**DATABASE MANAGEMENT SYSTEM MARIADB**





Disusun oleh:

Arief Budhiman (1301154360)

Fadila Zain (1301150444)

Firdausi Aufa Faradhila (1301154500)

IF-39-10

|  |
| --- |
| LAPORAN TUGAS BESAR  Diajukan untuk memenuhi tugas pada mata kuliah sistem basis data |

|  |
| --- |
| PROGRAM STUDI S1 TEKNIK INFORMATIKA  FAKULTAS INFORMATIKA  UNIVERSITAS TELKOM  BANDUNG  2017 |

DAFTAR ISI

BAB I

PENDAHULUAN

1. Latar Belakang
2. Rumusan Masalah

Adapun rumusan masalah yang didapat dari latar belakang tersebut adalah :

1. Bagaimana arsitektur memori dari DBMS MariaDB?
2. Bagaimana cara membuat Query Execution Plan pada MariaDB?
3. Bagaimana cara menghitung cost dari Query Execution Plan yang sudah dibuat?
4. Tujuan

Adapun tujuan yang didapat dari rumusan masalah tersebut adalah :

1. Untuk mengetahui bagaimana arsitektur memori dari DBMS MariaDB.
2. Untuk mengetahui cara membuat Query Execution Plan pada MariaDB.
3. Untuk mengetahui cara menghitung cost dari Query Execution Plan yang sudah dibuat.
4. Pembagian Tugas

**BAB II**

**PEMBAHASAN**

1. Arsitektur DBMS

Arsitektur DBMS merupakan suatu kerangka kerja bagi pembangunan pada sistem basis data yang memiliki tujuan untuk menyediakan sarana antar muka (interface) dalam mengakses data secara efisien tanpa adanya kesulitan tentang cara data direkam maupun dipelihara. DBMS juga memiliki arsitektur untuk melakukan abstraksi dari data, sehingga dapat diperoleh independensi data program.

Arsitektur DBMS terbagi menjadi tiga level, yaitu :

1. Level Eksternal atau Level Pandangan

Level eksternal atau level pandangan merupakan cara pandang pemakai terhadap basis data yang berbeda-beda. Namun cara pandang secara eksternal hanya terbatas pada entitas, atribut, dan hubungan antar entitas yang diperlukan.

1. Level Konseptual

Level konseptual merupakan suatu kumpulan cara pandang pemakai terhadap basis data yang menggambarkan bahwa data yang disimpan didalam basis data memiliki hubungan diantara keduanya tanpa memperhatikan besarnya penyimpanan dalam ukuran byte. Hal-hal tersebut dapat digambarkan dalam tingkat konseptual, antara lain :

* Semua entitas beserta atribut dan hubungannya.
* Batasan pada data yang ada.
* Informasi semantik tentang data yang ada.
* Keamanan dan integritas informasi.

Semua cara pandang yang terdapat pada level eksternal berupa data yang dibutuhkan oleh pemakai, harus sudah tercakup didalam level konseptual atau dengan kata lain dapat diturunkan dari data yang telah ada.

1. Level Internal atau Level Fisik

Level internal atau level fisik merupakan suatu perwujudan data pada komputer yang menggambarkan suatu basis data yang disimpan secara fisik didalam peralatan penyimpanan yang berkaitan dengan tempat penyimpanan. Pada level internal atau level fisik ini, terdapat beberapa hal yang harus diperhatikan, antara lain :

* Alokasi ruang penyimpanan data dan indeks.
* Deskripsi record untuk penyimpanan dengan ukuran penyimpanan untuk data elemen.
* Penempatan record.
* Penempatan data dan teknik enskripsi.

1. Arsitektur Memory

MariaDB memiliki arsitektur memori yang terdiri dari 4 database engine, yaitu MyISAM, InnoDB, XtraDB dan Aria. Masing-masing database engine memiliki kekurangan dan kelebihannya masing-masing. MyISAM dan Aria lebih cocok digunakan untuk tipe table yang tidak sering di*update*, sedangkan InnoDB dan XtraDB sebaliknya, lebih cocok digunakan pada tipe table yang sering dilakukan *update.* XtraDB merupakan pengembangan lanjutan dari InnoDB dan Aria adalah pengembangan dari MyISAM.

* MyISAM

MyISAM merupakan jenis storage engine yang paling sering digunakan pada website-website, data warehouse ataupun pada model aplikasi lainnya yang merupakan tipe default dari storage engine MYSQL. MyISAM juga merupakan suatu tipe tabel yang sederhana, stabil dan mudah digunakan.

Kelebihan utama MyISAM adalah kecepatan dan kestabilannya. Jika kita memilih tabel MyISAM, maka MYSQL secara otomatis akan menentukan salah satu dari tiga jenis tabel MyISAM yang ada, yaitu :

1. MyISAM Static

Jenis ini memiliki sifat yang pasti dan digunakan ketika semua kolom pada tabel didefinisikan dengan ukuran yang sudah pasti atau tidak terdapat kolom yang memiliki tipe seperti varchar, text, maupun blob. Jenis ini membuat MyISAM menjadi lebih cepat, aman, dan stabil.

1. MyISAM Dynamic

Jenis ini memiliki sifat efektif dan digunakan ketika terdapat kolom dengan tipe yang dinamis, seperti varchar. Jenis ini membuat MyISAM memiliki ukuran data atau file yang menyesuaikan dengan isi dari masing-masing kolom.

1. MyISAM Compressed

Jenis ini merupakan penggabungan dari jenis MyISAM static dan dinamic yang memiliki perintah myisamchk. Jenis ini memungkinkan MyISAM memiliki hasil yan lebih kecil dari segi ukuran. Namun karena jenis ini merupakan penggabungan dari jenis MyISAM static dan dinamis, maka jenis ini tidak dapat menggunakan operasi seperti insert, update, serta delete.

MyISAM memiliki beberapa kelebihan, antara lain :

1. Sederhana untuk dirancang dan dibuat.
2. Lebih cepat dari InnoDB secara keseluruhan akibat dari struktur sederhana sehingga memiliki sedikit sumber daya server.
3. Pengindeksan teks lengkap.
4. Sangat baik untuk membaca-intensif atau select pada tabel.

Selain memiliki kelebihan, MyISAM juga memiliki kurangan, seperti :

1. Tidak terdapat integritas data seperti kendala hubungan.
2. Tidak mendukung transaksi.
3. Lebih lambat dari InnoDB untuk tabel yang sering diinsert dan diupdate, karena seluruh tabel terkunci.

* InnoDB

InnoDB merupakan jenis storage engine MYSQL yang digunakan untuk menyimpan data yang sangat besar dan lebih besar jika dibandingkan dengan MyISAM. InnoDB juga merupakan tipe tabel MYSQL yang mendukung proses transaksi.

InnoDB memiliki beberapa kelebihan, antara lain :

1. InnoDB lebih baik digunakan karena terdapat integritas data dimana integritas datanya lebih prioritas.
2. Lebih cepat dalam proses insert dan update tabel karena menggunakan tingkat penguncian baris.

Selain memiliki kelebihan, InnoDB juga memiliki kekurangan, seperti :

1. Pencipta skema harus mengambil lebih banyak waktu dalam mendesain model data yang lebih kompleks daripada MyISAM karena InnoDB harus menjaga hubungan yang berbeda antar tabel dan database administrator.
2. Mengkonsumsi sumber daya sistem yang lebih seperti halnya pada RAM.
3. Tidak adanya pengindeksan teks penuh.

Maka dapat disimpulkan bahwa MyISAM dan InnoDB memiliki beberapa perbedaan, antara lain :

1. InnoDB lebih baru dalam hal pengembangan, sementara MyISAM lebih dulu ada.
2. InnoDB lebih kompleks sedangkan MyISAM lebih sederhana.
3. InnoDB semakin ketat dalam integritas data sedangkan MyISAM lebih longgar.
4. InnoDB menerapkan penguncian tingkat baris untuk insert dan update baris, sementara MyISAM menerapkan penguncian tingkat tabel.
5. InnoDB memiliki transaksi sedangkan MyISAM tidak.
6. InnoDB memiliki kunci asing (Foreign Key) dan constraint, sedangkan MyISAM tidak.
7. InnoDB memiliki sistem recovery yang lebih baik dibandingkan dengan MyISAM.
8. MyISAM memiliki indeks pencarian teks penuh, sementara InnoDB tidak.

* XtraDB

XtraDB merupakan suatu pengembangan dari InnoDB dimana XtraDB mengandung banyak perbaikan untuk memberikan kontrol atau pengawasan yang lebih baik terhadap disk I/O. XtraDB juga memiliki beberapa perubahan dari InnoDB seperti penguncian konten dari kode dan data yang sering diakses yang memungkinkan untuk memiliki jumlah yang lebih tinggi untuk membuat perubahan pada waktu yang sama.

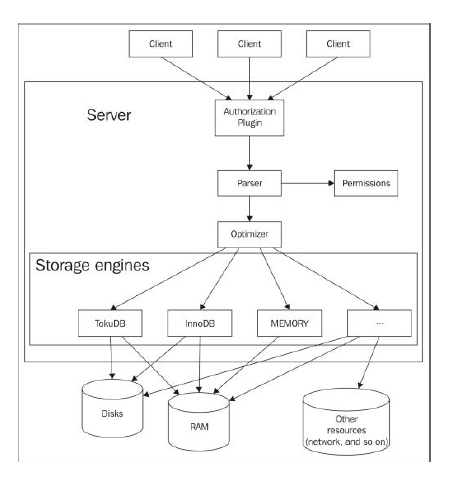
* Aria

Aria dibuat untuk menyempurnakan MyISAM yang digunakan untuk memulihkan data setelah adanya kesalahan. Data yang berubah pada Aria berupa data atomik yang dapat diterapkan seluruhnya atau pada tabel yang rusak.

Aria menggunakan format data yang berbeda yang disebut sebagai page atau halaman yang lebih cepat dan tidak pernah terjadi pemisahan yang terlalu banyak, tetapi memungkinkan untuk menggunakan format dinamik untuk menyesuaikan dengan MyISAM.

Aria dapat lebih baik dari MyISAM dalam penggunaannya ketika terjadi banyak transaksi pada saat yang bersamaan untuk mengakses data, dan MariaDB membuat Aria sebagai aplikasi baru, namun pengguna harus menyadari bahwa menulis dalam jumlah yang besar di Aria akan menjadi lambat, terlebih ketika adanya penggandaan indeks.

1. Background Process



Pada dasarnya, MariaDB menerima beberapa query SQL atau perintah, menggabungkannya, dan mengembalikan hasilnya. Lihat proses berikut dan komponen-komponen yang terlibat di dalamnya :

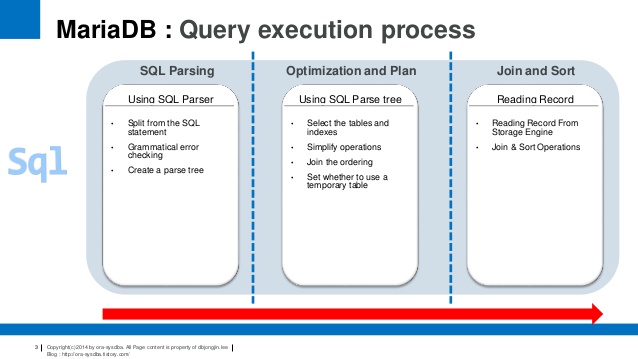
1. Ketika seorang pengguna menggunakan MariaDB, sebuah autentifikasi dilakukan berdasarkan pada hostname, username, dan password dari pengguna.
2. Jika login berhasil, pengguna dapat mengirimkan query SQL ke server.
3. Parser mengerti dan memahami string yang ada pada SQL.
4. Server memeriksa apakah pengguna memiliki izin untuk tindakan yang diminta.
5. Jika query terletak pada cache query, maka hasilnya akan dikembalikan kepada pengguna.
6. Optimizer akan mencoba untuk menemukan strategi eksekusi tercepat atau rencana query. Ini berarti optimizer memutuskan untuk mengurutkan tabel mana yang akan dibaca. Ini juga berarti memutuskan indeks mana yang akan diakses dan apakah tabel penyimpanan sementara akan digunakan. Strategi yang baik dapat sangat mengurangi akses ke disk dan mengurangi kompleksitas dari operasi yang dilakukan.
7. Media penyimpanan membaca dan menulis data dan indeks file dan beberapa cache yang bisa digunakan untuk mempercepat operasi. Beberapa fitur penting seperti transaksi dan foreign key yang diimplementasikan di tingkat media penyimpanan.
8. Struktur Files

Non partitioned Aria table terdiri dari 3 file fisik, yaitu :

1. Nama\_tabel.frm
2. Nama\_tabel.MAD
3. Nama\_tabel.MAI

File .frm (dari kata format) memuat definisi table, file ini selalu ada untuk storage engine yang menulis data ke disk. File .MAD berisi data, dan file .MAI berisi index.

1. Query Execution Process



Query execution process dari MariaDB dapat dibagi menjadi 3, yaitu:

1. SQL Parsing

SQL Parsing merupakan tahapan dimana query dipecah atau dibagi menjadi beberapa statement dengan menggunakan SQL parser, memeriksa kesalahan pada query, dan membuat parse tree dari query yang sudah dipecah (parsing).

1. Optimization and Plan

Optimization adalah tahap dimana query yang telah diproses sebelumnya dapat berjalan dengan menggunakan memori yang se-*optimal* mungkin dengan menggunakan data dari parse tree yang diperoleh dari proses sebelumnya

1. Join and Sort

Tahap terakhir adalah join dan sort. Pada tahap ini, dibaca record dari storage engine dan melakukan join dan sorting.

Setelah ketiga tahapan dilakukan, maka MariaDB akan menampilkan hasil dari query yang telah di-*input*kan.