

## BAB 2

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1. Tinjauan Pustaka

Pada Penelitian sejenis ini pernah dilakukan oleh Wasino dkk (2013); Maulani dkk (2015); Nilaliliana Prihatin (2017) ; Eka Rahmadyani(2016);dan Habimana (2015).

Penelitian yang pertama dilakukan oleh Wasino dkk (2013) dalam penelitian tersebut membandingkan optimasi *query* antara *nested join*, *hash join* dan *merge join* untuk pencarian data dengan memperhatikan waktu *response* pada database Microsoft SQL Server. Perbedaan terdapat pada penggunaan aplikasi database yang berbeda yaitu MySQL.

Penelitian yang kedua dilakukan oleh Maulani dkk (2015) dalam penelitian tersebut membandingkan antara *nested join* dan *hash join*. Penelitian yang diajukan ini topik objeknya berbeda perpustakaan dan penjualan online.

Penelitian yang ketiga dilakukan oleh Habimana (2015) dalam penelitian tersebut menjelaskan tentang bagaimana penulisan *query* yang baik dan cepat dalam hal waktu. Penelitian yang diajukan sama dalam penerapan optimalisasi *query* dan perbedaan terletak pada objek yang berbeda.

Penelitian yang keempat dilakukan oleh Nilaliliana Prihatin (2017) pada skripsi yang berjudul Sistem Informasi Pencatatan aktifitas perubahan data nilai mahasiswa, penelitian ini menggunakan indexing dan untuk menghasilkan *query* yang lebih optimal.

Penelitian yang kelima dilakukan oleh Eka Rahmadyani(2016), penelitian ini melakukan perbandingan Metode M2s Crossover Dan Metode Chunk Crossover pada aplikasi pembelajaran.

Penelitian – penelitian di atas digunakan sebagai rujukan dalam pembuatan skripsi tentang “Pengoptimalisasian Query Pada Study Kasus Sistem Informasi Penjualan Kue Di Toko Wien”. Adapun perbedaan penelitian – penelitian sebelumnya dengan penelitian yang dilakukan sekarang dapat dilihat pada Tabel 2.1. :

**Tabel 2.1. Perbandingan Dengan Penelitian Sebelumnya**

| <b>Penulis</b>     | <b>Judul</b>   | <b>Analisis yang digunakan</b>  | <b>Objek Penelitian</b> | <b>RDMS yang dipakai</b> |
|--------------------|--|---|-------------------------|--------------------------|
| Wasino dkk (2013)  | Perbandingan Optimasi Query Dengan Menggunakan Algoritma Join Berdasarkan Waktu Respon                   | Membandingkan antara query dengan algoritma hash join, merge join dan nested loop join        | -                       | SQL Server               |
| Maulani dkk (2015) | Analisis Perbandingan Optimasi Query Nested Join Dan Hash Join Pada Aplikasi Pencarian Data Berbasis Web | Membandingkan antara query Algoritma Hash Join, nested Join Scalar dan Nested Join Correlated | Perpustakaan            | MySQL                    |
| Habimana (2015)    | <i>Query Optimization Techniques - Tips For Writing Efficient And Faster SQL Queries</i>                 | Membandingkan antara query original dengan query yang telah dioptimalkan                      | -                       | ORACLE                   |

Tabel 2.1. Lanjutan

| Penulis           | Judul   | Analisis yang digunakan   | Objek Penelitian   | RDMS yang dipakai |
|-------------------|---|---|--|-------------------|
| Prihatin(2017)    | Optimasi Query Pada Sistem Informasi Pencatatan Aktifitas Perubahan Data Nilai Mahasiswa                        | Membuat aplikasi dengan tabel <i>indexing</i>                                     | Sistem Informasi Pencatatan aktifitas perubahan data nilai mahasiswa | MySQL             |
| Rahmadyani (2016) | Analisa Optimasi Query Database Dalam Pembelajaran Berbasis Web Dengan Metode M2s Crossover Dan Chunk Crossover | Optimasi Dengan Metode M2s Crossover Dan Metode Chunk Crossover                   | -  | -                 |
| Penulis (2018)    | Pengoptimalisasian Query Pada Study Kasus Sistem Informasi Penjualan Kue Di Toko Wien                           | Membandingkan antara query penggabungan tabel dengan where clause dan join clause | Sistem Informasi penjualan Kue Online di Toko Wien                   | MySQL             |

## 2.2. Dasar Teori

### 2.2.1 Database

Sistem pangkalan data atau basis data (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*Database Management System, DBMS*).

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya, penjelasan ini disebut skema. Skema

menggambarkan objek yang diwakili suatu basis data, dan hubungan di antara objek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data. Hal ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah Layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan di mana setiap tabel terdiri dari baris dan kolom. Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel.

Istilah basis data mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai sistem manajemen basis data (*Database Management System/DBMS*). Jadi secara konsep basis data atau *database* adalah kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relational*) dengan tatacara yang tertentu untuk membentuk data baru atau informasi. Atau basis data (*database*) merupakan kumpulan dari data yang saling berhubungan (relasi) antara satu dengan yang lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu.

Pada komputer, basis data disimpan dalam perangkat hardware penyimpan, dan dengan software tertentu dimanipulasi untuk kepentingan atau kegunaan tertentu. Hubungan atau relasi data biasanya ditunjukkan dengan kunci (*key*) dari tiap file yang ada. Data merupakan fakta atau nilai (*value*) yang tercatat atau merepresentasikan deskripsi dari suatu objek. Data yang merupakan fakta yang tercatat dan selanjutnya dilakukan pengolahan (proses) menjadi bentuk yang berguna atau bermanfaat bagi pemakainya akan membentuk apa yang disebut informasi. Bentuk informasi yang kompleks dan terintegrasi dan pengolahan

sebuah database dengan komputer akan digunakan untuk proses pengambilan keputusan pada manajemen akan membenuk Sistem Informasi Manajemen (SIM), data dalam basis data merupan item terkecil dan terpenting untuk membangun basis data yang baik dan valid. Data dalam basis data bersifat *integrated* dan *shared*.

Secara umum, database atau basis data berarti koleksi data yang saling terkait. Secara praktis, basis data dapat dianggap sebagai suatu penyusun data yang terstruktur yang disimpan dalam media pengingat (*harddisk*) yang tujuannya adalah agar data tersebut dapat diakses dengan mudah dan cepat (Kadir, 2008).

### **2.2.2 MySQL**

MySQL didefinisikan sebagai sistem manajemen database. Database merupakan struktur penyimpanan data untuk menambah, mengakses dan memproses data yang disimpan dalam sebuah database. Selain itu MySQL dapat dikatakan sebagai basis data terhubung (*Relational Database Management System /RDBMS*). Server database MySQL mempunyai kecepatan akses tinggi, mudah digunakan dan handal. MySQL dikembangkan untuk menangani database yang besar secara cepat dan telah sukses digunakan. Fitur utama MySQL (Kustiahningsih dan Anamisa, 2011).

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Structure Query Langguage*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, di mana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, di mana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

### **2.2.3 Structure Query Language (SQL)**

Sejarah SQL dimulai dari artikel seorang peneliti dari IBM bernama Jhonny Oracle yang membahas tentang ide pembuatan basis data relasional pada bulan Juni 1970. Artikel ini juga membahas kemungkinan pembuatan bahasa standar untuk mengakses data dalam basis data tersebut. Bahasa tersebut kemudian diberi nama SEQUEL (*Structured English Query Language*).

SQL (*Structured Query Language*) adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara de facto merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya. Standarisasi SQL dimulai pada tahun 1986, ditandai dengan dikeluarkannya standar SQL oleh ANSI. Standar ini sering disebut dengan SQL86. Standar tersebut kemudian diperbaiki pada tahun 1989 kemudian diperbaiki lagi pada tahun 1992. Versi terakhir dikenal dengan SQL92. Pada tahun 1999 dikeluarkan standar baru yaitu SQL99 atau disebut juga SQL99, akan tetapi kebanyakan implementasi mereferensi pada SQL92. Saat ini sebenarnya tidak ada

server basis data yang 100% mendukung SQL92. Hal ini disebabkan masing-masing server memiliki dialek masing-masing.

Melalui SQL, anda dapat melakukan fungsi administrator pada database seperti menjalankan query untuk mengambil data dalam database, mengakses data (read database), menginput data dalam database, menghapus data dari database, serta mengubah data yang berada dalam database.

#### **2.2.4 Optimalisasi Query**

Optimasi merupakan suatu langkah untuk mengoptimalkan waktu menjadi lebih efisien. Ketika sebuah *query* diberikan pada sistem database, optimasi penting dilakukan untuk memilih strategi yang efisien untuk mengevaluasi ekspresi relasi yang ditentukan. *Query optimization* adalah suatu proses untuk menganalisis *query*, menentukan sumber-sumber apa saja yang digunakan oleh *query* tersebut dan apakah penggunaan dari sumber tersebut dapat dikurangi tanpa merubah output. *Query optimization* dapat juga dikatakan sebuah prosedur untuk meningkatkan strategi evaluasi dari suatu *query* untuk membuat evaluasi tersebut menjadi lebih efektif, mencakup beberapa teknik seperti transformasi *query* ke dalam bentuk logika yang sama, memilih jalan akses yang optimal dan mengoptimalkan penyimpanan data. Tujuan dari *query optimization* adalah menemukan jalan akses yang termurah untuk meminimumkan total waktu pada saat proses sebuah *query* (Maulani dkk, 2015).