1. **Latar Belakang**

Pengelolaan gambar sangat erat kaitannya dengan *database*. Desain *database* yang baik sangat diperlukan untuk konten gambar, hal ini sangat penting untuk mendapatkan *database* yang efisien. Sehingga proses *query* yang dilakukan pada database bekerja secara optimal dan waktu yang dibutuhkan pada proses distribusi gambar dalam *database* dapat di optimasi. Proses distribusi gambar di database harus ditangani dengan secara efektif dan efisien agar tidak terjadi kegagalan saaat pengaksesan gambar, maupun kegagalan hubungan komunikasi yang dapat mengakibatkan beberapa atau mungkin banyak gambar yang tidak dapat diakses pada database. Masalah yang sering muncul pada distribusi database apabila suatu database mempunyai sejumlah data yang tersimpan dengan banyak *query* dan table, Suatu akses permintaan kedatabase mengakibatkan proses pencarian data atau *source* data akan lambat. Pada *source* data konvesional perlu dilakukan *query* bertingkat, mulai dari satu table kemudian table lain lalu dari satu *query* ke *query* yang lain. Bagaimana jika ada ratusan atu mungkin ribuan table dan *query* didalam suatu *database* serta terdistribusi sehingga menjalin hubungan antar *database.*

Banyak pendekatan yang dapat dilakukan untuk menyelesaikan masalah tersebut salah satunya dengan metode “*Vertikal Partitioning*”, *Vertikal Partitioning* digunakan untuk meningkatkan kinerja transaksi *database*. *Vertikal Partitioning* yang digunakan untuk menyimpan atribut yang paling sering diakses dalam memori utama dan lebih sedikit halaman dalam memori sekunder dapat diakses. Selama database terdistribusi, fragmen dialokasikan dan direplikasi di lokasi berbeda. *Vertikal Partitioning* akan membagi atribut ke dalam kelompok-kelompok, menciptakan fragmen. Dengan menerapkan *Vertikal Partitioning* pada database maka akan terdapat tiga perspektif penyimpanan: Database yang disimpan dengan memori tunggal, pada memori yang berbeda dan basis data terdistribusi.

Maka dengan munculnya masalah tersebut, disini penulis mencoba mengimplementasikan Algoritma Vertikal Partitioning dalam distribusi database dengan mengambil objek gambar sebagai data yang akan didistribusikan.

1. **Rumusan Masalah**

Rumusan masalah pada Tugas Akhir ini yaitu:

1. Eksplorasi algoritma *Vertical Partitioning* dalam proses fragmentasi dan pengalokasian data gambar.
2. Optimasi waktu yang diperlukan pada saat pendistribusian gambar (dengan ragam ukuran data) dalam relasional basis data.
3. Memaksimalkan kinerja database terhadap aksesibilitas lokal.
4. **Tujuan dan Manfaat**

Tujuan dari penyusunan Tugas Akhir ini untuk mengurangi jumlah atribut tidak relevan yang diakses dengan *query*, sehingga pengambilan gambar dari database *image gallery* lebih efisien.

Manfaat yang diperoleh dari penyusunan Tugas Akhir ini yaitu meningkatkan kinerja eksekusi *query*, *system* *throughput* dan performa dalam proses penyimpanan dan pengambilan gambar serta menyederhanakan proses pemeliharaan database *image gallery*.

1. **Batasan Masalah**

Pada Tugas Akhir ini data multimedia yang distribusikan berupa gambar dengan [*Client/Server Database Architecture*](http://docs.oracle.com/cd/B10501_01/server.920/a96521/ds_concepts.htm#12312) yang dijalankan pada *VMware Machine*.

1. **Metodologi Penelitian**

Masalah yang dibahas dalam Tugas Akhir ini diselesaikan dengan tahapan-tahapan sebagai berikut:

1. Pengumpulan Data

Pada tahap awal ini akan dilakukan pengumpulan data yang berkaitan dengan distribusi database image gallery.

1. Analisis Data

Setelah data dikumpulkan kemudian dianalisis untuk mentukan jenis gambar yang digunakan serta berapa banyak sample yang akan digunakan.

1. Pengolahan Data

Setelah jenis gambar ditentukan selanjutnya memperhitungkan proses distribusi yang akan diterapkan pada sistem.

1. Pemrosesan Data

Tahap ini akan dilakukan pemrosesan data dengan SQL server.

1. Pengembangan Aplikasi

Tahap ini merupakan tahap pengembangan sistem distribusi database vertikal.

1. Pengujian Hasil

Pada tahap ini akan dilakukan pengujian terhadap aplikasi yang telah dibangun.

1. Analisis Hasil

Hasil dari pengujian akan dianalisa apakah sudah sesuai dengan yang diharapkan.

1. Penarikan Kesimpulan

Setelah seluruh tahapan dijalankan, dapat ditarik kesimpulan sesuai dengan yang dibahas pada bagian tujuan Tugas Akhir.

1. **Tinjauan Pustaka**

Sejarah dalam penelitian sistem *database* adalah salah satu produktivitas yang luar biasa dan memberikan dampak ekonomi yang mengejutkan. Lebih kurang 20 tahun, sebagai dasar penelitian lapangan ilmu pengetahuan, penelitian *database* telah memicu layanan informasi industri sebesar $10 milyar per tahun khususnya di Amerika Serikat (Silberschatz et al, 1996).

Hasil dari penelitian *database* mendukung kemajuan sistem komunikasi, transportasi dan logistik, manajemen keuangan, sistem berbasis pengetahuan, akses ke literatur ilmiah dan sejumlah aplikasi sipil dan pertahanan lainnya. Selain itu sistem database juga digunakan sebagai landasan dalam kemajuan bidang ilmu seperti komputasi biologi (Silberschatz et al, 1996).

Saat ini *database* merupakan bagian integral dari kehidupan sehari-hari. Istilah *database* (dalam versi bahasa Indonesia adalah basis data) didefenisikan sebagai sekumpulan data (meliputi file/arsip/tabel) yang saling berhubungan dan disimpan dalam media penyimpanan data sehingga dapat dimanfaatkan dan digunakan dengan cepat dan mudah. *Database* tidak lagi dimiliki oleh satu departemen saja tetapi merupakan sumber daya perusahaan. Database tidak hanya digunakan untuk menyimpan data operasional suatu organisasi tetapi juga deskripsi data organisasi. Untuk alasan ini *database* juga didefenisikan sebagai kumpulan *self-describing* terintegrasi (Thomas, Carolyn, 2005)

* 1. **Pengenalan Data dan Data Gambar**

Menurut Bergeron (2003), data adalah kumpulan kejadian yang diangkat dari suatu kenyataan yang dapat berupa bilangan, terkait dengan angka-angka atau atribut yang bersifat kuantitas, yang berasal dari hasil observasi, eksperimen atau kalkulasi. Kumpulan data yang diperoleh dan saling berhubungan kemudian disaring dan dimaknai sehingga membentuk satu informasi mencakup keseluruhan data. Dalam Alter (2002), data dibagi ke dalam beberapa tipe yaitu:

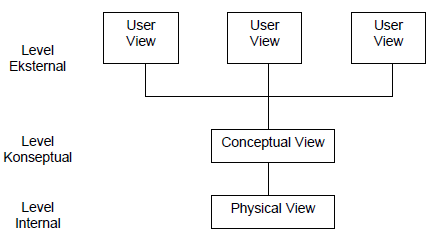
1. Data yang telah terdefenisikan, termasuk di dalamnya data berbentuk angka atau huruf atau gabungannya yang memiliki arti dan *format* khusus yang telah ditentukan untuk mengendalikan proses-proses perhitungan atau transaksi. Contoh: nomor kartu kredit, tanggal transaksi dan lainnya.
2. *Images,* data yang berbentuk gambar atau desain grafis
3. *Audio,* data yang bersifat suara, yang wujudnya adalah frekuensi.
4. *Video,* gabungan antara data berbentuk gambar dan suara.

Seorang penulis, George Abell (1969) dalam karyanya yang berjudul *“Exploration of the Universe”*, data gambar didefenisikan sebagai representasi optik dari objek yang dihasilkan oleh sinar cahaya dari objek yang dibiaskan atau dipantulkan oleh lensa atau cermin.Sifat umum dari suatu gambar yaitu menunjukkan kecerahan objeknya dalam domain spasial. Atribut utama yang harus dimiliki oleh entitas gambar yaitu ukuran, resolusi, tingkat kecerahan dan jenis file gambar itu sendiri.

* 1. **Pengenalan Sistem Database**

Sistem *database* diartikan sebagai sekumpulan database dengan para pengguna yang menggunakannya, baik dari segi perancangan, pengelolahan dan pemeliharaannya secara bersama-sama. Sistem *database* dirancang untuk mengelolah informasi. Pengelolaan informasi tersebut meliputi penyimpanan, penyediaan, dan mekanisme manipulasi informasi. Selain itu sistem *database* menjamin keamanan informasi yang tersimpan didalamnnya dari berbagai ancaman.

Sistem *database* disusun oleh beberapa komponen antara lain perangkat keras, sistem operasi, database, DBMS (*Database Management System*) dan pengguna (*user, programmer*, dan lainnya). Proses pengelolaan data yang dilakukan dalam sistem database tidak dapat dilihat secara fisik. Oleh karena itu proses pengelolaan hanya dapat diabstraksikan sebagai berikut ini :

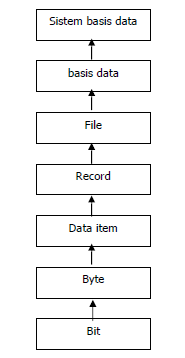


Gambar 4.1. Abstraksi Sistem *database*

Berdasarkan gambar 4.1:

1. *Conceptual view*merupakan pandangan yang berkaitan dengan permasalahan data-data apa saja yang diperlukan untuk disimpan dalam *database* dan penjelasan mengenai hubungan antar data yang satu dengan lainnya. *Conceptual view* dapat disetarakan dengan skema, dilakukan *database* *administrator*.
2. *Physical view*merupakan bentuk implementasi dari *conceptual view*, yaitu pandangan tentang bagaimana data disimpan dalam media penyimpan data.
3. *User view*dapat disejajarkan dengan sub skema.

Sistem *database* juga mencakup semua bentuk komponen data yang ada dalam suatu sistem sedangkan *database* merupakan komponen utama yang menyusun sistem *database* tersebut (Thomas, Carolyn, 2005).



Gambar 4.2. Komponen utama penyusun sistem *database*.

* 1. **Tujuan Sistem Database**

Kemunculan sistem *database* mengawali pengelolaan komputerisasi data komersial, sebagai contoh pengelolaan data organisasi universitas yang melibatkan penyimpanan informasi mengenai siswa, infrastruktur universitas dan data kepegawaian. Dalam contoh tersebut dibutuhkan cara untuk menyimpan informasi pada komputer dengan menyimpannya dalam sistem operasi. Untuk memungkinkan pengguna dalam memanipulasi informasi diperlukan sistem *database* untuk mengelola sejumlah file informasi terstruktur (Silberschatz et al, 2009).

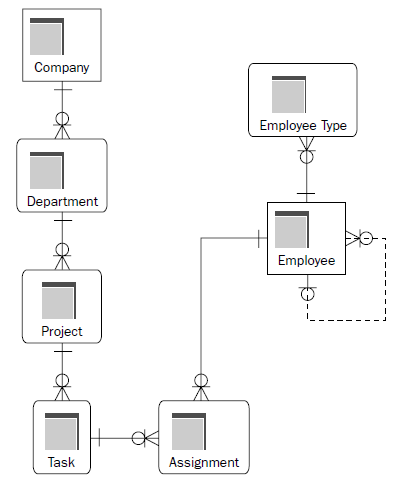
* 1. **DBMS (Database Management System)**

DBMS adalah suatu sistem *software* yang memungkinkan pengguna untuk mendefenisikan, menciptakan, memeliharan dan mengontrol akses pada database. DBMS juga berinteraksi dengan program aplikasi pengguna dan *database*. Secara umum DBMS menyediakan fasilitas sebagai berikut:

1. DDL (*Data Defenition Language*) yang memungkinkan pengguna untuk mendefenisikan *database*, menciptakan dan menentukan struktur tabel, serta *constraints* pada data yang akan disimpan dalam database.
2. DML (*Data Manipulation Language*) merupakan instruksi-instruksi yang yang digunakan untuk menampilkan, menambahkan, mengubah dan menghapus data dari *database*.
3. SQL (*Structured Query Language*) yang merupakan bahasa query yang digunakan untuk DBMS relasional.
4. *Security System* yang bertujuan menyediakan fasilitas keamanan untuk mencegah pengguna yang tidak sah mengakses database.
5. *Integrity System* bertujuan memberikan pertahanan yang konsisten terhadap data yang tersimpan.
6. CCS (*Concurrency Control System*) yang mengijinkan berbagai akses database (Thomas, Carolyn, 2005).
   1. **RDM (Relational Database Models)**

RDM bertujuan memperbaiki struktur hirarki dari suatu data, seperti pada gambar 4.3. Dengan adaya RDM, setiap tabel dapat diakses langsung tanpa harus mengakses seluruh objek induk. Selain manfaat lain yang didapat dari RDM adalah setiap tabel dapat dihubungkan secara bersamaan terlepas dari posisi hirarkinya. Proses relasi tidak dibatasi oleh struktur hirarki, sehingga tabel induk (*parent tables*) dapat dihubungkan dengan satu atau lebih tabel anak (*dependent table*) (Powell, 2006, Silberschatz et al, 2009).

.



Gambar 4.3. Hirarki *Relational Database Model*

* 1. **Desain Database dan Model ERD (Entity Relationship Database)**

Dalam merancang lingkungan aplikasi *database* yang lengkap dan struktural, disarankan mengikuti aspek-aspek penting seperti *physical, logical* dan *view levels*. Oleh karena itu, diperlukan skema desain *database* struktural yang menggabungan antar entitas utama yang terdapat dari data-data yang dikumpulkan. ERD merupakan cara yang tepat untuk menghubungan antar entitas data model dan mengidentifikasi entitas yang terkait untuk direpresentasikan dalam *database* (Sudarshan, 2004). Selain itu ERD juga berguna dalam hal pemetaan struktur data atau informasi dari suatu organisasi ke dalam bentuk skema konseptual. Dalam merancang ERD ada tiga konsep yang harus dipertimbangkan, yaitu entitas, hubungan dan atribut yang digunakan. (Silberschatz et al, 2009).

* 1. **Relational Model**

RDBMS (*Relational Database Model System*) telah menjadi *software* yang dominan dalam pengolahan data saat ini. RDBMS adalah representasi generasi kedua dari DBMS didasarkan pada *relational data models* yang diusulkan oleh E.F.Codd (1970). Dalam *relational model*, seluruh data terstruktur secar logis dalam tabel. Setiap tabel memiliki nama atribut yang disebut kolom (*columns)*. Setiap *columns* memiliki atribut baris (*tuple*) yang berisi data.

* 1. **Query dan Structure Query Language**

*Query* adalah perintah-perintah untuk mengakses data pada sistem basis data, *Query* merupakan bahasa untuk melakukan manipulasi terhadap database yang telah distandartkan dan lebih dikenal dengan nama *Structure Query Language*. Query dibedakan menjadi 2, yaitu :

1. Untuk membuat/mendefinisikan obyek-obyek *database* seperti membuat tabel, relasi dan sebagainya. Biasanya disebut dengan *Data Definition Language* (DDL)
2. Untuk memanipulasi data, yang biasanya dikenal dengan *Data Manipulation Language* (DML). Manipulasi data bisa berupa menambah, mengubah atau menghapus data. Pengambilan informasi yang diperlukan dari *database*, yang mana datanya diambil dari tabel maupun dari *query* sebelumnya.
3. *Structure Query Language*

SQL bahasa yang digunakan untuk menyimpan, memanipulasi dan melakukan query data dalam database relasional. Generasi pertama SQL muncul pada tahun 1974 yang dikembangkan oleh IBM. Database realasional komersial yang pertama dirilis oleh Relational Software (kemudian menjadi Oracle). Jenis perintah SQL :

* 1. DDL (*Data Definition Language*)
  2. DML (*Data Manipulation Language*)
  3. DCL (*Data Control Language*)

Struktur Dasar:

1. ***Select*, berkaitan dengan operasi proyeksi**

Pada aljabar relasional. Digunakan untuk mendaftar atribut yang ingin dikeluarkan sebagai hasil *query*

Operasi select digunakan untuk mengambil sebagian atau seluruh isi tabel dari suatu basisdata.

Contoh : *“Tentukan nama-nama dari semua cabang* *bank dalam relasi loan “*

Query-nya :

**SELECT branch-name FROM loan**

1. ***From*, berkaitan dengan operasi produk**

Kartesian (relasi mana yang akan di-scan). Klausa ***from* menunjukkan daftar relasi** yang dilibatkan dalam *query*. Berkaitan dengan operasi produk kartesian pada aljabar relasional.

Contoh: *Produk Kartesian dari borrower x loan*

**select \* from *borrower, loan***

1. ***Where*, berkaitan dengan predikat seleksi.**

Klausa ***where* menspesifikasi kondisi yang harus** dipenuhi oleh hasil *query.* Berkaitan dengan predikat seleksi pada aljabar relasional.

Contoh : *“Temukan semua loan number untuk pinjaman-pinjaman yang dibuat pada cabang Perryridge dengan jumlah lebih besar dari $1200”.*

Query-nya ditulis sebagai berikut :

**SELECT loan-number FROM loan**

**WHERE branch-name = “Perryridge” and amount**>1200;

* 1. **Data manipulation language (DML)**

*Data manipulation language* (DML) merupakan perintah-perintah yang digunakan untuk menampilkan, menambahkan, mengubah, dan menghapus data di dalam objek-objek yang didefinisikan pada DDL.

1. **INSERT TABLE**

Perintah insert table digunakan untuk menambahkan data-data pada table yang kita buat, perintah-perintah yang digunakan adalah sebagai berikut: ***Insert into “nama\_tabel” values (nilai1,nilai2,….,nilai n)***

1. **UPDATE TABLE**

Perintah update table digunakan untuk mengubah data yang telah kita masukkan ke dalam table yang ada dengan data yang baru. Adapun baris perintah umum yang digunakan dalam Update table adalah sebagai berikut: ***update nama\_tabel set nama\_kolom = “data yang baru” where kondisi yang diminta***

1. **DROP TABLE**

**DROP TABLE** perintah drop table digunakan untuk mengahpus table yang telah kita buat tadi, adapun perintah yang digunakan adalah sebagai berikut: ***drop table “nama\_tabel”*** sebagai contoh kita akan menghapus table siswa, maka perintah yang digunakan adalah sebagai berikut: ***drop table siswa***

1. **ADD COLUMN**

Perintah ini digunakan untuk menambah kolom dalam table apabila kita salah membuat table. Akan tetapi kolom ini langsung berada di urutan belakang pada kolom-kolom lain pada table tersebut. Adapun perintah untuk menambahkan kolom dengan *add column* adalah sebagai berikut: ***Alter table “nama\_tabel” add “nama\_kolom” “tipe”data (ukuran)”***

1. **DROP COLUMN**

Perintah *drop column* akan menghapus kolom kolom yang kita inginkan pada table yang ada, adapun perintah umumnya adalah sebagai berikut: ***Alter table “nama\_tabel” drop column “nama\_kolom”.***

([http://www.1keydata.com](http://www.1keydata.com/sql/), 2013)***.***

* 1. **Distribusi Database**

Basis data terdistribusi (DDB) adalah kumpulan dari beberapa, database secara logis saling terkait yang didistribusikan melalui jaringan komputer. Sistem manajemen database terdistribusi (distributed DBMS) kemudian didefinisikan sebagai sistem perangkat lunak yang memungkinkan manajemen basis data terdistribusi dan membuat distribusi transparan kepada pengguna. (M. Tamer Özsu, Patrick Valduriez, 2006).

* 1. **Partitioning**

*Partitioning* merupakan teknik desain *database* yang mampu mengurangi waktu saat melakukan pembacaan data. Data secara fisik disimpan oleh *database administrator* sehingga dapat meningkatkan kemampuan aplikasi untuk mengakses data. *“Telah lama diketahui bahwa database partitioning adalah jawaban untuk meningkatkan kinerja dan skalabilitas database relasional"* dinyatakan oleh CodeFutures, sebuah perusahaan pengembangan perangkat lunak.

Terdapat dua jenis partitioning:

1. Horizontal   
   Pada horizontal partitioning, database dibagi menjadi baris tabel. Ini bentuk fisik berbasis baris dataset bahwa baik dapat diatasi secara individual — di partisi tunggal — atau lain dapat ditangani dari satu partisi untuk semua partisi.

"Semua kolom didefinisikan ke tabel yang ditemukan dalam setiap set partisi sehingga atribut tabel sebenarnya tidak ada hilang" oleh Robin Schumacher, Direktur manajemen produk untuk MySQL.

1. Vertikal

Partisi vertikal adalah data dalam tabel dibagi secara vertikal. Khusus kolom di database disertakan dalam dataset. Setiap partisi mencakup semua baris. Pendekatan ini juga dapat digunakan untuk disebut pengelompokan data, yang merupakan istilah yang digunakan untuk menggambarkan pengumpulan data biner yang disimpan sebagai entitas tunggal dalam database.

"Contoh partisi vertikal mungkin berupa tabel yang berisi teks atau kolom dengan jumlah luas yang kemudian dipecah menjadi dua tabel yang memiliki kolom paling direferensikan dalam satu tabel dan jarang dirujuk oleh teks atau data lain," dijelaskan Schumacher.

([http://www.datavail.com](http://www.datavail.com/), 2013).

1. **Daftar Pustaka**

Lisabet, Rodriguez, Li, Xiaoou, 2011, A Vertical Partitioning Algorithm for Distributed Multimedia Databases, Department of Computer Science, CINVESTAV-IPN

Silberschatz, A., Korth, H.F. dan Sudarshan, S., 2009, Database System Concepts, The Mc-Graw Hill Companies.

Thomas, M.C. dan Carolyn E.B, 2005, Database Systems A Practical Approach to Design Implementation and Management, edisi 4, Addison Wesley.

Powell, Gavin, 2006, Beginning Database Desain, Wiley Publishing, Inc., Indianapolis, Indiana.

Agrawal, S., Narasayya, V. dan Yang, B., 2004, Integrating Vertical and Horizontal Partitioning into Automated Physical Database Design, Paris, France.

M. Tamer Özsu, Patrick Valduriez, 2006, Principles of Distributed Database Systems, edisi 3, Springe, New York.

DATAVAIL, [http://www.datavail.com/category-blog/database-partitioning options/](http://www.datavail.com/category-blog/database-partitioning%20options/), tanggal akses: 24 November 2013.

1keydata, <http://www.1keydata.com/sql/>, tanggal akses: 24 November 2013.