `tb_detailtransaksi_ibfk_2` FOREIGN KEY (`kodebuku`) REFERENCES `perpustakaan_db`.`tb_buku` (`kodebuku`), INDEX `kode_transaksi` (`kode_transaksi`), INDEX `kodebuku` (`kodebuku`) )ENGINE=InnoDB DEFAULT CHARACTER SET=latin1 COLLATE=latin1_swedish_ci ;
```

G. Sql Create Master Transaksi

```
CREATE TABLE `perpustakaan_db`.`tb_mastertransaksi` ( `kode_transaksi` varchar(5) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL, `tgl_transaksi` date NULL DEFAULT NULL, `kodeanggota` varchar(5) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, PRIMARY KEY (`kode_transaksi`), CONSTRAINT `tb_mastertransaksi_ibfk_1` FOREIGN KEY (`kodeanggota`) REFERENCES `perpustakaan_db`.`tb_anggota` (`kodeanggota`), INDEX `kodeanggota` (`kodeanggota`) )ENGINE=InnoDB DEFAULT CHARACTER SET=latin1 COLLATE=latin1_swedish_ci ;
```

Hasil dari perancangan basis data ini, dihasilkan 7 tabel basis data dan dimasukkan ke dalam basis data yang bernama perpustakaan dan pengimplementasian dari rancangan basis data yang terdiri atas 7 tabel yang ada dilakukan dengan menggunakan aplikasi Xampp dengan koneksi server localhost. Perancangan basis data yang mengacu kepada model data relasional khususnya basis data perpustakaan dimaksudkan agar dalam setiap tabel yang terdapat didalam database perpustakaan saling memiliki keterkaitan demi menjamin integritas data. Selain itu, model data relasional akan memberikan gambaran yang jelas dan memberikan kemudahan bagi programmer ketika ingin membangun aplikasi pengelolaan data buku perpustakaan sekolah. Keberhasilan dalam membangun sebuah aplikasi tidak terlepas dari sebuah model basis data yang baik. Inilah yang menjadi dasar utama diperlukan sebuah rancangan basis data dengan menerapkan mode relasional.

KESIMPULAN

Dari proses perancangan dan pengujian yang dilakukan, dapat disimpulkan bahwa pembuatan basis data yang rapi dan terstruktur sangat berpengaruh terhadap kinerja Sistem Informasi Perpustakaan. Dengan membagi data ke dalam tabel-tabel yang jelas, melakukan normalisasi, dan menentukan relasi antar entitas, sistem menjadi lebih mudah dikelola dan tidak menimbulkan data ganda atau konflik.

Hasil perancangan menghasilkan tujuh tabel utama yang saling terhubung dan siap digunakan dalam aplikasi perpustakaan. Desain ini membuat proses seperti pencarian buku, mencatat peminjaman, ataupun mengelola anggota bisa berjalan lebih cepat dan lebih teratur.

Selain itu, penggunaan indeks pada kolom penting terbukti dapat mempercepat proses pencarian data, terutama pada tabel transaksi yang jumlahnya paling cepat bertambah. Dengan struktur yang sudah dirancang dengan baik dan didukung MySQL sebagai DBMS, sistem perpustakaan dapat bekerja lebih stabil, efisien, dan siap dikembangkan lebih lanjut.

Singkatnya, desain basis data yang solid adalah pondasi utama agar aplikasi perpustakaan dapat bekerja dengan benar, cepat, dan mudah dipelihara ke depannya.

DAFTAR PUSTAKA

- Aksi.co. (2024, October 14). Contoh database peminjaman buku perpustakaan 2024. <https://aksi.co/2024/10/14/contoh-database-peminjaman-buku-perpustakaan-2024/>
- Pamungkas, R. (2018). Optimalisasi query dalam basis data MySQL menggunakan index. ResearchGate. https://www.researchgate.net/publication/325530763_Optimalisasi_Query_Dalam_Basis_Data_My_Sql_Menggunakan_Index
- Sanjaya, R. (2019). Implementasi data warehouse untuk analisis peminjaman buku pada perpustakaan. ResearchGate. https://www.researchgate.net/publication/332670429_Implementasi_Data_Warehouse_untuk_Analisis_Peminjaman_Buku_pada_Perpustakaan

- Setiawan, A. (2019). Perancangan basis data perpustakaan sekolah dengan menerapkan model data relasional. ResearchGate.
https://www.researchgate.net/publication/339658864_Perancangan_Basis_Data_Perputakaan_Sekolah_dengan_Menerapkan_Model_Data_Relasional
- SLiMS Development Team. (n.d.). Optimalisasi pelayanan dan efisiensi perpustakaan melalui penerapan SLiMS 9.6.1 Bulian. ResearchGate.
https://www.researchgate.net/publication/391145949_OPTIMALISASI_PELAYANAN_DAN_EFISIENSI_PERPUSTAKAAN_MELALUI_PENERAPAN_SLIMS_961_BULIAN
- Bulian, S. (n.d.). Analisis dan desain sistem informasi manajemen perpustakaan menggunakan program PHP dan database MySQL. <https://ejournal.stikom22januarikendari.ac.id/JurnalFinish/Analisis%20dan%20Desain%20Sistem%20Informasi%20Manajemen%20Perpustakaan%20Menggunakan%20Program%20PHP%20dan%20Database%20MySQL.pdf>