

SKRIPSI

**STUDI DAN IMPLEMENTASI DBMS BERBASIS MEMORI,
VOLTDDB**



Aditya Putra Leksono

NPM: 2015730066

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2019**

UNDERGRADUATE THESIS

**STUDY AND IMPLEMENTATION OF MEMORY-BASED
DBMS, VOLTDB**



Aditya Putra Leksono

NPM: 2015730066

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2019**

LEMBAR PENGESAHAN

STUDI DAN IMPLEMENTASI DBMS BERBASIS MEMORI, VOLTDB

Aditya Putra Leksono

NPM: 2015730066

Bandung, 28 Mei 2019

Menyetujui,

Pembimbing

Dr. Veronica Sri Moertini

Ketua Tim Penguji

Anggota Tim Penguji

Kristopher David Harjono, M.T.

Rosa De Lima, M.Kom.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

STUDI DAN IMPLEMENTASI DBMS BERBASIS MEMORI, VOLTDB

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 28 Mei 2019

Meterai Rp. 6000

Aditya Putra Leksono
NPM: 2015730066

ABSTRAK

DBMS berbasis memori menyimpan seluruh data di dalam memori. Hal tersebut dapat meningkatkan kecepatan pemrosesan data secara signifikan. Waktu pemrosesan data pada DBMS berbasis memori akan unggul lebih cepat dari DBMS konvensional yang memanfaatkan *disk* sebagai media penyimpanannya. Dalam DBMS konvensional, data perlu dipindahkan dari *disk* ke memori utama terlebih dahulu untuk diproses.

Salah satu perangkat lunak DBMS berbasis memori ialah VoltDB. VoltDB merupakan salah satu perangkat lunak DBMS berbasis memori yang berfokus pada *fast data*. *Fast data* merupakan *real-time* data yang perlu diproses secara cepat. Desain *database* dari VoltDB mendukung kebutuhan akan *fast data* tersebut. Dengan desain *database* partisi, seluruh transaksi dan *request record* dapat berjalan secara paralel.

Penelitian ini dilakukan untuk membandingkan kecepatan pemrosesan kueri antara DBMS berbasis memori dengan DBMS konvensional. Pemrosesan kueri yang diuji adalah untuk operasi INSERT, DELETE, dan SELECT. Pada penelitian ini VoltDB dibandingkan dengan salah satu DBMS konvensional yaitu MySQL. Jumlah data bahan uji kecepatan eksekusi kueri pada VoltDB dan MySQL adalah 1.000, 10.000, dan 100.000 *records*. Hasil akhir yang dibandingkan adalah rata-rata waktu eksekusi kueri dari tujuh kali pengujian. Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa VoltDB unggul dari MySQL di seluruh poin pengujian dan VoltDB akan berjalan maksimal ketika tabelnya dipartisi dan melakukan pemanggilan *stored procedure* secara *asynchronous*.

Kata-kata kunci: DBMS konvensional, DBMS berbasis *disk*, *disk*, DBMS berbasis memori, memori, VoltDB

ABSTRACT

Memory-based DBMS stores all data in memory. This can increase data processing speed significantly. Data processing time in memory-based DBMS will be far superior than conventional DBMS that uses disk as its primary storage media. In conventional DBMS, data needs to be moved from disk to main memory first to be processed.

One of the memory-based DBMS software is VoltDB. VoltDB is a memory-based DBMS software that focuses on fast data. Fast data is real-time data that typically need to be analyzed quickly to make rapid decisions. VoltDB's database design supports the need for fast data. With the partition database design, all transaction and request records can run in parallel.

This research was conducted to compare the processing speed of queries between memory-based DBMS and conventional DBMS. Processing queries tested are for INSERT, DELETE, and SELECT operations. In this research VoltDB is compared to MySQL, one of the conventional DBMS. The amount of data for query execution speed tests on VoltDB and MySQL are 1,000, 10,000 and 100,000 records. The value that compared between VoltDB and MySQL is the average query execution time of seven tests. Based on the test results that have been done, it can be concluded that VoltDB is superior to MySQL in all test points and VoltDB will run optimally when the table is partitioned and call stored procedures asynchronously.

Keywords: Conventional DBMS, disk-based DBMS, disk, memory-based DBMS, memory, VoltDB

Dipersembahkan kepada keluarga dan orang-orang terdekat penulis

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat-Nya penulis dapat menyelesaikan skripsi yang berjudul "Studi dan Implementasi DBMS Berbasis Memori, VoltDB". Dalam proses penyusunan skripsi, penulis banyak mendapat kesempatan untuk mempelajari hal-hal baru serta mendapatkan banyak bantuan dan dukungan baik secara langsung maupun tidak langsung dari banyak pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih pada seluruh pihak yang telah berperan dalam penyusunan skripsi ini. Secara khusus penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua, serta adik penulis yang senantiasa mendoakan dan memberikan dukungan kepada penulis selama penyusunan skripsi ini.
2. Ibu Dr. Veronica Sri Moertini selaku dosen pembimbing yang telah memberi arahan dan masukan selama penyusunan skripsi.
3. Ibu Vania Natali, M.T yang telah memberi saran dan masukan dalam penyusunan skripsi ini.
4. Bapak Kristopher David Harjono M.T. dan Ibu Rosa De Lima M. Kom. selaku dosen penguji yang telah memberikan kritik dan saran untuk memperbaiki penyusunan skripsi ini.
5. Andi Tangguh, Aria Lesmana, Ario Laksono, Enrico, Hapsari Laksmi, Nadya Vio, Reyner Alexander, dan Yonathan Kristian yang telah menyemangati dan membantu melepas penat penyusunan skripsi.
6. M. Rifqi, Tedi Tri, dan Vincent Joel yang telah mendukung dan menyemangati penulis untuk menuntaskan penulisan skripsi ini.
7. Enrico, Hizkia Steven, dan Reyner Alexander yang telah membantu penulis agar tetap hidup sehat dengan berolahraga setiap minggu.
8. Irvan Wijaya, Matthew Ariel, dan M. Rifqi yang saling membantu dan memberi informasi sebagai mahasiswa bimbingan skripsi Ibu Veronica.
9. Rekan-rekan teknik informatika angkatan 2015 yang tidak dapat disebutkan satu-persatu.
10. Pihak-pihak lain yang belum telah membantu penulisan skripsi ini, yang terus memberikan do'a dan semangat kepada penulis.

Akhir kata, penulis berharap agar skripsi ini dapat bermanfaat dan memberikan kontribusi baik untuk pembelajaran maupun penelitian selanjutnya.

Bandung, Mei 2019

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	5
2.1 Konsep DBMS	5
2.1.1 Abstraksi Data pada DBMS	6
2.1.2 Independensi Data	6
2.1.3 Model Data	7
2.2 Arsitektur DBMS	7
2.2.1 <i>Interface Driver</i>	8
2.2.2 <i>SQL Engine</i>	9
2.2.3 <i>Transaction Engine</i>	10
2.2.4 <i>Relational Engine</i>	11
2.2.5 <i>Storage Engine</i>	12
2.2.6 <i>Memori</i>	12
2.2.7 <i>Database Files</i>	12
2.3 <i>Data Storage</i> DBMS	14
2.4 Manajemen Transaksi DBMS	15
2.4.1 Kontrol Konkurensi	15
2.4.2 <i>Recovery</i>	16
2.5 DBMS Berbasis Memori	17
2.5.1 Arsitektur DBMS Berbasis Memori	17
2.6 <i>Data Storage</i> DBMS Berbasis Memori	19
2.7 Manajemen Transaksi DBMS Berbasis Memori	19
2.7.1 Kontrol Konkurensi	19
2.7.2 <i>Recovery</i>	19
2.8 VoltDB	19
2.8.1 Arsitektur VoltDB	20
2.8.2 Desain <i>Database</i>	21
2.8.3 Transaksi	22

2.8.4	<i>Stored Procedure</i>	23
2.8.5	<i>Recovery</i>	26
2.8.6	Penggunaan Memori	26
3	STUDI EKSPLORASI DAN EKSPERIMEN VOLTDB	29
3.1	Pendahuluan	29
3.1.1	Instalasi VoltDB	29
3.1.2	Utilitas VoltDB	30
3.1.3	Konfigurasi <i>Cluster</i> dan <i>Database</i>	30
3.1.4	Inisialisasi <i>Database</i>	31
3.1.5	Memulai <i>Database</i>	32
3.1.6	Menghentikan <i>Database</i>	32
3.1.7	<i>Skema Database</i>	33
3.2	Modul Program <i>Create Table</i>	34
3.3	Modul Program Kueri <i>Database</i> dan <i>Retrieve Data</i>	35
4	ANALISIS DAN PERANCANGAN PERANGKAT LUNAK DEMO	39
4.1	Gambaran Umum Perangkat Lunak Demo VoltDB	39
4.2	Analisis Perangkat Lunak Demo VoltDB	39
4.2.1	Diagram <i>Use-Case</i>	39
4.2.2	Skenario	40
4.3	Perancangan Perangkat Lunak Demo VoltDB	43
4.3.1	Perancangan Antarmuka Perangkat Lunak Demo VoltDB	43
5	IMPLEMENTASI DAN PENGUJIAN	47
5.1	Implementasi Perangkat Lunak Demo	47
5.1.1	Deskripsi Perangkat Keras dan Perangkat Lunak yang Digunakan	47
5.1.2	Implementasi Antarmuka	47
5.2	Pengujian Perangkat Lunak Demo VoltDB	52
5.3	Eksperimen untuk Uji Performa VoltDB	52
5.3.1	Metode	53
5.3.2	Tujuan	53
5.3.3	Pra-Pengujian	53
5.3.4	Pengujian Operasi INSERT	54
5.3.5	Pengujian Operasi DELETE	60
5.3.6	Pengujian Operasi SELECT 1	66
5.3.7	Pengujian Operasi SELECT 2	69
5.3.8	Kesimpulan	75
6	KESIMPULAN DAN SARAN	77
6.1	Kesimpulan	77
6.2	Saran Penelitian Lanjutan	78
	DAFTAR REFERENSI	79
	A KODE PROGRAM MODUL EKSPERIMEN	81
	B KODE PROGRAM PERANGKAT LUNAK DEMO	83
	C KODE PROGRAM PENGUJIAN <i>DATABASE</i> PEMILIHAN	101
C.1	Data builder untuk SELECT sederhana	101
C.2	Data builder untuk SELECT kompleks	102
C.3	INSERT	102

C.4 DELETE	105
C.5 SELECT 1	108
C.6 SELECT 2	110
D KODE PROGRAM PENGUJIAN <i>DATABASE</i> PELANGGAN	115
D.1 Generator data	115
D.2 Data builder untuk SELECT sederhana	116
D.3 Data builder untuk SELECT kompleks	116
D.4 INSERT	117
D.5 DELETE	120
D.6 SELECT 1	123
D.7 SELECT 2	125

DAFTAR GAMBAR

2.1	Tingkatan abstraksi data pada DBMS	6
2.2	Contoh <i>database</i> relasional	7
2.3	Arsitektur DBMS	8
2.4	Komponen <i>optimizer</i>	10
2.5	<i>Database files</i>	12
2.6	Visualisasi index B-tree	13
2.7	Struktur <i>disk</i>	14
2.8	Alur pemrosesan data pada DBMS konvensional	15
2.9	Arsitektur DBMS berbasis memori	17
2.10	Pengelolaan data DBMS berbasis memori	18
2.11	Arsitektur H-Store	20
2.12	Arsitektur VoltDB - <i>Cluster</i>	20
2.13	Arsitektur VoltDB - <i>Node</i>	21
2.14	Contoh partisi	22
2.15	Contoh replikasi	22
2.16	Contoh transaksi partisi tunggal	23
2.17	Contoh transaksi multi-partisi	23
2.18	Eksekusi <i>Stored procedure</i> dalam partisi	24
2.19	Ilustrasi pemanggilan <i>stored procedure</i> secara <i>synchronous</i>	25
2.20	Ilustrasi pemanggilan <i>stored procedure</i> secara <i>asynchronous</i>	25
2.21	<i>Recovery</i> pada VoltDB	26
2.22	Kategori memori dalam VoltDB	27
3.1	Pemulaian <i>Database</i> VoltDB	32
3.2	Hasil eksekusi <i>stored procedure</i>	34
3.3	Potongan kode program ModulCreateTable.java	35
3.4	Hasil modul program <i>create table</i>	35
3.5	Potongan kode program ModulKueri.java	36
3.6	Struktur VoltTable	36
3.7	Hasil modul program kueri <i>database</i> dan <i>retrieve data</i>	37
4.1	Diagram <i>use case</i> perangkat lunak demo VoltDB	40
4.2	Rancangan halaman utama - Tab jelajahi	43
4.3	Rancangan halaman utama - Tab partisi	44
4.4	Rancangan halaman utama - Tab SQL	44
4.5	Rancangan halaman utama - Tab operasi lainnya	45
4.6	Rancangan halaman <i>Create Table</i>	45
4.7	Rancangan halaman eksekusi <i>Stored Procedure</i>	46
5.1	Halaman utama - Tab jelajahi	48
5.2	Halaman utama - Tab partisi	49
5.3	Halaman utama - Tab SQL	49
5.4	Halaman utama - Tab operasi lainnya	50

5.5	Halaman <i>create table</i>	51
5.6	Halaman eksekusi <i>stored procedure</i>	51
5.7	Skema <i>database</i> pemilihan	54
5.8	Skema <i>database</i> pelanggan	54
5.9	<i>Execution plan</i> kueri INSERT sebelum mempartisi tabel VOTES pada kolom PHONE_NUMBER	55
5.10	<i>Execution plan</i> kueri INSERT setelah mempartisi tabel VOTES pada kolom PHONE_NUMBER	55
5.11	<i>Execution plan</i> kueri INSERT pada MySQL	56
5.12	Diagram hasil pengujian INSERT tanpa mempartisi tabel VOTES pada kolom PHONE_NUMBER	56
5.13	Diagram hasil pengujian INSERT dengan mempartisi tabel VOTES pada kolom PHONE_NUMBER	57
5.14	<i>Execution plan</i> kueri INSERT sebelum mempartisi tabel CUSTOMER pada kolom ID	58
5.15	<i>Execution plan</i> kueri INSERT setelah mempartisi tabel CUSTOMER pada kolom ID	58
5.16	<i>Execution plan</i> operasi INSERT pada MySQL	58
5.17	Diagram hasil pengujian INSERT tanpa mempartisi tabel CUSTOMER pada kolom ID	59
5.18	Diagram hasil pengujian INSERT dengan mempartisi tabel CUSTOMER pada kolom ID	60
5.19	<i>Execution plan</i> kueri DELETE sebelum mempartisi tabel VOTES pada kolom CONTESTANT_NUMBER	61
5.20	<i>Execution plan</i> kueri DELETE setelah mempartisi tabel VOTES pada kolom CONTESTANT_NUMBER	61
5.21	<i>Execution plan</i> kueri DELETE pada MySQL	62
5.22	Diagram hasil pengujian DELETE tanpa mempartisi tabel VOTES pada kolom CONTESTANT_NUMBER	62
5.23	Diagram hasil pengujian DELETE dengan mempartisi tabel VOTES pada kolom CONTESTANT_NUMBER	63
5.24	<i>Execution plan</i> kueri DELETE sebelum mempartisi tabel CUSTOMER pada kolom ID	64
5.25	<i>Execution plan</i> kueri DELETE setelah mempartisi tabel CUSTOMER pada kolom ID	64
5.26	<i>Execution plan</i> operasi DELETE pada MySQL	64
5.27	Diagram hasil pengujian DELETE tanpa mempartisi tabel CUSTOMER pada kolom ID	65
5.28	Diagram hasil pengujian DELETE dengan mempartisi tabel CUSTOMER pada kolom ID	65
5.29	<i>Execution plan</i> kueri SELECT pada tabel CONTESTANTS	66
5.30	<i>Execution plan</i> kueri SELECT pada MySQL	67
5.31	Diagram hasil pengujian SELECT pada tabel CONTESTANTS	67
5.32	<i>Execution plan</i> kueri SELECT pada tabel CUSTOMER	68
5.33	<i>Execution plan</i> kueri SELECT pada MySQL	68
5.34	Diagram hasil pengujian SELECT pada tabel CUSTOMER	69
5.35	Hasil eksekusi kueri SELECT kompleks pada <i>database</i> pemilihan	70
5.36	<i>Execution plan</i> kueri SELECT kompleks pada <i>database</i> pemilihan (VoltDB)	71
5.37	<i>Execution plan</i> kueri SELECT kompleks pada <i>database</i> pemilihan (MySQL)	71
5.38	Diagram hasil pengujian SELECT kompleks pada <i>database</i> pemilihan (VoltDB)	72
5.39	Hasil eksekusi kueri SELECT kompleks pada <i>database</i> pelanggan	73
5.40	<i>Execution plan</i> kueri SELECT kompleks pada <i>database</i> pelanggan (VoltDB)	73
5.41	<i>Execution plan</i> kueri SELECT kompleks pada <i>database</i> pelanggan (MySQL)	74

5.42 Diagram hasil pengujian SELECT kompleks pada <i>database</i> pelanggan (VoltDB) . .	75
--	----

DAFTAR TABEL

3.1	Tugas pengelolaan <i>database</i>	29
3.2	Tabel fitur pada <i>file</i> konfigurasi	31
3.3	Tabel contoh	36
5.1	Tabel hasil pengujian perangkat lunak demo VoltDB	52
5.2	Tabel hasil pengujian kueri INSERT tanpa mempartisi tabel VOTES pada kolom PHONE_NUMBER	56
5.3	Tabel hasil pengujian kueri INSERT dengan mempartisi tabel VOTES pada kolom PHONE_NUMBER	57
5.4	Tabel hasil pengujian kueri INSERT tanpa mempartisi tabel CUSTOMER pada kolom ID	59
5.5	Tabel hasil pengujian kueri INSERT dengan mempartisi tabel CUSTOMER pada kolom ID	59
5.6	Tabel hasil pengujian kueri DELETE tanpa mempartisi tabel VOTES pada kolom CONTESTANT_NUMBER	62
5.7	Tabel hasil pengujian kueri DELETE dengan mempartisi tabel VOTES pada kolom CONTESTANT_NUMBER	63
5.8	Tabel hasil pengujian kueri DELETE tanpa mempartisi tabel CUSTOMER pada kolom ID	65
5.9	Tabel hasil pengujian kueri DELETE dengan mempartisi tabel CUSTOMER pada kolom ID	65
5.10	Tabel hasil pengujian kueri SELECT pada tabel CONTESTANTS	67
5.11	Tabel hasil pengujian kueri SELECT pada tabel CUSTOMER	69
5.12	Tabel hasil pengujian kueri SELECT kompleks pada <i>database</i> pemilihan	72
5.13	Tabel hasil pengujian kueri SELECT kompleks pada <i>database</i> pelanggan	74

BAB 1

PENDAHULUAN

Bab ini berisi tentang latar belakang yang mendasari pembuatan skripsi ini, rumusan masalah yang muncul dari latar belakang, tujuan yang ingin dicapai, ruang lingkup pembahasan, metodologi penelitian yang digunakan, dan sistematika penulisan dari skripsi ini

1.1 Latar Belakang

Peningkatan jumlah data secara drastis diikuti dengan meningkatnya kebutuhan pemrosesan data secara *real-time* dalam skala besar mengharuskan *database management system* (DBMS) konvensional agar dapat mengelola data yang semakin besar tersebut dengan cepat dan efektif. DBMS konvensional memanfaatkan *disk* untuk pengolahan datanya sehingga operasi I/O pada *disk* menjadi hambatan utama dalam pengembangan performa DBMS konvensional.

Adanya kebutuhan untuk mengakses dan mengolah data berukuran besar dengan sangat cepat merupakan alasan utama pengembangan DBMS berbasis memori sebagai pengganti DBMS konvensional. DBMS berbasis memori menyimpan seluruh datanya di dalam memori [1]. Perbandingan kecepatan pengolahan data pada DBMS berbasis memori diklaim jauh lebih cepat dibandingkan DBMS konvensional yang memanfaatkan *hard-disk drive* (HDD) atau *solid state drive* (SSD) guna penyimpanan dan pengolahan datanya. Beberapa DBMS berbasis memori yg dapat mengelola data dan memproses kueri di memori komputer telah dikembangkan para peneliti (a. l. VoltDB, MemSQL, dashDB, dll.) dan perlu diteliti cara kerja dan kinerjanya. Fokus dalam skripsi ini adalah mengenai VoltDB yang merupakan salah satu sistem perangkat lunak DBMS berbasis memori. VoltDB menyediakan penyimpanan dan kemampuan komputasi terdistribusi atau dapat dijalankan dalam sebuah *cluster* [2]. VoltDB tersedia dalam edisi *community* dan *enterprise*. Edisi *enterprise* menyediakan fitur yang lebih banyak dibandingkan dengan edisi *community*.

DBMS berbasis memori merupakan teknologi yang masih terbilang baru. Oleh karena itu, DBMS berbasis memori khususnya VoltDB perlu dipelajari dan dimanfaatkan untuk mengelola dan mengakses data yang besar dengan waktu yang cepat.

1.2 Rumusan Masalah

Berdasarkan deskripsi di atas maka rumusan masalah yang dapat dibentuk sebagai berikut:

1. Apa saja perbedaan antara DBMS konvensional dengan DBMS berbasis memori?
2. Bagaimana cara kerja VoltDB?
3. Bagaimana membangun perangkat lunak demo untuk VoltDB?
4. Bagaimana perbandingan kinerja VoltDB (DBMS berbasis memori) dengan MySQL (DBMS konvensional)?

1.3 Tujuan

Adapun tujuan yang akan dicapai melalui skripsi ini adalah:

1. Memahami konsep DBMS konvensional dan DBMS berbasis memori.
2. Memahami konsep dan penggunaan VoltDB.
3. Membangun perangkat lunak demo yang mengimplementasikan VoltDB.
4. Menguji kinerja VoltDB dan MySQL pada operasi INSERT, DELETE, dan SELECT.

1.4 Batasan Masalah

Untuk lebih membatasi dan mengarahkan pembahasan terhadap tujuan yang ingin dicapai tersebut, maka ditetapkan batasan-batasan masalah sebagai berikut:

1. Perangkat lunak demo dibangun dengan memanfaatkan bahasa pemrograman *Java*.
2. Penelitian dilakukan terhadap VoltDB edisi *community*.
3. Penelitian terhadap VoltDB dilakukan di *single node*.
4. Pengujian kinerja VoltDB terhadap operasi INSERT, DELETE, dan SELECT.

1.5 Metodologi

Metodologi yang digunakan dalam penyusunan skripsi ini adalah:

1. Studi literatur tentang DBMS konvensional.
2. Studi literatur tentang DBMS berbasis memori.
3. Studi tentang VoltDB dan MySQL.
4. Melakukan instalasi VoltDB dan MySQL.
5. Merancang dan mengimplementasikan perangkat lunak demo.
6. Merancang dan membangun kueri serta *source code* untuk pengujian perbandingan antara VoltDB dengan MySQL.
7. Melakukan pengujian dan eksperimen terhadap VoltDB dan MySQL pada operasi INSERT, DELETE, dan SELECT.
8. Membuat kesimpulan terhadap hasil studi yang diperoleh.
9. Menulis dokumen skripsi.

1.6 Sistematika Pembahasan

Skripsi ini terdiri atas 6 bab, yang terdiri dari Bab 1 Pendahuluan, Bab 2 Dasar Teori, Bab 3 Studi dan Eksplorasi VoltDB, Bab 4 Analisis dan Perancangan Perangkat Lunak Demo, Bab 5 Implementasi dan Pengujian, Bab 6 Kesimpulan dan Saran.

1. Bab 1 Pendahuluan

Bab 1 berisi tentang latar belakang yang mendasari pembuatan skripsi ini, rumusan masalah yang muncul dari latar belakang, tujuan yang ingin dicapai, ruang lingkup pembahasan, metodologi penelitian yang digunakan, dan sistematika penulisan dari skripsi ini.

2. Bab 2 Landasan Teori

Bab 2 berisi teori mengenai konsep DBMS konvensional, DBMS berbasis memori, dan VoltDB. Pada DBMS konvensional dijelaskan mengenai konsep, arsitektur, *data storage*, dan manajemen transaksi dari DBMS konvensional. Pada DBMS berbasis memori dijelaskan mengenai arsitektur, *data storage*, dan manajemen transaksi dari DBMS berbasis memori. Lalu, dijelaskan mengenai VoltDB yang merupakan studi kasus DBMS berbasis memori dalam skripsi ini.

3. Bab 3 Studi Eksplorasi dan Eksperimen VoltDB

Bab 3 berisi tentang pembahasan mengenai studi eksplorasi dan eksperimen VoltDB. Di dalam bab ini dijelaskan tentang cara instalasi dan konfigurasi VoltDB pada *server*. Selain itu dijelaskan juga tentang cara menggunakan VoltDB dan dibuat modul program yang dapat berinteraksi dengan *database* VoltDB.

4. Bab 4 Analisis dan Perancangan Perangkat Lunak Demo

Bab 4 berisi pembahasan mengenai analisis dan perancangan perangkat lunak demo VoltDB. Di dalam bab ini dipaparkan tentang gambaran umum perangkat lunak demo, analisis *use-case*, skenario, dan perancangan antarmuka perangkat lunak demo VoltDB.

5. Bab 5 Implementasi dan Pengujian

Bab 5 berisi pembahasan mengenai implementasi antarmuka dan pengujian perangkat lunak demo beserta pembahasan mengenai uji performa VoltDB.

6. Bab 6 Kesimpulan dan Saran

Bab 6 berisi kesimpulan yang dapat diambil dari penulisan skripsi ini. Selain itu diberikan juga saran-saran untuk pengembangan lebih lanjut skripsi ini di masa yang akan datang.