

Laporan Tugas

Implementasi Mysql pada studi kasus pada suatu perusahaan



Dipersiapkan oleh :

Muhammad Ariq Faridzki

2242004

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER
"AMIK BANDUNG"

2023

Daftar Isi

Pendahuluan.....	3
Tujuan.....	3
Jurnal Acuan.....	3
Studi Kasus.....	4
Masalah.....	4
Solusi.....	4
Implementasi.....	5
Testing.....	7
Kesimpulan.....	10
Referensi.....	11

Pendahuluan

Basis data (Database) adalah suatu kumpulan informasi yang terstruktur yang biasanya disimpan dalam sistem komputer, basis data itu sendiri dikendalikan oleh *Database Management system (DBMS)* dan sebagian basis data menggunakan bahasa SQL untuk mengatur basis data.

Mysql adalah *software Relational Database Management System (RDBMS)* open source yang sangat terkenal di bawah lisensi *General Public License (GPL)* untuk model basis data ini menggunakan struktur tabel.

Zaman ini data semakin banyak diproduksi setiap harinya sekitar **328.77 juta** terabytes (Duarte, 2023) dan semakin banyak juga yang menggunakan database untuk mengatur datanya oleh karena itu tujuan dari laporan ini adalah untuk memahami implementasi basis data menggunakan Mysql agar mengerti bagaimana manfaatnya dalam menggunakan basis data.

Tujuan

- Memahami penggunaan basis data dalam studi kasus
- Memahami struktur tabel dan hubungan antar tabel
- Memahami manfaat penggunaan basis data

Jurnal Acuan

Implementasi Metode Indexing dan Penggunaan Subquery untuk Optimalisasi Database Rawat Jalan Rumah Sakit Menggunakan Mysql

Penulis :

- Wilsen Grivin Mokodaser (Magister Ilmu Komputer)
- Monica Dwijayanti (Fakultas Teknologi Informasi)
- Samidi (Universitas Budi Luhur)

Studi Kasus

Masalah

Sejak awal tahun 2020, wabah covid yang meningkat membuat orang membutuhkan akses pada layanan rumah sakit. Di Indonesia proses penerimaan pasien di bagian poliklinik rawat jalan masyarakat masih menggunakan sistem pendaftaran manual (David Laksamana Caesar, 2019). Oleh karena itu banyak kendala seperti di antrian, pendaftaran, konsultasi dokter, bagian pembayaran dan juga pemeriksaan lanjut seperti ruangan radiologi atau laboratorium. Proses rata-rata waktu dari mengambil nomor antrian sampai pendaftaran adalah 15,02 menit untuk satu pasien.

Dengan adanya basis data maka beralihnya dari metode manual ke penggunaan sistem basis data tetapi dikarenakan data tersebut meningkat setiap saat maka bisa menurun performanya karena query yang belum optimal.

Solusi

Oleh karena itu solusi yang diterapkan oleh penulis adalah :

1. Mengimplementasikan metode basis data
2. Mengoptimasi query basis data menggunakan metode indexing dan Subquery dalam basis data MySQL

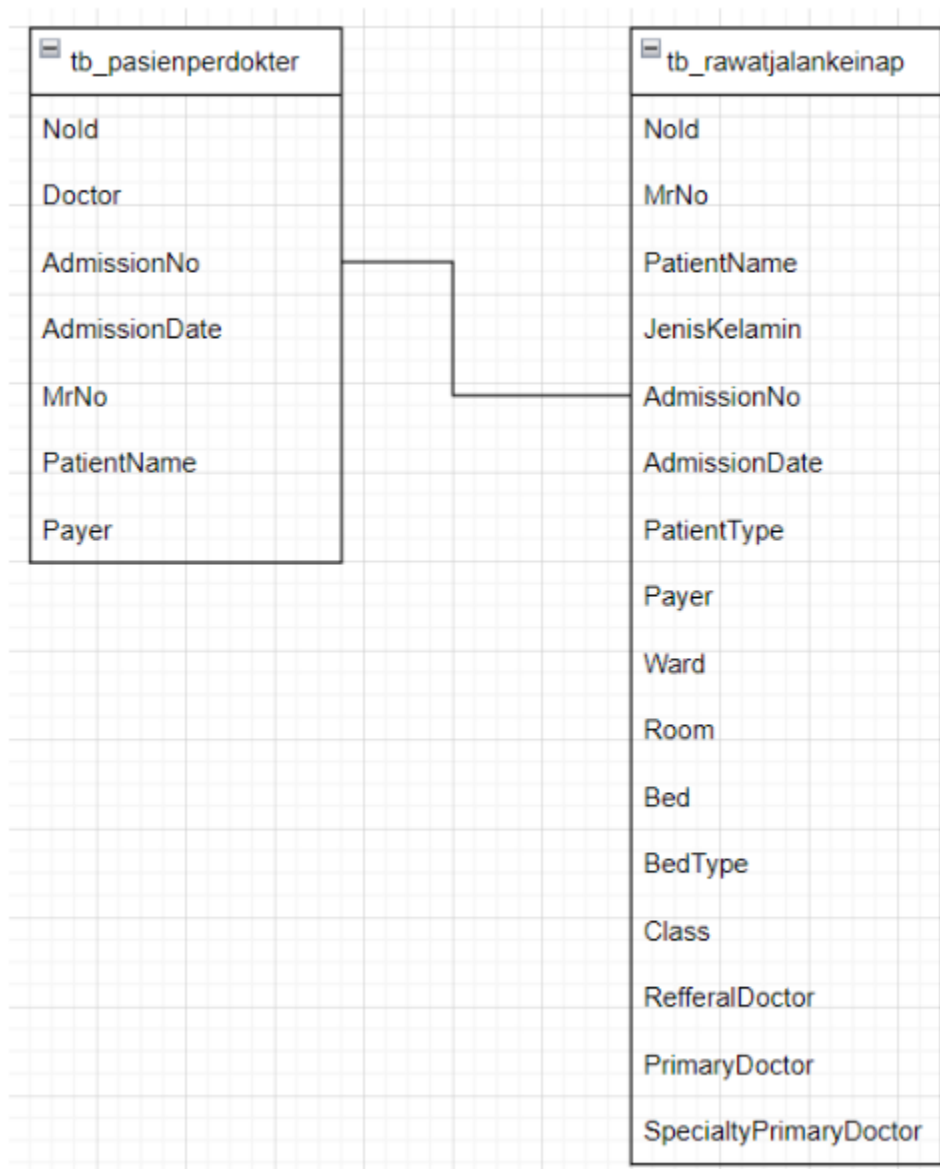
Metode indexing digunakan untuk mempercepat operasi mengambil data dengan diatur sedemikian rupa untuk membuat struktur data (Trivedi, 2023) dengan ini basis data dapat menemukan data dengan cepat tanpa harus menelusuri seluruh data dalam tabel

Implementasi

Penulis mengambil data pendukung yaitu :

- Pasien per dokter tahun 2019 - 2022
- Pasien rawat jalan tahun 2019 - 2022

Setelah itu penulis mengkonversi data dalam bentuk .CSV dan di *import* ke basis data MySQL yang menghasilkan tabel `tb_pasienperdokter` dan `tb_rawatjalankeinap`



Gambar 1 (Mokodaser et al.) Struktur Tabel

Dari gambar tersebut saya menyimpulkan bahwa ada hubungan relasi dari *tb_pasienperdokter* dan *tb_rawatjalankeinap* yang dari kolom *AdmissionNo* sebagai referensi data.

Dan penulis juga menduplikat tabel tersebut dan menambahkan *index* dalam tabelnya dengan cara

```
CREATE INDEX index_rawatjalankeinap ON index_rawatjalankeinap (AdmissionDate, AdmissionNo, Bed, BedType, Class, JenisKelanin, MrNo, NoId, PatientName, PatientType, Payer, PrimaryDoctor, ReferralDoctor, Room, SpecialtyPrimaryDoctor,Ward) USING BTREE;
```

```
CREATE INDEX index_pasienperdokter ON index_pasienperdokter (AdmissionDate, AdmissionNo, Doctor, MrNo, NoId, PatientName, Payer) USING BTREE;
```

Gambar 2 (Mokodaser et al.) Implementasi membuat tabel

dan berikut hasil implementasinya

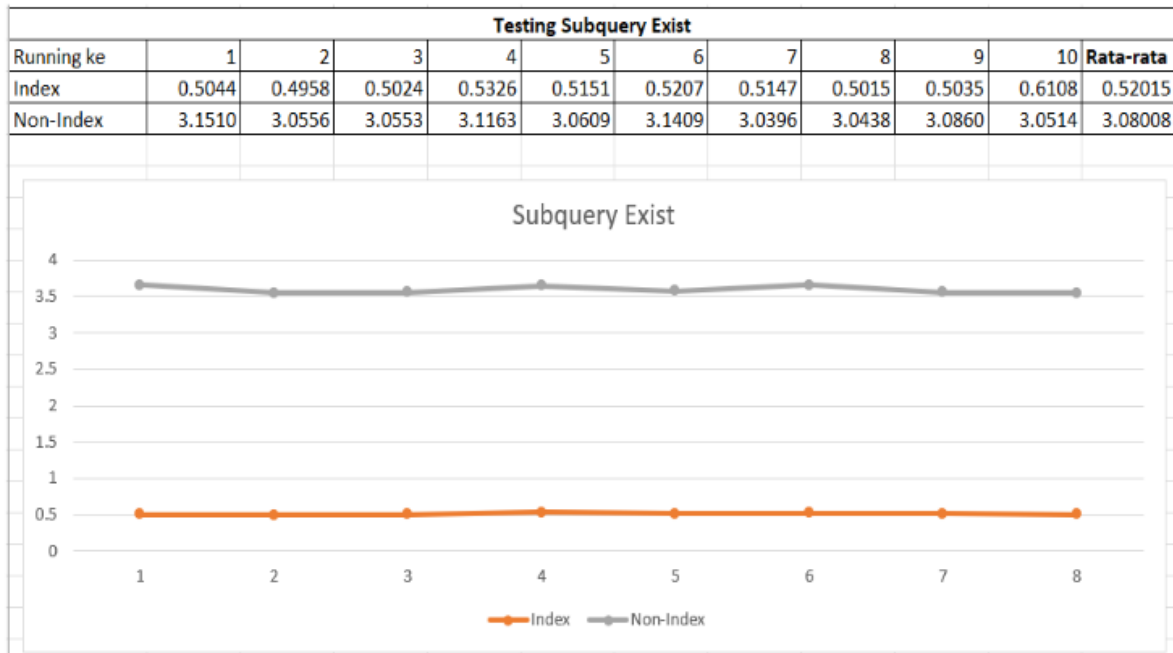
Table	Action	Rows
index_pasienperdokter	Browse Structure Search Insert Empty Drop	222956
index_rawatjalankeinap	Browse Structure Search Insert Empty Drop	38551
tb_pasienperdokter	Browse Structure Search Insert Empty Drop	222956
tb_rawatjalankeinap	Browse Structure Search Insert Empty Drop	38551
4 table(s)	Sum	523.014

Gambar 3 (Mokodaser et al.) List Tabel

Testing

Penulis mengetes kinerja basis data Mysql dengan membandingkan tabel yang telah terindex dan yang belum dengan cara mengambil data spesifik dengan menggabungkan (JOIN) tabel *tb_pasienperdokter* dan *tb_rawatjalankeinap*

Berikut gambar dibawah sebagai hasil yang diperoleh



Gambar 4 (Mokodaser et al.) hasil query menggunakan EXIST

Dan berikut query yang di jalankan oleh penulis

Query EXIST dengan tabel non-index:

```
SELECT
    a.Doctor,
    a.AdmissionDate,
    a.MrNo,
    b.PatientName, b.AdmissionDate, b.Class, b.Payer, b.Room
FROM
    tb_pasienperdokter a
LEFT JOIN
    tb_rawatjalankeinap b
ON
    a.AdmissionNo > 'OPA1908010522'
WHERE EXISTS (
    SELECT AdmissionNo FROM tb_rawatjalankeinap
    WHERE AdmissionDate
    BETWEEN '17-Jan-19' AND '25-Feb-19' AND a.AdmissionNo =
    b.AdmissionNo );
```


Query EXIST dengan tabel di index:

```
SELECT
    a.Doctor,a.AdmissionDate,a.MrNo,
    b.PatientName,
    b.AdmissionDate,
    b.Class,
    b.Payer,
    b.Room
FROM
    index_pasienperdokter a
LEFT JOIN
    index_rawatjalankeinap b
ON
    a.AdmissionNo > 'OPA1908010522'
WHERE
    EXISTS (
        SELECT AdmissionNo FROM index_rawatjalankeinap
        WHERE AdmissionDate BETWEEN '17-Jan-19' AND '25-Feb-19'
        AND a.AdmissionNo = b.AdmissionNo );
```

Penjelasan implementasi:

- Bagian *SELECT* digunakan untuk mengambil data dari kolom yang ada dalam tabel yang sudah dibuat
- Bagian *FROM* untuk target tabel yang ingin datanya diambil
- Bagian *LEFT JOIN* adalah suatu teknik untuk menggabungkan kedua tabel
- Bagian *ON* digunakan untuk mereferensikan data yang cocok agar saat penggabungan data yang cocok akan diambil
- Bagian *WHERE* digunakan untuk sebagai kriteria apa yang mau data di ambil
- Bagian *EXIST* adalah untuk mengecek apakah tabel yang diambil ada data walaupun hanya 1 yang keluar, bagian ini juga merupakan subquery yang dimana ada query dalam query
- Bagian *BETWEEN* untuk menentukan kriteria dengan jarak yang diinginkan yang biasanya masuk pada bagian *WHERE*

Kesimpulan

Setelah mempelajari jurnal penulis saya mulai memahami implementasi penggunaan basis data dengan studi kasus seperti rumah sakit walaupun hanya 2 tabel, meskipun begitu data yang dianalisa sebanyak 222 ribu dan 38 ribu rows dengan waktu rata - rata 0,5 detik setelah di index, saya juga yakin dengan performa kinerja seperti ini mampu untuk menopang banyak data kedepannya dan meningkatkan kualitas layanan

Saya juga mulai memahami kenapa basis data MySQL sangat sering dipakai oleh perusahaan besar. Walaupun begitu metode indexing ini sangat bagus untuk penggunaan mengambil data akan tetapi akan mempengaruhi kinerja INSERT, UPDATE dan DELETE dikarenakan data yang diubah mengharuskan index tersebut untuk diubah juga. (Vileikis)

Walaupun begitu dalam contoh studi kasus ini sangat cocok untuk penggunaan index dikarenakan sangat sering untuk mengambil data untuk menganalisa pasien.

Referensi

- Duarte, Fabio. "Amount of Data Created Daily (2023)." *Exploding Topics*, Fabio Duarte, 3 April 2023, <https://explodingtopics.com/blog/data-generated-per-day>. Accessed 29 September 2023.
- Mokodaser, Wilsen Grivin, et al. "Implementasi Metode Indexing dan Penggunaan Subquery untuk Optimalisasi Database Rawat Jalan Rumah Sakit Menggunakan Mysql | CogITO Smart Journal." *CogITO Smart Journal*, 27 December 2022, <https://cogito.unklab.ac.id/index.php/cogito/article/view/415>. Accessed 30 September 2023.
- Trivedi, Aditya. "Indexing in MySQL." *Scaler*, Senin May 2023, <https://www.scaler.com/topics/indexing-in-mysql/>. Accessed 30 September 2023.
- Vileikis, Lukas. "A Guide to MySQL Indexes." *Severalnines*, Severalnines, 18 September 2020, <https://severalnines.com/blog/guide-mysql-indexes/>. Accessed 30 September 2023.