

OPTIMALISASI ADMINISTRASI BASIS DATA: STRATEGI UNTUK MANAJEMEN DATA YANG EFISIEN

DATABASE ADMINISTRATION OPTIMIZATION: STRATEGIES FOR EFFICIENT DATA MANAGEMENT

Audy Fitri Ariani^{1*}, Dea Puspita Anggraeni¹, Carissa Renatasari¹

*E-mail: audy.fitri10@gmail.com

¹ Sistem Informasi, Ilmu Komputer, UPN “Veteran” Jawa Timur

Abstrak

Penelitian ini membahas tentang optimalisasi administrasi basis data dengan penggunaan replikasi dan fragmentasi untuk manajemen data yang efisien. Tujuan dari penelitian ini adalah untuk meningkatkan efisiensi manajemen data dengan menggunakan teknik replikasi dan fragmentasi. Metode yang digunakan dalam penelitian ini adalah studi literatur dan uji coba. Hasil penelitian menunjukkan bahwa penggunaan replikasi dan fragmentasi meningkatkan efisiensi manajemen data, dengan rata-rata perbedaan keseluruhan operasi (*insert*, *update*, *delete*, dan *select*) sebesar 34.59% lebih cepat. Dengan menggunakan teknik ini, pengguna dapat mengelola data dengan lebih efisien dan mengurangi waktu yang dibutuhkan untuk melakukan tugas-tugas administrasi basis data. Jurnal ini dapat menjadi referensi bagi para pengembang dan administrator basis data yang ingin meningkatkan efisiensi manajemen data mereka.

Kata kunci: administrasi basis data, replikasi, fragmentasi

Abstract

This research discusses the optimization of database administration by using replication and fragmentation for efficient data management. The purpose of this research is to improve the efficiency of data management by using replication and fragmentation techniques. The methods used in this research are literature studies and trials. The results show that the use of replication and fragmentation improves the efficiency of data management, with an average difference in overall operations (insert, update, delete, and select) of 34.59% faster. By using these techniques, users can manage data more efficiently and reduce the time needed to perform database administration tasks. This journal can be a reference for database developers and administrators who want to improve their data management efficiency.

Keywords: database administration, replication, fragmentation.

1. PENDAHULUAN

Dalam era digital yang terus berkembang, administrasi basis data yang efektif dan responsif menjadi semakin penting bagi pengelolaan data organisasi. Penelitian ini bertujuan untuk membahas teknik replikasi dan fragmentasi data sebagai upaya untuk meningkatkan efisiensi manajemen data.

Replikasi dalam sistem basis data terdistribusi berarti menyimpan salinan data di berbagai lokasi fisik atau logis. Sementara itu, fragmentasi memecah data menjadi bagian-bagian yang lebih kecil, sehingga memungkinkan pemrosesan data secara paralel dan efisien. Dalam hal ini, replikasi meningkatkan aksesibilitas dan ketersediaan data [1].

Untuk mencapai tujuan penelitian ini, kami akan menggunakan literatur untuk mempelajari konsep dasar, prinsip, dan keuntungan dari replikasi dan fragmentasi dalam manajemen data.

Selanjutnya, kami akan melakukan uji coba untuk menerapkan teknik replikasi dan fragmentasi data antar *Database* menggunakan Oracle *Database* [2]. Uji coba ini bertujuan untuk mengevaluasi seberapa efektif dan menguntungkan penggunaan teknik replikasi dan fragmentasi dalam meningkatkan efisiensi manajemen.

Dalam rangka mencapai optimasi administrasi basis data, penelitian ini akan difokuskan pada implementasi replikasi dan fragmentasi antar *database* dengan menggunakan *database* Oracle. Diharapkan bahwa dengan menggabungkan kedua pendekatan ini, efisiensi manajemen data dapat ditingkatkan secara signifikan.

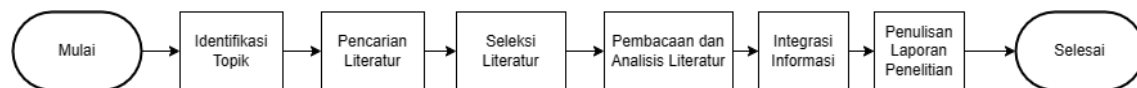
2. METODOLOGI

Penelitian ini menggunakan pendekatan kombinasi antara studi literatur dan eksperimen untuk mencapai tujuan penelitian dalam mengoptimalkan administrasi basis data dengan teknik replikasi dan fragmentasi.

2.1. Studi Literatur

Metodologi studi literatur adalah pendekatan penelitian yang menggunakan sumber sekunder terdahulu yang relevan dengan topik penelitian dari berbagai publikasi ilmiah seperti jurnal, buku, laporan riset, dan sumber literatur lainnya. Pemilihan studi literatur ini karena membantu menemukan landasan teori yang kami angkat mengenai strategi untuk manajemen data yang efisien dan menjadi rujukan yang terpercaya.

Tahapan penelitian studi literatur yang dilakukan dalam penelitian ini meliputi (Gambar 1):



Gambar 1. Alur Tahapan Studi Literatur

Berikut adalah penjelasan alur tahapan studi literatur untuk topik yang kami angkat yaitu strategi untuk manajemen data yang efisien.

1. **Identifikasi Topik** : Identifikasi topik dilakukan melalui pencarian literatur awal dan pembacaan singkat untuk memahami isu-isu utama dalam domain penelitian yaitu mengenai optimalisasi basis data dengan melakukan strategi manajemen data yang efisien.
2. **Pencarian Literatur** : Dilakukan dengan pencarian literatur melalui basis data akademik, perpustakaan digital, jurnal elektronik, dan sumber literatur terpercaya lainnya mengenai topik yang diangkat.
3. **Seleksi Literatur** : Seleksi literatur dilakukan sesuai kriteria inklusi dan eksklusi yang ditetapkan.
4. **Pembacaan dan Analisis Literatur** : Setelah memilih literatur yang relevan, dilakukan pembacaan dan analisis terhadapnya. Artikel-artikel terpilih dibaca secara rinci dan dipahami isinya. Catatan penting dan temuan juga dikumpulkan dan diorganisir.
5. **Integrasi Informasi** : Informasi dan temuan dari artikel-artikel digabungkan dan disusun dalam kerangka konseptual yang kohesif.
6. **Penulisan Laporan Penelitian** : Laporan penelitian ini menyajikan informasi dan pemahaman yang diperoleh secara sistematis dari studi literatur.

Basis Data

basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi. Basis data adalah kumpulan

terorganisir dari data yang saling terkait yang disimpan secara elektronik dalam suatu sistem komputer [3]. Basis data dirancang untuk mengumpulkan, menyimpan, mengelola, dan menyediakan akses terhadap informasi yang relevan. Bahasa Query, seperti SQL (Structured Query Language), digunakan untuk mengambil dan memanipulasi data dalam basis data. Dengan bahasa query, pengguna dapat membuat pertanyaan atau pernyataan untuk mengambil data yang dibutuhkan atau melakukan operasi lainnya, seperti penyisipan, pembaruan, dan penghapusan data [4].

Manajemen Data

Manajemen data adalah serangkaian praktik dan kegiatan yang dilakukan untuk mengelola data dengan efisien dan efektif. Manajemen data meliputi proses pengumpulan, pengorganisasian, penyimpanan, pengolahan, dan pemeliharaan data agar dapat digunakan secara efektif dan efisien [5]. Tujuan dari manajemen data adalah memastikan data tersedia, akurat, aman, terorganisir, dan dapat diakses dengan mudah oleh pengguna yang berwenang. Dengan melaksanakan manajemen data yang efektif, organisasi dapat mengoptimalkan penggunaan data, meningkatkan efisiensi operasional, dan mendukung pengambilan keputusan yang lebih baik[1].

Replikasi Data

Replikasi data merupakan proses pembuatan salinan data yang identik dan mendistribusikannya ke beberapa lokasi. Tujuan dari replikasi data adalah untuk meningkatkan ketersediaan, keandalan, dan skalabilitas basis data [6]. Dengan replikasi data, pengguna dapat mengakses data dengan lebih cepat dan menghindari resiko kehilangan data jika terjadi kegagalan pada satu lokasi. Selain itu, replikasi data juga memungkinkan pemrosesan data secara paralel, yang dapat mengurangi beban dan meningkatkan kinerja sistem basis data. Dalam konteks basis data, replikasi data dapat dilakukan dengan beberapa pendekatan. Salah satu pendekatan umum adalah replikasi penuh, di mana seluruh basis data disalin ke setiap server replika [7]. Pendekatan lainnya adalah replikasi sebagian, di mana hanya sebagian data yang direplikasi ke setiap server.

Fragmentasi Data

Fragmentasi data adalah proses membagi data menjadi fragmen yang lebih kecil berdasarkan kriteria tertentu, seperti atribut atau lokasi. Dengan fragmentasi data, pengolahan data dapat dilakukan secara terdistribusi dan paralel, yang berpotensi meningkatkan efisiensi dan kinerja sistem basis data [7]. Selain itu, fragmentasi data juga dapat mendukung skalabilitas dan pemisahan tugas dalam lingkungan yang kompleks.

Terdapat dua jenis fragmentasi data yang umum digunakan yaitu fragmentasi horizontal dan fragmentasi vertikal. Fragmentasi horizontal melibatkan pembagian baris data menjadi fragmen yang lebih kecil. Jenis fragmentasi ini berguna ketika terdapat beban kerja yang besar pada tabel atau entitas tertentu karena dapat mengurangi konflik akses dan mempercepat eksekusi kueri. Fragmentasi vertikal melibatkan pemisahan kolom data ke dalam fragmen yang berbeda [8]. setiap fragmen berisi sejumlah kolom data yang relevan atau memiliki karakteristik yang serupa. dengan jenis fragmentasi ini, hanya kolom yang dibutuhkan oleh kueri atau aplikasi tertentu yang diakses, mengurangi jumlah data yang harus dipindahkan dan mempercepat waktu respon kueri.

Efisiensi Manajemen Data

Efisiensi dalam manajemen data berfokus pada penggunaan sumber daya yang optimal untuk mengelola data dengan cara yang hemat waktu, biaya, dan upaya. Tujuan utamanya adalah meningkatkan produktivitas, kualitas, dan ketersediaan data, sambil mengurangi pemborosan dan kelebihan dalam proses manajemen data [9]. Pengelolaan data dapat dikatakan telah berjalan efisien ketika beberapa kriteria berikut terpenuhi:

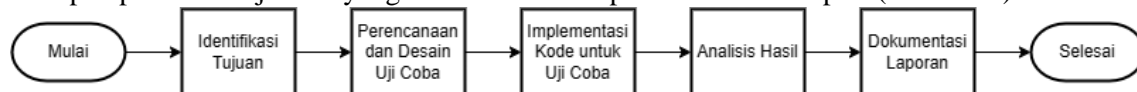
- 1) Ketersediaan data yang tinggi: Data dapat diakses dengan mudah dan tersedia untuk pengguna yang berwenang ketika dibutuhkan. Sistem penyimpanan dan manajemen data yang efisien memastikan bahwa data tersedia secara konsisten dan dapat diakses dengan waktu respons yang cepat.
- 2) Kualitas data yang tinggi: Data dikelola dengan baik dan memiliki kualitas yang tinggi. Ini berarti data akurat, terbaru, lengkap, dan konsisten. Manajemen data yang efisien mencakup pembersihan dan pemeliharaan data secara teratur untuk memastikan data tetap relevan dan bebas dari kesalahan.
- 3) Proses yang terotomatisasi: Proses-proses manajemen data yang dapat diotomatisasi telah diimplementasikan. Ini termasuk pengumpulan data, pembaruan data, integrasi data, dan pemrosesan data rutin lainnya. Otomatisasi mengurangi intervensi manusia yang memakan waktu dan memungkinkan penanganan data yang lebih cepat dan konsisten.
- 4) Penggunaan sumber daya yang efisien: Manajemen data yang efisien mengoptimalkan penggunaan sumber daya seperti penyimpanan, komputasi, dan jaringan. Data disimpan dan diproses dengan cara yang hemat sumber daya sehingga mengurangi biaya operasional dan memaksimalkan efisiensi.
- 5) Kinerja yang baik: Sistem basis data dan proses manajemen data memberikan kinerja yang baik. Query dan pemrosesan data dilakukan dengan cepat, waktu respons yang rendah, dan throughput yang tinggi. Penggunaan indeks, optimasi kueri, dan pemantauan kinerja sistem membantu memastikan kinerja yang optimal.

Efisiensi pengelolaan data memberikan manfaat dalam hal produktivitas, pengambilan keputusan yang lebih baik, dan pengurangan biaya operasional sehingga operasional perusahaan dapat berjalan optimal [10].

2.2. Uji Coba

Metodologi uji coba atau eksperimen adalah pendekatan penelitian dengan melakukan praktik langsung dan iterasi untuk menguji dan meningkatkan penelitian yang ingin dicapai. Pemilihan metodologi uji coba ini karena dapat membantu mengukur dan mengamati hasil secara objektif, meningkatkan keterampilan teknis dan mendorong perbaikan dan pengembangan berkelanjutan melalui iterasi dan umpan balik [1]

Tahapan penelitian uji coba yang dilakukan dalam penelitian ini meliputi (Gambar 2):



Gambar 2. Alur Tahapan Uji Coba

Berikut adalah penjelasan alur tahapan uji coba untuk topik yang kami angkat yaitu strategi untuk manajemen data yang efisien.

Berikut adalah langkah-langkah umum dalam metodologi eksperimen Query:

1. Identifikasi: Menentukan tujuan penelitian dengan jelas, yaitu untuk mengoptimalkan administrasi basis data dengan menggunakan replikasi dan fragmentasi untuk manajemen data yang efisien.
2. Perencanaan dan desain uji coba: Merencanakan dan mendesain eksperimen dengan hati-hati. Hal ini meliputi pemilihan variabel yang akan diamati, pemilihan data, aplikasi, dan teknik yang akan digunakan, serta menetapkan metrik evaluasi untuk mengukur keberhasilan. Tujuan perencanaan ini adalah untuk memastikan bahwa uji coba dilakukan secara sistematis dan dapat menghasilkan data yang valid.

3. Implementasi kode untuk uji coba: Melakukan implementasi praktis dengan menulis kode yang diperlukan untuk menjalankan uji coba. Selama tahap ini, penting untuk mengikuti prinsip-prinsip praktik yang baik agar hasil yang diperoleh dapat diandalkan dan dapat direplikasi oleh orang lain.
4. Analisis hasil: Menganalisis hasil yang dihasilkan dari uji coba. Proses ini melibatkan pengolahan data yang telah dikumpulkan, mengidentifikasi temuan menarik, dan melakukan analisis statistik atau kualitatif sesuai kebutuhan. Tujuan analisis hasil ini adalah untuk memahami dampak penggunaan teknik replikasi dan fragmentasi terhadap efisiensi manajemen data.
5. Dokumentasi laporan: Menulis laporan penelitian yang merangkum semua langkah-langkah, hasil, dan kesimpulan penelitian. Laporan ini meliputi deskripsi detail tentang tujuan penelitian, metodologi yang digunakan, hasil yang diperoleh, dan kesimpulan yang dapat diambil dari penelitian tersebut. Dokumentasi laporan penting agar penelitian dapat dipahami dan direplikasi oleh orang lain serta menjadi referensi bagi pengembang dan administrator basis data yang tertarik untuk meningkatkan efisiensi manajemen data mereka.

Dalam melakukan uji coba software yang digunakan yaitu (Tabel 1):

Tabel 1. Tools dan Software Uji Coba

Tools dan Software yang digunakan	
1.	SQL Developer
2.	Oracle 10g Enterprise Edition

3. HASIL DAN PEMBAHASAN

Penelitian ini akan membahas mengenai implementasi praktik penggunaan replikasi dan fragmentasi data dalam suatu sistem administrasi basis data serta langkah-langkah yang dilakukan untuk menerapkan teknik replikasi, dan fragmentasi data antar *database*. Akan dilakukan pengamatan hasil pengujian kinerja untuk menilai efisiensi dan keberhasilan dari penggunaan replikasi dan fragmentasi dalam meningkatkan performa, ketersediaan, dan pengelolaan data dengan membandingkan respon waktu untuk melakukan eksekusi terhadap perintah query *Insert*, *Update*, *Delete*, dan *Select*.

3.1. Implementasi Replikasi dan Fragmentasi Data

Dalam uji coba replikasi antar *database* di satu laptop, kami mereplikasi tabel "mahasiswa" dengan atribut sesuai pada Tabel 2.

Tabel 2. Atribut tabel Mahasiswa

Mahasiswa	
nim	VARCHAR2 (10)
nama	VARCHAR2 (50)
jenis_kelamin	VARCHAR2 (10)
tanggal_lahir	DATE
alamat	VARCHAR2 (100)
email	VARCHAR2 (50)
telepon	VARCHAR2 (15)
program_studi	VARCHAR2 (50)
semester	NUMBER (2)
ipk	NUMBER (3, 2)

"Orcl" adalah sumber replikasi atau *database* utama, dan "db_replika" adalah *database* target. Data dari tabel "Mahasiswa" pada *database* utama "orcl" akan disalin dan diperbanyak pada *database* target "db_replika" selama proses replikasi, yang memungkinkan untuk mempertahankan dan memperbarui informasi siswa secara konsisten antara kedua *database*.

3.1.1. Replikasi Data

- Database Links

Penggunaan *database* link pada Oracle *Database* sangat penting untuk menghubungkan *database* pusat, yang menggunakan *database* "Orcl" sebagai basis datanya, dengan target replikasi, yaitu *database* "db_replika", dalam proses replikasi data. Konfigurasi *database* link memungkinkan transfer data yang terstruktur dan sinkronisasi secara teratur atau secara real-time. Dalam kasus ini, *database* pusat (ORCL) berfungsi sebagai sumber data utama, sedangkan *database* "db_replika" berfungsi sebagai sumber data sekunder.

Query pada Tabel 3 digunakan untuk membuat sebuah *database* link yang menghubungkan antara dua *database*, yaitu *database* "Orcl" dan *database* "Db_replika".

Tabel 3. Query Database Link

<i>From Database Orcl to Database Db_replika</i>	<i>From Database Db_replika to Database Orcl</i>
<pre>CREATE DATABASE LINK "DBREPLIKA_HR_LINK.REGRESS.RDBMS.DEV. US.ORACLE.COM" CONNECT TO "ODY" IDENTIFIED BY VALUES '055168D46DC7C1E50480B0595AD79B7687' USING 'db_repli'</pre>	<pre>CREATE DATABASE LINK "ORCL_HR_LINK.REGRESS.RDBMS.DEV.US.OR ACLE.COM" CONNECT TO "HR" IDENTIFIED BY VALUES '052E076E33046CEC53BAD9E2EA3C078227' USING 'orcl'</pre>

Database link memungkinkan pemantauan dan pengaturan data replikasi yang lebih baik. Dengan menggunakan *database* link, dapat melakukan akses dan mengambil data yang relevan dari *database* pusat serta mentransfernya secara efisien ke *database* target replikasi. Selain itu, *database* link memungkinkan penggunaan perintah *Insert*, *Update*, *Delete*, dan *Select* yang diperlukan untuk memastikan bahwa data selaras antara *database* pusat dan replikasi.

- Trigger

Penggunaan trigger *insert*, *update*, dan *delete* pada *database* utama sangat penting untuk replikasi data dalam *database* Oracle. Trigger ini dibuat untuk memantau perubahan data di *database* utama dan secara otomatis mereplikasi perubahan ke *database* target. Pada Tabel 4 merupakan query yang digunakan untuk membuat trigger *update*, *insert*, *delete* pada *database* utama (orcl) untuk melakukan replikasi pada *database* target (db_replika)

Tabel 4. Query Trigger Update, Insert, Delete

<i>Trigger Insert</i>	<i>Trigger Update</i>	<i>Trigger Delete</i>
<pre>create or replace trigger after_insert after insert on mahasiswa for each row BEGIN INSERT INTO mahasiswa@dbreplika_hr_l ink (nim, nama,jenis_kelamin,tangg</pre>	<pre>create or replace TRIGGER after_update AFTER UPDATE ON mahasiswa for each row BEGIN UPDATE mahasiswa@dbreplika_hr_l ink SET nama = :new.nama, jenis_kelamin =</pre>	<pre>create or replace TRIGGER after_delete AFTER DELETE ON mahasiswa for each row BEGIN delete from mahasiswa@dbreplika_hr_l ink where nim = :old.nim; END;</pre>


```
al_lahir, alamat, email, :new.jenis_kelamin,  
telepon, program_studi, tanggal_lahir =  
semester, ipk) :new.tanggal_lahir,  
VALUES alamat =  
(:new.nim, :new.nama, :new.alamat,  
:new.jenis_kelamin, email = :new.email,  
:new.tanggal_lahir, telepon =  
:new.alamat, :new.email, :new.telepon,  
:new.telepon, program_studi =  
:new.program_studi, :new.program_studi,  
:new.semester, semester =  
:new.ipk); :new.semester,  
END; ipk = :new.ipk  
WHERE nim = :new.nim;  
END;
```

Dalam replikasi data menggunakan oracle *database*, proses replikasi data pada *database* target akan berjalan apabila operasi **commit** telah dilakukan pada *database* utama. Perubahan data yang disebabkan oleh *trigger* akan dimasukkan kedalam *database* target. *Trigger Insert* akan menambah data baru kedalam tabel yang sesuai di *database* target, *trigger update* akan mereplikasi perubahan pada data yang telah diubah di *database* utama ke tabel yang berkorespondensi di *database* target, dan *Trigger delete* akan menghapus data dari tabel di *database* target yang telah dihapus dari tabel di *database* utama.

3.1.2. Fragmentasi Data

Dalam uji coba penelitian ini, penerapan partition fragmentation pada *database* target dapat meningkatkan efisiensi data [11]. Dengan fragmentasi partisi, data pada tabel dapat dipecah menjadi beberapa bagian atau partisi yang lebih kecil. Pembuatan partisi fragmentasi dilakukan ketika ingin membuat tabel yang menjadi target pada saat replikasi. Untuk melakukan Fragmentasi partisi data pada *database* replikasi, digunakanlah query pada Tabel 5 yang dituliskan pada *database* target (db_replika).

Tabel 5. Partition Fragmentation

Partition Fragmentation

```
CREATE TABLE mahasiswa (  
  nim VARCHAR2(10),  
  nama VARCHAR2(50),  
  jenis_kelamin VARCHAR2(10),  
  tanggal_lahir DATE,  
  alamat VARCHAR2(100),  
  email VARCHAR2(50),  
  telepon VARCHAR2(15),  
  program_studi VARCHAR2(50),  
  semester NUMBER(2),  
  ipk NUMBER(3, 2)  
)  
PARTITION BY RANGE (ipk)  
(  
  PARTITION part_mahasiswa_1 VALUES LESS THAN (2.50),  
  PARTITION part_mahasiswa_2 VALUES LESS THAN (3.00),  
  PARTITION part_mahasiswa_3 VALUES LESS THAN (3.50),  
  PARTITION part_mahasiswa_4 VALUES LESS THAN (MAXVALUE)  
) ;
```

3.2. Hasil Uji Coba Penggunaan Replikasi dan Fragmentasi Data

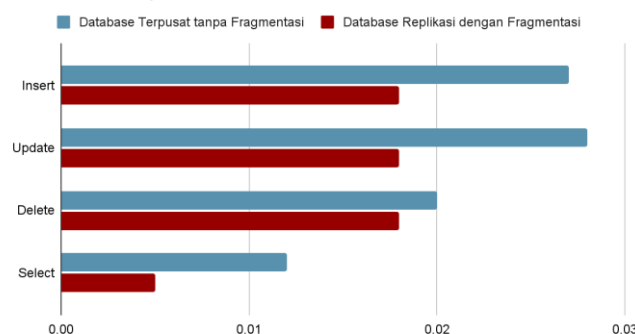
Tabel 3. Hasil Uji Coba Database Terpusat

proses yg dilakukan	proses query			
	Database terpusat tanpa <i>partition fragmentation</i>			
	1	2	3	rata - rata
<i>Insert</i>	0.029	0.025	0.027	0.027
<i>Update</i>	0.031	0.031	0.023	0.028
<i>Delete</i>	0.017	0.24	0.018	0.020
<i>Select</i>	0,018	0,011	0,008	0.012

Tabel 4. Hasil Uji Coba Database Replikasi

proses yg dilakukan	Proses query			
	Database replikasi dengan <i>partition fragmentation</i>			
	1	2	3	rata - rata
<i>Insert</i>	0.021	0.015	0.017	0.018
<i>Update</i>	0.017	0.018	0.018	0.018
<i>Delete</i>	0.019	0.019	0.017	0.018
<i>Select</i>	0,006	0,005	0,005	0.005
Rata - rata				34.59%

Rata - rata respon waktu



Gambar 3. Diagram Perbandingan Rata - Rata Waktu Respon

Berdasarkan uji coba yang telah dilakukan dalam penelitian ini, hasilnya menunjukkan bahwa pada *database* terpusat tanpa fragmentasi, terdapat peningkatan waktu eksekusi untuk operasi insert, update, delete, dan select. Rata-rata waktu yang dibutuhkan adalah 0.027 detik untuk operasi insert, 0.028 detik untuk operasi update, 0.020 detik untuk operasi delete, dan 0.012 detik untuk operasi select. Hasil ini menunjukkan bahwa *database* terpusat tanpa fragmentasi memerlukan waktu lebih lama dalam melakukan operasi data.

Sedangkan dalam *database* replikasi dengan fragmentasi, hasil uji coba menunjukkan peningkatan dalam performa operasi data. Waktu yang diperlukan untuk operasi insert, update, delete, dan select menjadi lebih cepat. Operasi insert hanya memerlukan waktu rata-rata 0.018 detik, sedangkan operasi update, delete, dan select masing-masing hanya memerlukan waktu rata-rata 0.018 detik dan 0.005 detik. Perbedaan waktu yang lebih rendah menunjukkan bahwa replikasi data berhasil mengurangi waktu eksekusi operasi data dengan efektif.

Penelitian ini menunjukkan bahwa penggunaan replikasi dan fragmentasi dalam manajemen data menghasilkan rata-rata perbedaan waktu eksekusi operasi sebesar 34.59% lebih cepat. Perbedaan waktu yang lebih rendah ini secara efektif meningkatkan efisiensi manajemen data. Dengan menerapkan teknik replikasi dan fragmentasi, pengguna dapat mengelola data dengan lebih efisien, mempercepat operasi data, dan mengurangi waktu yang diperlukan untuk tugas-tugas

administrasi basis data. Hasil penelitian ini memberikan bukti nyata bahwa penggunaan replikasi dan fragmentasi dalam manajemen data memberikan manfaat yang signifikan bagi organisasi. Temuan ini dapat menjadi acuan penting bagi pengembang dan administrator basis data yang ingin meningkatkan efisiensi manajemen data mereka.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil uji coba yang telah dilakukan, dapat disimpulkan bahwa penggunaan teknik replikasi dan fragmentasi dalam administrasi basis data dapat meningkatkan efisiensi manajemen data. Pada *database* terpusat tanpa fragmentasi, terjadi peningkatan waktu eksekusi untuk operasi *Insert*, *Update*, *Delete*, dan *Select*. Hal ini menunjukkan bahwa pengelolaan data pada *database* terpusat tanpa fragmentasi memerlukan waktu yang lebih lama. Namun, pada *database* replikasi dengan fragmentasi, terlihat peningkatan performa dengan waktu eksekusi yang lebih cepat untuk operasi data tersebut. Rata-rata perbedaan keseluruhan dari operasi *Insert*, *Update*, *Delete*, dan *Select* adalah sebesar 34.59% lebih cepat. Hasil ini menunjukkan bahwa penggunaan replikasi data dan fragmentasi secara efektif mengurangi waktu eksekusi operasi data.

Saran yang dapat diberikan berdasarkan penelitian ini adalah organisasi dan pengembang basis data sebaiknya mempertimbangkan penggunaan teknik replikasi dan fragmentasi dalam administrasi basis data mereka untuk meningkatkan efisiensi manajemen data. Perencanaan yang hati-hati, pemilihan teknik yang sesuai dengan kebutuhan, implementasi yang baik, dan mengikuti perkembangan terbaru dalam teknologi basis data terdistribusi juga penting untuk mengoptimalkan administrasi basis data.

5. DAFTAR RUJUKAN

- [1] M. Mashuri, "Implementasi Sistem Database Terdistribusi Dengan Metode Partial Replication," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 3, no. 2, pp. 113–120, Sep. 2020, doi: 10.31539/intecom.v3i2.1582.
- [2] "Oracle Database Advanced Replication".
- [3] Huawei Technologies Co., Ltd., *Database Principles and Technologies – Based on Huawei GaussDB*. Singapore: Springer Nature Singapore, 2023. doi: 10.1007/978-981-19-3032-4.
- [4] A. Ubaidillah, "PERANCANGAN DAN IMPLEMENTASI SISTEM DATABASE TERDISTRIBUSI MENGGUNAKAN ORACLE".
- [5] S. Shafiee, S. C. Friis, L. Lis, U. Harlou, Y. Wautelet, and L. Hvam, "A Database Administration Tool to Model the Configuration Projects," in *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Bangkok: IEEE, Dec. 2018, pp. 341–345. doi: 10.1109/IEEM.2018.8607654.
- [6] A. Satria Ajinusa, A. B. Murti Wijaya, and A. Aziz, "PEMBANGUNAN SISTEM INFORMASI PRESENSI PERKULIAHAN MENGGUNAKAN BASIS DATA TERDISTRIBUSI DENGAN METODE REPLIKASI-ASYNCHRONOUS," *J. Teknol. Inf. ITS smart*, vol. 4, no. 2, p. 48, Sep. 2016, doi: 10.20961/its.v4i2.1760.
- [7] H. Murti and E. Lestariningsih, "PENDEKATAN REPLIKASI DAN FRAGMENTASI DALAM BASIS DATA TERDISTRIBUSI UNTUK PENYIMPANAN TABEL BASIS DATA FUZZY," 2018.
- [8] V. K. Tiwari and HCL Tech, Texas, United States, "Oracle Database Backup Testing," *Int. J. Trend Sci. Res. Dev.*, vol. Volume-2, no. Issue-3, pp. 2043–2044, Apr. 2018, doi: 10.31142/ijtsrd11572.
- [9] P. Petrov, I. Kuyumdzhev, R. Malkawi, G. Dimitrov, and O. Bychkov, "Database Administration Practical Aspects in Providing Digitalization of Educational Services," *Int.*

- J. Emerg. Technol. Learn. IJET*, vol. 17, no. 20, pp. 274–282, Oct. 2022, doi:
10.3991/ijet.v17i20.32785.
- [10] M. N. Gedam and B. B. Meshram, “Database Private Security Jurisprudence: A Case Study using Oracle,” *Int. J. Database Manag. Syst.*, vol. 13, no. 03, pp. 01–21, Jun. 2021, doi:
10.5121/ijdms.2021.13301.