JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 1842-1854

EVALUASI KINERJA NON-CRYPTOGRAPHIC HASH FUNCTIONS (NCHFS) DALAM SINKRONISASI DATA PADA DBMS ORACLE DAN POSTGRES

Caroluce Ricky Harkris Nowo*[1)], Pratyaksa Ocsa Nugraha Saian[2)]
1. Universitas Kristen Satya Wacana, Indonesia
2. Universitas Kristen Satya Wacana, Indonesia

Article Info
Kata Kunci: Evaluasi Kinerja; NCHFs; Sinkronisasi Data

Keywords: Performance Evaluation; NCHFs; Data Synchronization

Article history:

Received 29 September 2024
Revised 13 Oktober 2024
Accepted 4 November 2024
Available online 4 December 2024

DOI :
https://doi.org/10.29100/jipi.v9i4.5531

* Corresponding author.
Caroluce Ricky Harkris Nowo
E-mail address:
672020191@student.uksw.edu

ABSTRAK
Dalam lingkungan bisnis modern, informasi dan data memiliki peran penting dalam pengambilan keputusan serta operasional perusahaan. Salah satu tantangan yang dihadapi adalah sinkronisasi data, terutama bagi perusahaan besar seperti PT XYZ, yang merupakan salah satu pemain terbesar industri ritel di Indonesia. Permasalahan muncul karena infrastruktur teknologi informasi mereka sedang dalam proses pergantian dan belum dapat sepenuhnya digantikan oleh yang lebih modern. Oleh karena itu, diperlukan proses sinkronisasi yang efektif untuk memungkinkan integrasi yang lancar antara perangkat lunak yang lama dan yang baru. Penelitian ini bertujuan untuk mengoptimalkan sinkronisasi data antara DBMS Oracle dan PostgreSQL dengan menggunakan Non-Cryptographic Hash Functions (NCHFs). Sebanyak tujuh NCHFs dievaluasi dalam eksperimen, dengan mempertimbangkan waktu eksekusi, penggunaan sumber daya, serta keberhasilan dan akurasi sinkronisasi. Hasil penelitian menunjukkan bahwa MetroHash secara konsisten memberikan kinerja terbaik. Dibandingkan dengan fungsi hash kriptografis standar seperti SHA-256, MetroHash berhasil mengoptimalkan waktu eksekusi hingga 29% lebih cepat. Ketika dibandingkan dengan NCHFs terbaik lainnya, seperti CityHash, MetroHash masih unggul dengan selisih waktu eksekusi sebesar 5%. Namun, pilihan penggunaannya harus disesuaikan dengan kebutuhan aplikasi dan situasi perusahaan. Selain MetroHash, CityHash juga merupakan alternatif yang layak dipertimbangkan. Di sisi lain, FNV-1a tidak disarankan karena kinerjanya yang kurang memuaskan.

ABSTRACT
In the modern business environment, information and data play a crucial role in decision-making and company operations. One of the challenges faced is data synchronization, especially for large companies like PT XYZ, which is one of the biggest players in the retail industry in Indonesia. Issues arise because their information technology infrastructure is currently undergoing a transition and cannot yet be fully replaced by more modern systems. Therefore, an effective synchronization process is needed to facilitate smooth integration between old and new software. This research aims to optimize data synchronization between DBMS Oracle and PostgreSQL using Non-Cryptographic Hash Functions (NCHFs). Seven NCHFs were evaluated in experiments, considering execution time, resource usage, as well as synchronization success and accuracy. The research results indicate that MetroHash consistently delivers the best performance. Compared to standard cryptographic hash functions like SHA-256, MetroHash successfully optimizes execution time up to 29% faster. When compared to other top NCHFs, such as CityHash, MetroHash still outperforms with a margin of 5% in execution time difference. However, its usage should be tailored to the application needs and company situation. Besides MetroHash, CityHash is also a viable alternative to consider. On the other hand, FNV-1a is not recommended due to its unsatisfactory performance.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 1842-1854

I. PENDAHULUAN

DALAM era modern di mana informasi dan data memiliki peran penting dalam pengambilan keputusan dan operasi bisnis, sinkronisasi data menjadi aspek yang sangat penting dalam mengelola basis data [1]. *Sinkronisasi* merupakan proses untuk menjaga agar data tetap konsisten di beberapa perangkat atau sistem. Hal ini penting untuk memastikan bahwa semua orang memiliki informasi terbaru, terutama saat data sering mengalami perubahan. Proses ini dapat dilakukan dengan berbagai metode, tergantung pada kebutuhan aplikasi atau sistem yang ada [2]. *Sinkronisasi data* dapat dilakukan antara database yang homogen (jenis yang sama) atau heterogen (jenis yang berbeda) [3]. *Sinkronisasi data* memiliki banyak manfaat, seperti meningkatkan ketersediaan, reliabilitas, dan efisiensi data, serta memfasilitasi integrasi dan kolaborasi antara sistem yang berbeda.

Sinkronisasi data bukanlah tugas yang sederhana. Proses ini dihadapkan pada tantangan inkonsistensi data, keamanan data, dan performa data dimana semua konflik data harus diselesaikan [4]. *Konflik data* terjadi ketika ada perubahan yang saling bertentangan pada data yang disinkronkan, dan inkonsistensi data muncul saat ada perbedaan antara data sumber dan data tujuan [5], [6]. *Performa data* berhubungan dengan kecepatan dan akurasi selama proses sinkronisasi data. PT XYZ merupakan salah satu perusahaan ritel terbesar di Indonesia yang telah memperluas jaringannya dan menjadi salah satu toko ritel yang paling mudah diakses. Meskipun demikian, selama proses transisi teknologi, perusahaan menghadapi sejumlah tantangan yang terkait dengan sinkronisasi data. Masalah muncul karena perangkat lunak lama yang menggunakan *Oracle Forms* tidak dapat sepenuhnya digantikan oleh perangkat lunak baru berbasis web yang mulai diterapkan oleh perusahaan. Kondisi ini mewajibkan perusahaan untuk melakukan sinkronisasi data secara rutin, namun sayangnya, proses ini memakan waktu yang signifikan, berkisar antara 1 hingga 5 jam untuk setiap iterasi sinkronisasi. Oleh karena itu, diperlukan pengembangan yang dapat mengoptimalkan proses sinkronisasi data untuk mengatasi tantangan ini secara efisien.

Salah satu faktor yang mempengaruhi performa data dalam sinkronisasi adalah fungsi hashing. *Fungsi hashing* adalah fungsi yang digunakan untuk menerima data dan mengubahnya menjadi nilai hash yang unik dan tetap [7], [8]. Nilai hash ini dapat digunakan untuk membandingkan dan mengidentifikasi data dengan cepat dan mudah. Fungsi hash memiliki banyak varian dan bisa dikategorikan menjadi dua kategori besar, yaitu fungsi hash kriptografi dan non-kriptografi. *Fungsi hash non-kriptografi* (NCHFs) merupakan fungsi yang digunakan untuk mengubah data menjadi nilai hash, perbedaan dengan fungsi hashing biasa NCHFs tidak memperhatikan keamanan sehingga bisa lebih cepat dan menghemat sumber daya [9], [10]. NCHFs memiliki peran penting dalam mengoptimalkan performa sinkronisasi data, dimana data dapat diubah menjadi nilai hash dengan cepat, sehingga memperpendek proses identifikasi dan perbandingan data. NCHFs memiliki berbagai algoritma seperti *CityHash*, *SpookyHash, FNV-1a, MurmurHash3, MetroHash, FarmHash*, dan *XXHash*. Setiap algoritma memiliki kelebihan dan kekurangannya masing-masing. Sebagai contoh, *CityHash* dikenal karena memiliki performa yang tinggi, sementara *MurmurHash3* memiliki keunggulan dalam distribusi data yang baik. Oleh karena itu, pemilihan algoritma hashing yang tepat sangat penting untuk memastikan efisiensi dan keakuratan dalam proses sinkronisasi.

Dalam konteks penelitian ini, fokus penelitian terletak pada optimalisasi sinkronisasi data pada *Database Management System* (DBMS) Oracle dan Postgres dengan pemanfaatan NCHFs. Oracle merupakan DBMS yang dikembangkan oleh Oracle Corporation dan merupakan salah satu DBMS komersial paling populer di dunia [11]. Postgres merupakan DBMS open-source yang dikembangkan oleh University of California. Postgres mendukung sebagian besar standar SQL dan menyediakan banyak fitur modern [12]. Kedua jenis DBMS ini digunakan secara luas di berbagai industri, termasuk PT.XYZ yang memiliki basis data Oracle dan Postgres di lokasi yang berbeda dan perlu melakukan sinkronisasi data secara berkala. Kecepatan dan akurasi dari berbagai fungsi hash non-kriptografi dalam proses sinkronisasi akan diukur untuk memperoleh fungsi hashing terbaik dalam kasus ini.

Pada penelitian oleh Sateesan dkk telah diteliti pengembangan algoritma hash non-kriptografis yang dioptimalkan untuk digunakan dalam perangkat keras. Mereka berfokus pada menciptakan algoritma-algoritma baru yang berlandaskan pada desain *Xoodoo-NC*, sebuah fungsi hash non-kriptografis. Pendekatan ini melibatkan penggabungan beberapa putaran sandi kunci simetris untuk meningkatkan kinerja algoritma, sambil tetap mempertahankan sifat-sifat perubahan total dari fungsi tersebut. Penelitian tersebut berfokus pada pengoptimalan perangkat keras dengan memanfaatkan NCHFs yang dikembangkan, berbeda dengan penelitian yang dilakukan di PT.XYZ yang memanfaatkan NCHFs untuk pengoptimalan proses sinkronisasi database [13].

Di sisi lain, dalam konteks metode sinkronisasi data pada DBMS yang berbeda, terdapat penelitian yang dilakukan oleh Ayu Helinda dkk. Mereka mengkaji permasalahan sinkronisasi data antar DBMS yang berbeda jenis. Metode yang mereka gunakan adalah dengan memanfaatkan algoritma MD5 untuk membandingkan nilai hash data di tabel sumber dengan data di tabel tujuan. Namun, penelitian ini mengungkapkan bahwa algoritma

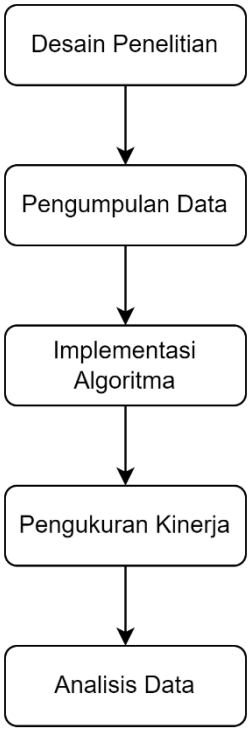
MD5 memiliki kelemahan dalam hal waktu sinkronisasi, terutama ketika diterapkan pada basis data dengan volume data yang besar. Waktu eksekusi proses sinkronisasi MD5 secara signifikan berkaitan dengan jumlah data rekaman yang perlu disinkronisasi. Oleh karena itu, penelitian ini mendorong perlunya penelitian lebih lanjut untuk meningkatkan efisiensi metode sinkronisasi, terutama pada basis data yang besar dan heterogen. Penelitian tersebut memiliki kemiripan dengan penelitian yang dilakukan oleh PT.XYZ dimana keduanya sama-sama memanfaatkan fungsi hashing untuk mengoptimalkan perbandingan data pada proses sinkronisasi, perbedaannya terletak pada jenis fungsi hashing yang diterapkan [14].

Penelitian Sateesan dkk mengeksplorasi optimasi algoritma hash non-kriptografis dengan fokus pada desain *Xoodoo-NC* untuk implementasi perangkat keras. Meskipun penelitian tersebut tidak secara langsung terkait dengan sinkronisasi data pada DBMS, namun memberikan pemahaman yang lebih dalam tentang kinerja NCHFs. Dengan demikian, penelitian ini melengkapi penelitian sebelumnya dengan mengaplikasikan konsep dan pemahaman tentang NCHFs ke dalam konteks sinkronisasi data antar DBMS. Penelitian Ayu Helinda dkk mengevaluasi sinkronisasi data antar DBMS menggunakan algoritma MD5, memberikan pemahaman terhadap tantangan dalam sinkronisasi data, terutama dalam konteks basis data yang besar dan heterogen. Penelitian ini menyumbang pemahaman yang lebih dalam dengan memperluas cakupan penelitian sebelumnya dengan mempertimbangkan berbagai NCHFs sebagai alternatif untuk meningkatkan efisiensi proses sinkronisasi data. Hal ini mengindikasikan bahwa solusi tidak hanya bergantung pada penggunaan algoritma MD5, tetapi juga mengeksplorasi pilihan lain yang dapat meningkatkan efektivitas sinkronisasi data secara keseluruhan.

Temuan dari penelitian-penelitian sebelumnya, seperti yang dilakukan oleh Sateesan dkk yang mengoptimalkan algoritma hash non-kriptografis untuk implementasi perangkat keras, dan Ayu Helinda dkk yang mengevaluasi sinkronisasi data antar DBMS dengan menggunakan algoritma MD5, dapat memberikan wawasan tambahan terkait optimalisasi algoritma hash dalam konteks sinkronisasi data. Dengan merangkum hasil dan kesimpulan dari penelitian-penelitian tersebut, diharapkan penelitian ini dapat memberikan kontribusi yang substansial dalam memperdalam pemahaman dan meningkatkan efisiensi pada proses sinkronisasi data pada DBMS Oracle dan Postgres.

II. METODE PENELITIAN

Metodologi penelitian ini terdiri dari beberapa tahap antara lain Desain Penelitian, Pengumpulan Data, Implementasi Algoritma, Pengukuran Kinerja dan Analisis Data. Alur penelitian dapat dilihat pada Gambar 1.

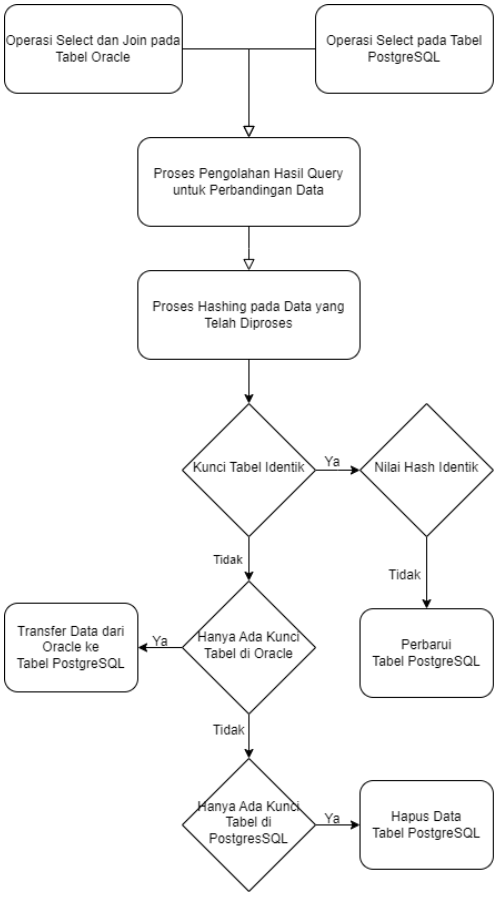


Gambar. 1. Tahapan Penelitian

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 1842-1854

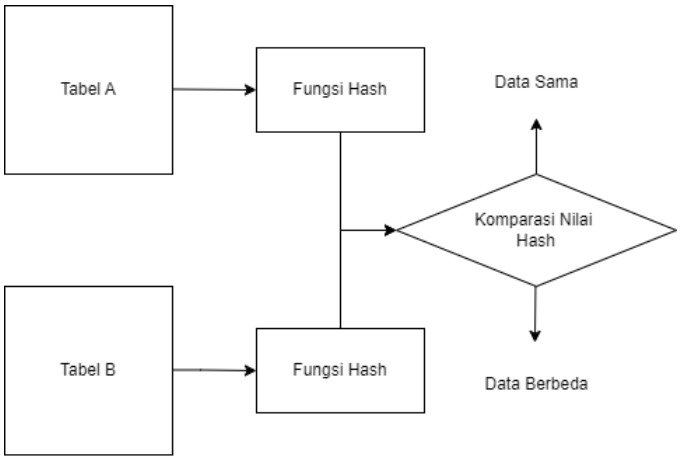
A. Desain Penelitian

Eksperimen ini dilakukan dengan tujuan untuk mengevaluasi kinerja *Non-Cryptographic Hash Functions* (NCHFs) dalam sinkronisasi data antara DBMS Oracle dan PostgreSQL. Eksperimen ini akan dirancang sebagai eksperimen komparatif.



Gambar. 2. Proses Sinkronisasi Data

Proses sinkronisasi melibatkan penggunaan NCHFs untuk mempercepat proses komparasi dan pencarian data. Setiap baris data dari tabel di kedua database diambil, kemudian diolah agar menghasilkan data yang serupa. Data yang telah diolah kemudian dikonversi menjadi nilai hash menggunakan NCHFs. Nilai hash dari data tersebut kemudian dibandingkan untuk mencari baris data yang berbeda. Proses sinkronisasi ini hanya dilakukan satu arah [15], yaitu perubahan pada database Oracle akan disinkronkan ke database PostgreSQL. Aplikasi baru yang menggunakan database PostgreSQL milik PT XYZ tidak memerlukan sinkronisasi bolak-balik karena aplikasi tersebut sudah mendukung proses sinkronisasi otomatis ke database Oracle setiap kali terjadi perubahan data.



Gambar. 3. Komparasi Data Dengan Hashing

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 1842-1854

B. *Pengumpulan Data*

Proses pengumpulan data dengan mempertimbangkan berbagai hal antara lain:

1. Identifikasi Jenis Data yang Dibutuhkan
   Analisis awal dilakukan dengan identifikasi jenis data yang diperlukan untuk eksperimen. Proses identifikasi dilakukan dengan menentukan tabel, kolom, dan relasi data yang terdapat dalam aplikasi bisnis yang menggunakan sistem manajemen basis data (DBMS) Oracle dan PostgreSQL. Dalam konteks ini, data yang digunakan dalam pengujian terdiri dari lima tabel, dengan jumlah data berkisar antara 15 ribu hingga 176 ribu entri.
2. Sumber Data
   Dalam penelitian ini, data yang digunakan berasal dari PT XYZ, yang merupakan sumber data utama untuk eksperimen. Data ini mencerminkan jenis data yang umumnya ada dalam aplikasi bisnis yang menggunakan DBMS Oracle dan PostgreSQL.
3. Data Pengujian dan Data Pelatihan
   Dalam proses ini, dua set data telah disiapkan: satu untuk pengujian dan satu untuk pelatihan. Data pengujian digunakan selama eksperimen untuk mengukur kinerja *Non-Cryptographic Hash Function* (NCHFs). Data pelatihan akan digunakan untuk mengoptimalkan NCHFs atau melakukan pengujian dengan parameter khusus untuk mengetahui kondisi-kondisi tertentu yang mempengaruhi proses sinkronisasi [16].
4. Anonimitas dan Privasi
   Dalam tahap pengumpulan data, menjaga privasi dan kerahasiaan data merupakan prioritas. Prosedur penyamaran telah diterapkan untuk melindungi informasi sensitif yang mungkin ada dalam data. Data dari PT XYZ akan diubah sedemikian rupa sehingga informasi pribadi atau rahasia tidak dapat diidentifikasi kembali. Selain itu, langkah-langkah privasi dan keamanan data diikuti sesuai dengan peraturan dan aturan yang berlaku, serta berdasarkan persetujuan dari PT XYZ. Hal ini termasuk izin penggunaan data, penyimpanan data yang aman, dan penghapusan data yang sudah tidak dibutuhkan setelah eksperimen selesai. Dengan demikian, integritas dan privasi data telah terjaga sepanjang penelitian.

C. *Implementasi Algoritma*

Untuk menerapkan *Non-Cryptographic Hash Functions* (NCHFs) dalam skenario sinkronisasi data antara DBMS Oracle dan PostgreSQL, telah dikembangkan skrip khusus menggunakan bahasa pemrograman Python. Skrip ini dirancang untuk mengelola koneksi dari kedua DBMS, mengintegrasikan NCHFs dalam logika sinkronisasi data, menangani perbedaan data, dan mengoptimalkan kinerja sinkronisasi.

TABEL I
PSEUDOCODE PROSES SINKRONISASI DATA

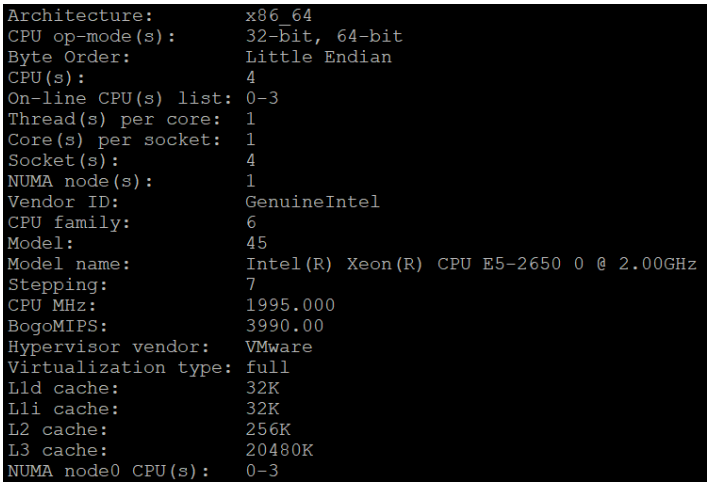
```
untuk setiap entri A di TabelA:
    jika entri A tidak ada di TabelB:
        tambahkan entri A ke TabelB
   lainnya jika entri A ada di TabelB:
        jika nilai hash dari entri A di TabelB berbeda dengan nilai hash dari entri A di
TabelA:
            update entri A di TabelB dengan nilai dari TabelA

untuk setiap entri B di TabelB:
    jika entri B tidak ada di TabelA:
        hapus entri B dari TabelB
```

Selain itu, akan disertakan langkah-langkah dokumentasi yang jelas untuk memudahkan pemahaman dan pemeliharaan di masa depan. Dokumentasi tersebut akan memastikan bahwa implementasi sesuai dengan aturan dan fitur khusus dari masing-masing DBMS. Setelah selesai, skrip ini akan diuji coba dan divalidasi untuk memastikan sinkronisasi data yang konsisten dan efisien dalam skenario eksperimen.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 1842-1854

D. Pengukuran Kinerja



Gambar. 4. Spesifikasi Perangkat Keras

Proses pengujian kinerja dilakukan pada server milik PT. XYZ yang sering digunakan untuk pengujian aplikasi dan database dummy. Server ini merupakan server dummy dengan sistem operasi Linux Ubuntu 18.04, CPU Intel Xeon E5-2650, dan RAM sebesar 8GB. Untuk melakukan pengujian, digunakan tiga aplikasi sebagai berikut:

- FortiClient VPN
  Merupakan aplikasi yang digunakan untuk membuat koneksi aman dan terenkripsi ke jaringan server PT. XYZ. Dalam konteks pengujian kinerja, koneksi ke jaringan server ini sangat penting karena memungkinkan para peneliti atau penguji untuk mengakses sumber daya dan lingkungan server yang diperlukan untuk menjalankan percobaan. Dengan menggunakan FortiClient VPN, pengguna dapat terhubung ke jaringan server dengan aman melalui koneksi VPN, sehingga data yang ditransmisikan antara pengguna dan server terlindungi dari ancaman keamanan eksternal [17].
- WinSCP
  Merupakan aplikasi yang memfasilitasi proses upload dan pengolahan skrip Python ke dalam server. Dalam konteks pengujian kinerja, skrip Python digunakan untuk melakukan berbagai tugas, seperti mengelola koneksi database, mengimplementasikan algoritma, dan menganalisis hasil pengujian. Dengan menggunakan WinSCP, pengguna dapat dengan mudah mengunggah skrip Python yang diperlukan ke dalam server, serta mengelola, memodifikasi, dan menghapus skrip tersebut sesuai kebutuhan pengujian.
- PuTTY
  Merupakan aplikasi yang digunakan untuk mengakses server secara remote melalui protokol SSH (Secure Shell). Dalam konteks pengujian kinerja, PuTTY memungkinkan pengguna untuk terhubung ke jaringan server dan menjalankan perintah langsung dari terminal remote. Hal ini memungkinkan pengguna untuk mengeksekusi skrip Python, mengunduh atau memperbarui *library* yang diperlukan, dan melakukan tindakan administratif lainnya pada server tanpa perlu berada di lokasi fisik server. Dengan menggunakan PuTTY, pengguna dapat mengelola server secara efisien dan melakukan tindakan yang diperlukan untuk menjalankan percobaan pengujian.

Dengan menggunakan aplikasi-aplikasi tersebut, proses pengujian kinerja dapat dilakukan secara efisien, sehingga semua aspek dari skrip dan pengaturan telah diuji dengan baik di lingkungan server yang sesuai. Proses pengujian kinerja dilakukan dengan memperhatikan parameter sebagai berikut:

1. Waktu Eksekusi
   Metrik ini bertujuan untuk mengukur seberapa efisien waktu yang dibutuhkan dalam proses sinkronisasi data antara DBMS Oracle dan PostgreSQL. Waktu eksekusi dipisahkan menjadi tiga tahapan utama: pertama, waktu untuk melakukan query select data; kedua, waktu untuk membandingkan data; dan ketiga, waktu untuk memproses operasi insert, update, atau delete data. Penghitungan waktu eksekusi dilakukan dengan merekam waktu awal dan akhir saat program dijalankan, kemudian mengurangkan keduanya untuk mendapatkan total waktu yang dibutuhkan. Fokus utama pengujian adalah pada tahapan membandingkan data, yang menjadi pusat evaluasi kinerja. Semua waktu eksekusi diukur dengan presisi dalam satuan milidetik.
2. Penggunaan Sumber Daya

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 1842-1854

Ini adalah metrik yang mengukur penggunaan sumber daya selama proses sinkronisasi. Sumber daya yang perlu diperhatikan meliputi:

- Penggunaan CPU

  Untuk mengukur sejauh mana CPU yang digunakan selama proses sinkronisasi, maka dimanfaatkanlah *library* Python yaitu *psutil*. Dengan memanggil fungsi "psutil.cpu_percent", informasi penggunaan CPU dalam bentuk persentase penggunaan dapat dihasilkan [18]. Fungsi ini akan dipanggil selama proses hashing dan komparasi data untuk mengetahui seberapa besar sumber daya CPU yang digunakan. Selain itu, penggunaan CPU dalam unit pengukuran yang lebih spesifik, seperti *megahertz*, juga dapat diukur dengan menggunakan fungsi-fungsi lain yang disediakan oleh *psutil*.
- Penggunaan Memori

TABEL II
PROSES PERHITUNGGAN MEMORY DENGAN LIBRARY TRACEMALLOC

```
import tracemalloc

def calculate_hash_CityHash(data_dict):
    tracemalloc.start()
    data_str = "|".join(
        f"{key}:{value}" if value is not None else f"{key}:NULL"
        for key, value in data_dict.items()
    )
    result = str(cityhash.CityHash64(data_str.encode()))
    current, peak = tracemalloc.get_traced_memory()
    print(f"Current memory usage: {current / 10**6} MB")
    print(f"Peak memory usage: {peak / 10**6} MB")
    tracemalloc.stop()
    return result
```

Untuk mengukur seberapa banyak memori yang digunakan selama proses sinkronisasi, digunakanlah dua library Python, yaitu "memory_profiler" dan "tracemalloc" [19], [20]. Library "memory_profiler" digunakan untuk mengukur secara keseluruhan seberapa besar memori yang digunakan oleh program sinkronisasi. Sementara itu, "tracemalloc" digunakan untuk memberikan detail yang lebih rinci tentang penggunaan memori untuk setiap *Non-CryptogrNaphic Hash Functions* (NCHFs). Dengan memanfaatkan kedua *library* ini, informasi mengenai jumlah memori yang digunakan selama proses sinkronisasi dan detail penggunaan memori untuk masing-masing NCHFs dapat diperoleh.

3. Keberhasilan dan Akurasi Sinkronisasi

Metrik ini mengukur persentase keberhasilan dan akurasi sinkronisasi data antara DBMS Oracle dan PostgreSQL. Keberhasilan dan akurasi sinkronisasi dapat dihitung sebagai jumlah data yang berhasil disinkronkan secara tepat dibagi dengan jumlah total data yang perlu disinkronkan. Rumusnya adalah sebagai berikut:

$$Akurasi\ data = \left(\frac{Jumlah\ kolom\ yang\ berhasil}{Total\ jumlah\ kolom} \times \frac{Baris\ data\ yang\ berhasil}{Total\ baris\ data}\right) \times 100\ \% \quad (1)$$

$$Akurasi\ sinkronisasi = \frac{Jumlah\ data\ yang\ berhasil\ disinkronisasi}{Total\ jumlah\ data\ yang\ perlu\ disinkronisasi} \times 100\ \% \quad (2)$$

Metrik ini memberikan pemahaman tentang seberapa baik *Non-Cryptographic Hash Functions* (NCHFs) menjaga konsistensi data antara kedua DBMS. Data yang dihasilkan selama eksperimen akan mencakup hasil dari metrik-metrik di atas. Informasi ini penting untuk analisis mendalam terhadap pengaruh penggunaan NCHFs terhadap kinerja sinkronisasi data.

E. Analisis Data

Data yang diperoleh dari eksperimen akan dianalisis untuk mengevaluasi sejauh mana *Non-Cryptographic Hash Functions* (NCHFs) mempengaruhi kinerja sinkronisasi data antara DBMS Oracle dan PostgreSQL. Dalam analisis ini, akan dievaluasi efisiensi algoritma, yang memberikan gambaran tentang seberapa baik algoritma sinkronisasi data berkinerja dengan dan tanpa penggunaan NCHFs. Efisiensi akan dihitung menggunakan rumus berikut:

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 1842-1854

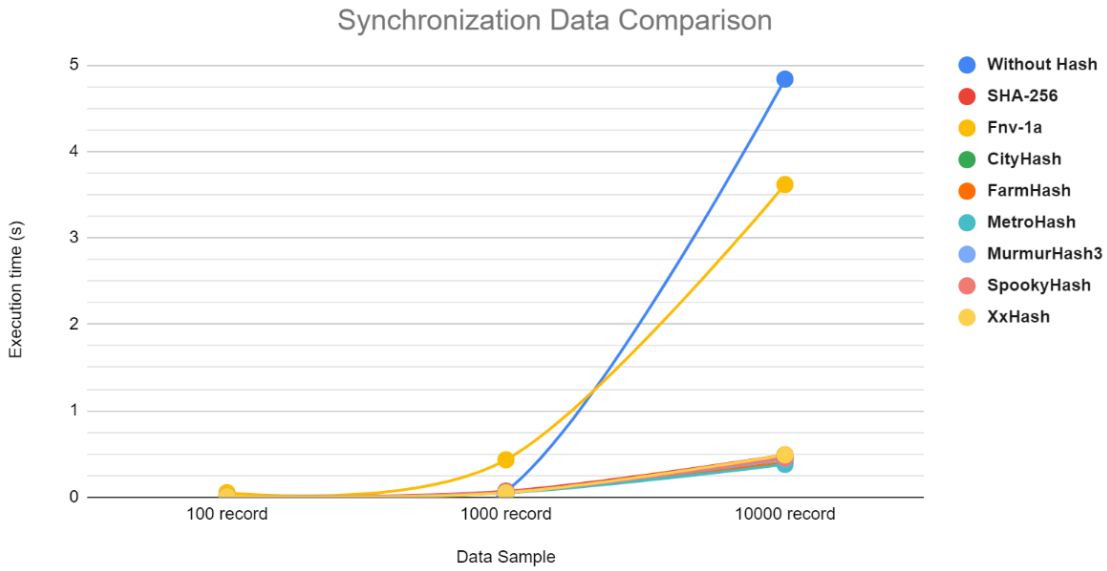
$$Efisiensi = \frac{Waktu\ hasing\ terendah}{Waktu\ hasing\ algoritma\ tertentu} \times \frac{Jumlah\ CPU\ terendah}{Jumlah\ CPU\ algoritma\ tertentu} \times \frac{Jumlah\ memori\ terendah}{Jumlah\ memori\ algoritma\ tertentu} \quad (1)$$

Rumus tersebut memberikan perbandingan antara waktu eksekusi, penggunaan CPU, dan penggunaan memori dari algoritma sinkronisasi data. Semakin tinggi nilai efisiensi, semakin baik kinerja algoritma sinkronisasi data dengan penggunaan NCHFs. Dengan demikian, analisis data akan melibatkan evaluasi kinerja algoritma sinkronisasi data, termasuk penggunaan NCHFs, untuk memperoleh pemahaman yang lebih baik tentang efektivitas teknik tersebut dalam meningkatkan sinkronisasi data.

III. HASIL DAN PEMBAHASAN

Berdasarkan eksperimen yang dilakukan, hasil penelitian menunjukkan bahwa penerapan *Non-Cryptographic Hash Functions* (NCHFs) dalam proses sinkronisasi data antara DBMS Oracle dan PostgreSQL memiliki dampak signifikan terhadap kinerja proses tersebut. Beberapa hasil kunci dari penelitian ini antara lain:

1. Waktu Eksekusi



Gambar. 5. Rata-rata Waktu Komparasi Data Selama Proses Sinkronisasi

Penerapan NCHFs berhasil mengoptimalkan waktu eksekusi proses sinkronisasi data secara signifikan. Dibandingkan dengan metode sinkronisasi tanpa menggunakan NCHFs, terjadi peningkatan kecepatan komparasi data sebesar 38% untuk 1000 data dan 1179% untuk 10.000 data. Meskipun terjadi peningkatan performa pada data ribuan, namun proses komparasi untuk data di bawah 1000 tetap lebih cepat apabila tanpa menggunakan metode hashing.

TABEL III
RATA-RATA WAKTU KOMPARASI DATA

Hash Function	100 record	1000 record	1000 record
Without Hash Function	**0.00089126**	0.0681022	4.8364864
SHA-256	0.00671998	0.067361	0.4884754
FNV-1a	0.05042088	0.4316832	3.6164358
CityHash	0.00498202	0.053261	0.4453678
FarmHash	0.00542568	0.05408392	0.403098
MetroHash	0.0054946	**0.0492986**	**0.3780884**
MurmurHash3	0.00610428	0.0559334	0.440352
SpookyHash	0.00561438	0.0551636	0.455098
xxHash	0.00607822	0.050273	0.4884754

Execution Time (seconds)

Penggunaan algoritma hashing tertentu, seperti *MetroHash*, menunjukkan kinerja yang paling optimal dalam mengurangi waktu eksekusi. Dibandingkan dengan algoritma hashing kriptografis seperti SHA-256, *Non-Cryptographic Hash Functions* (NCHFs) seperti *MetroHash* lebih cepat sebesar 29% dalam proses hashing dan komparasi data. Hal ini menunjukkan bahwa *MetroHash* memberikan peningkatan kecepatan yang signifikan, menambah nilai efisiensi dalam sinkronisasi data antara DBMS Oracle dan PostgreSQL. Keunggulan kinerja ini menjadi penting dalam konteks aplikasi bisnis di mana efisiensi waktu eksekusi menjadi faktor utama dalam mempertahankan produktivitas dan efektivitas operasional.

TABEL IV
RATA-RATA WAKTU PROSES HASHING

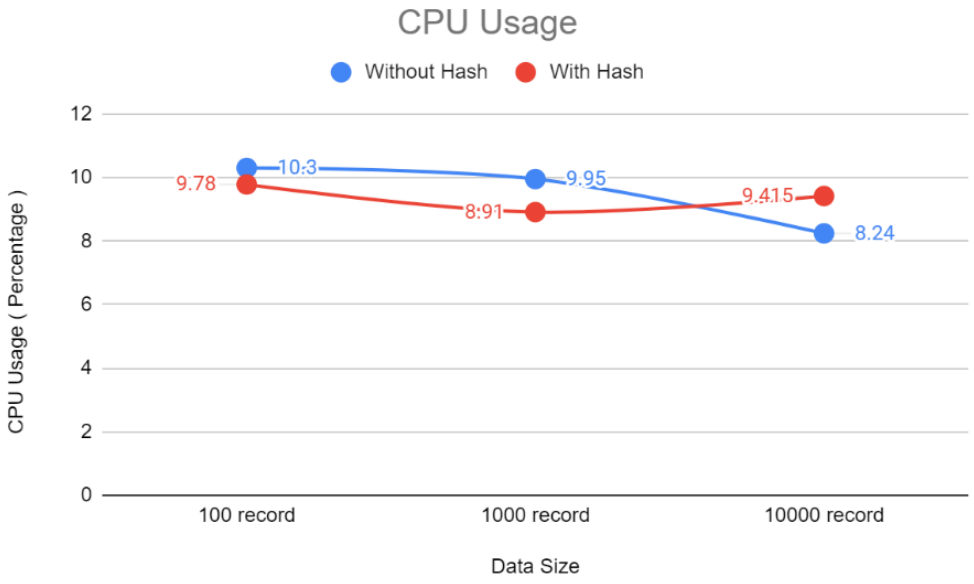
Hash Function	Tabel A	Tabel B	Tabel C	Tabel D	Tabel E
SHA-256	3.382719	1.4748164	0.459786	1.414419	4.5300922
FNV-1a	26.231139	10.8172234	3.279058	5.6095214	20.1884374
SpookyHash	2.5952418	1.237428	0.3363008	1.1314212	3.6620412
CityHash	2.3894318	1.1746744	0.3351736	1.1365534	3.3353982
MurmurHash3	2.7707056	1.2968266	0.3669346	1.2280494	3.7994672
MetroHash	**2.2300476**	**1.1240158**	**0.2933838**	**1.0615214**	**3.2589372**
FarmHash	2.3171614	1.1659894	0.3258984	1.1144266	3.2773572
xxHash	2.4472662	1.2557208	0.357741	1.160527	3.412066

Execution Time (seconds)

Dalam pengujian hashing terhadap 5 tabel yang dimiliki oleh PT.XYZ, hasil menunjukkan bahwa *Metro-Hash* mencapai waktu hashing tercepat di semua tabel tersebut, menandakan keunggulan kinerjanya. Selain itu, dari 7 *Non-Cryptographic Hash Functions* (NCHFs) yang diuji, 6 di antaranya menunjukkan performa yang lebih unggul dibandingkan dengan algoritma hashing kriptografis standar seperti SHA-256. Namun, yang perlu diperhatikan adalah bahwa *FNV-1a* menunjukkan kinerja hashing yang paling rendah di antara semua fungsi hashing yang dievaluasi.

Berdasarkan hasil pengujian, terdapat peningkatan waktu eksekusi yang signifikan dibandingkan dengan penelitian sebelumnya yang dilakukan oleh Ayu Helinda dan rekan-rekannya tentang sinkronisasi data antar DBMS dengan algoritma MD5 [14]. Sebagai contoh, dalam penelitian mereka, diperlukan waktu sekitar 12 detik untuk memproses 10.000 data, sementara penelitian ini menunjukkan bahwa algoritma NCHFs *MetroHash* hanya membutuhkan waktu sekitar 0,37 detik. Peningkatan ini dipengaruhi oleh perbe-daan jenis data, algoritma sinkronisasi, dan fungsi hashing yang digunakan.

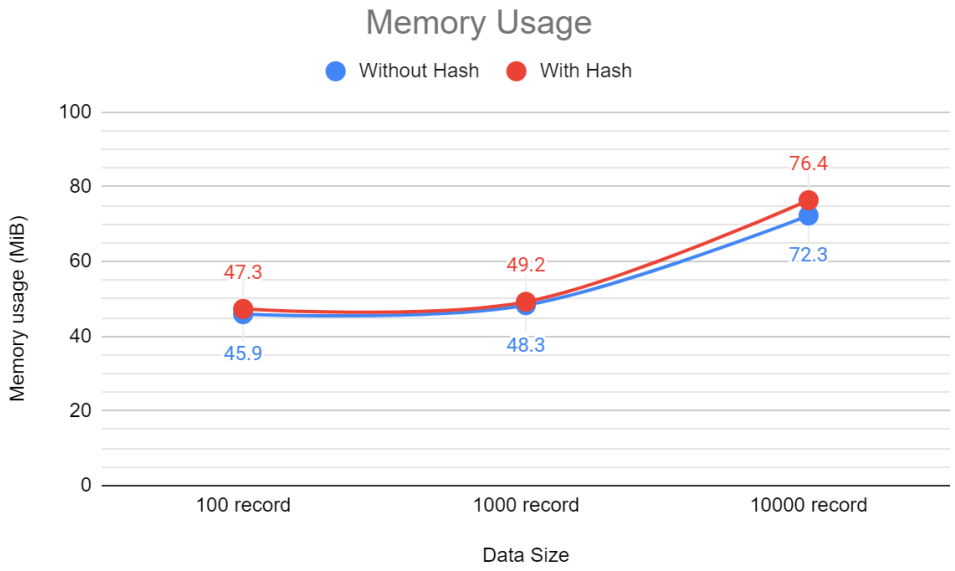
2. Penggunaan Sumber Daya



Gambar. 6. Rata-rata Penggunaan CPU Selama Proses Sinkronisasi

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 4, Desember 2024, Pp. 1842-1854

Dalam penggunaan *Non-Cryptographic Hash Functions* (NCHFs) selama proses sinkronisasi, terdapat perubahan dalam penggunaan sumber daya. Meskipun terjadi peningkatan efisiensi dengan pengurangan penggunaan CPU sebesar 1.36%, namun terdapat kompensasi dengan peningkatan penggunaan memori sebesar 3.52%. Hal ini terjadi dikarenakan dalam penggunaan NCHFs diperlukan memori tambahan untuk menyimpan nilai hash. Oleh karena itu, perlu diperhatikan bahwa meskipun penggunaan CPU menurun, namun peningkatan penggunaan memori menunjukkan adanya aspek tambahan yang perlu dipertimbangkan dalam penerapan NCHFs.



Gambar. 7. Rata-rata Penggunaan Memori Selama Proses Sinkronisasi

Penerapan NCHFs tidak hanya meningkatkan kecepatan eksekusi, tetapi juga mengoptimalkan penggunaan sumber daya CPU. Namun, hal ini berbanding lurus dengan peningkatan penggunaan memori yang dibutuhkan untuk menyimpan nilai hash dari masing-masing data.

- Penggunaan CPU

TABEL V
RATA-RATA PENGGUNAAN CPU SELAMA PROSES HASHING

Hash Function	Core 1	Core 2	Core 3	Core 4	Average
SHA-256	10.04	10.82	8.02	7.14	9.005
FNV-1a	10.9	5.6	13.82	7.7	9.505
SpookyHash	7.2	14.04	9.82	2.6	8.415
CityHash	6.78	9.32	12.22	7.82	9.035
MurmurHash3	16.1	10.68	3.6	8.6	9.745
MetroHash	8.36	2.82	4.4	19.5	8.77
FarmHash	6.98	14.2	10.2	10.56	10.485
xxHash	10.04	10.82	8.02	7.14	9.005

CPU Usage (%)

Dalam pengujian hashing terhadap salah satu tabel milik PT.XYZ, *xxHash* menunjukkan penggunaan CPU tertinggi, sementara *FNV-1a* menunjukkan penggunaan CPU terendah. *FNV-1a* memerlukan waktu hashing yang lebih lama, sehingga wajar jika penggunaan CPU-nya lebih rendah.

- Penggunaan Memori

TABEL VI
RATA-RATA PENGGUNAAN MEMORI DALAM SATU KALI HASHING DATA

Hash Function	Memory	Peak Memory
SHA-256	0.003384	0.00466
FNV-1a	0.001873	0.002458
CityHash	0.000953	0.002274
SpookyHash	0.000934	0.002274
MurmurHash	0.000978	0.002274
MetroHash	0.000954	0.002274
FarmHash	0.000954	0.002274
xxHash	0.00095	0.002274

Memory Usage (MiB)

Analisis memori dilakukan pada setiap NCHFs dengan memperhatikan jumlah memori yang digunakan saat proses hashing satu data. Hasilnya menunjukkan bahwa secara umum, NCHFs menggunakan memori lebih sedikit dibandingkan dengan algoritma kriptografis seperti SHA-256. *xxHash* menunjukkan keunggulan sebagai yang paling efisien dalam penggunaan memori. Meskipun demikian, *FNV-1a* menunjukkan kasus yang berbeda dimana memori yang digunakan jauh lebih tinggi dibandingkan dengan NCHFs lainnya.

3. Keberhasilan dan Akurasi Sinkronisasi

$$Akurasi\ data\ = \left(\frac{15}{15} \times \frac{176919}{176919}\right) \times 100\% = \ 100\%$$

$$Akurasi\ sinkronisasi = \frac{176919}{176919} \times 100\% = \ 100\%$$

Gambar. 8. Perhitungan Akurasi Tabel Item PT.XYZ

Selama proses pengujian, tidak terjadi masalah atau kegagalan dalam sinkronisasi data, baik sebelum maupun setelah penerapan *Non-Cryptographic Hash Functions* (NCHFs). Proses sinkronisasi berjalan dengan lancar tanpa hambatan. Ketika diperiksa lebih lanjut, hasil menunjukkan bahwa akurasi sinkronisasi data cukup tinggi, hal ini menunjukkan bahwa NCHFs telah berhasil memelihara integritas data antara kedua sistem basis data.

Semua data berhasil disinkronkan secara tepat, sehingga nilai akurasi untuk semua data adalah 100%. Meskipun tidak ada masalah yang ditemukan selama pengujian, penting untuk mencatat bahwa penggunaan NCHFs memerlukan pertimbangan lebih lanjut untuk memahami potensi risikonya, termasuk kemungkinan terjadinya hash yang sama untuk nilai yang berbeda. Perlu dilakukan penelitian lebih lanjut untuk menganalisis risiko ini dengan lebih mendalam.

4. Analisa Data

TABEL VII
NILAI EFISIENSI NCHFS

Hash Function	Efficiency Score
FNV1a	0.07865123857
CityHash	0.8721474132
SpookyHash	0.8245003276
MurmurHash	0.6950179602
MetroHash	0.9313779745
FarmHash	0.8520397317
xxHash	0.731338786

Dari hasil perhitungan nilai efisiensi untuk masing-masing *Non-Cryptographic Hash Functions* (NCHFs) yang diperoleh berdasarkan rumus berikut:

$$Efisiensi = \frac{Waktu\ hasing\ terendah}{Waktu\ hasing\ algoritma\ tertentu} \times \frac{Jumlah\ CPU\ terendah}{Jumlah\ CPU\ algoritma\ tertentu} \times \frac{Jumlah\ memori\ terendah}{Jumlah\ memori\ algoritma\ tertentu} \quad (1)$$

Nilai efisiensi ini dihitung berdasarkan 3 parameter utama: waktu hashing, penggunaan CPU, dan penggunaan memori. Hasil pengujian terhadap ketiga parameter ini pada data milik PT.XYZ memberikan informasi bahwa *MetroHash* merupakan NCHFs dengan nilai efisiensi terbaik, sehingga layak untuk diterapkan pada skrip sinkronisasi data milik PT.XYZ. Meskipun *FNV-1a* menggunakan CPU paling sedikit, namun NCHFs ini memakan waktu hashing yang terlalu lama, menjadikannya NCHFs paling tidak efisien dalam pengujian ini.

Dari hasil eksperimen, terlihat bahwa penerapan NCHFs, terutama *MetroHash*, dalam sinkronisasi data antara DBMS Oracle dan PostgreSQL memberikan peningkatan kinerja yang signifikan. Namun, ada beberapa hal yang perlu diperhatikan berdasarkan hasil penelitian ini. Meskipun *MetroHash* berhasil mengoptimalkan waktu eksekusi, penting untuk diingat bahwa penggunaan memori yang lebih tinggi juga perlu dipertimbangkan. Selain itu, terdapat batasan dalam penelitian seperti pemilihan NCHFs yang terbatas, sehingga tidak mencakup semua NCHFs yang tersedia kemudian fokus pada DBMS spesifik, yaitu Oracle dan PostgreSQL serta, penelitian ini dilakukan dalam konteks bisnis PT XYZ, yang mungkin tidak mencerminkan semua skenario bisnis lainnya.

IV. KESIMPULAN

Penelitian ini berhasil mengevaluasi dan membandingkan kinerja tujuh *Non-Cryptographic Hash Functions* (NCHFs) dalam konteks sinkronisasi data antara DBMS Oracle dan PostgreSQL. Hasil penelitian menunjukkan bahwa penerapan NCHFs memberikan kontribusi positif dalam meningkatkan efisiensi sinkronisasi data. Proses yang sebelumnya memakan waktu berkisar 1 hingga 5 jam, kini telah dioptimalkan menjadi kurang dari 30 menit. Dalam konteks PT XYZ, optimalisasi sinkronisasi data dengan memilih algoritma hashing yang sesuai dapat menghasilkan proses yang lebih cepat dan efisien. Meskipun demikian, tidak semua NCHFs memberikan kinerja yang sama baiknya, dan beberapa di antaranya bahkan lebih lambat daripada algoritma hashing umum. Hasil penelitian menunjukkan bahwa *MetroHash*, *CityHash*, dan *FarmHash* terbukti sebagai pilihan yang baik dalam kasus ini, dengan *MetroHash* menunjukkan kinerja tertinggi dalam pengurangan waktu eksekusi. Kesimpulannya, pemilihan dan implementasi NCHFs yang tepat dapat memberikan manfaat nyata dalam meningkatkan kinerja sinkronisasi data antara DBMS Oracle dan PostgreSQL. Saran untuk PT XYZ adalah mempertimbangkan penggunaan algoritma hashing seperti *MetroHash* atau *CityHash* untuk mengoptimalkan proses sinkronisasi data mereka. Selain itu, penelitian lebih lanjut diperlukan untuk mengevaluasi dan menghindari algoritma hashing yang memiliki kinerja buruk seperti *FNV-1a* agar dapat menghasilkan proses sinkronisasi data yang lebih efisien dan cepat.

DAFTAR PUSTAKA

[1] K. Nakatani, T.-T. Chuang, and D. Zhou, "Data Synchronization Technology: Standards, Business Values and Implications," *Commun. Assoc. Inf. Syst.*, vol. 17, p. 44, 2006, [Online]. Available: https://api.semanticscholar.org/CorpusID:9411565

[2] M. B. Kekgathetse, M. B. Kekgathetse, and K. J. Letsholo, "A survey on database synchronization algorithms for mobile device," *Article in Journal of Theoretical and Applied Information Technology*, vol. 10, no. 1, 2016, [Online]. Available: https://www.researchgate.net/publication/300187546

[3] S. Ram and V. Ramesh, "Management of Heterogeneous and Autonomous Database Systems," 1999. [Online]. Available: https://api.semanticscholar.org/CorpusID:53774227

[4] E. P. Csirmaz and L. Csirmaz, "Data Synchronization: A Complete Theoretical Solution for Filesystems," *Future Internet*, vol. 14, no. 11, Nov. 2022, doi: 10.3390/fi14110344.

[5] D. Emanuel and A. Arévalo, "Hashing: Types, Benefits and Security Issues," Feb. 2024, [Online]. Available: https://ssrn.com/abstract=4718938

[6] A. Halevy, A. Rajaraman, and J. Ordille, *Data Integration: The Teenage Years*. 2006.

[7] P. P. Pittalia, "International Journal of Computer Science and Mobile Computing A Comparative Study of Hash Algorithms in Cryptography," 2019. [Online]. Available: www.ijcsmc.com

[8] Stinson, Douglas Robert, and Maura Paterson, "Cryptography Theory and Practice Fourth Edition," 2018. Accessed: Nov. 02, 2023. [Online]. Available: https://www.perlego.com/book/2193350/cryptography-theory-and-practice-pdf

[9] R. Patgiri, S. Nayak, and N. Muppalaneni, *Bloom Filter: A Data Structure for Computer Networking, Big Data, Cloud Computing, Internet of Things, Bioinformatics and Beyond*. 2021.

[10] A. Mittelbach and M. Fischlin, "Non-cryptographic Hashing," in *The Theory of Hash Functions and Random Oracles: An Approach to Modern Cryptography*, A. Mittelbach and M. Fischlin, Eds., Cham: Springer International Publishing, 2021, pp. 303–334. doi: 10.1007/978-3-030-63287-8_7.

[11] V. Banja, M. Ilić, L. Kopanja, D. Zlatković, M. Trajković, and D. Ćurguz, "The 7th International conference Knowledge management and informatics MICROSOFT SQL SERVER AND ORACLE: COMPARATIVE PERFORMANCE ANALYSIS," 2021.

[12] S. Andjelic, S. Obradovic, and B. Gacesa, "A performance analysis of the DBMS - MySQL vs PostgreSQL," *Communications - Scientific Letters of the University of Žilina*, vol. 10, no. 4. University of Žilina, pp. 53–57, 2008. doi: 10.26552/com.c.2008.4.53-57.

[13] A. Sateesan, J. Biesmans, T. Claesen, J. Vliegen, and N. Mentens, "Optimized algorithms and architectures for fast non-cryptographic hash functions in hardware," *Microprocess Microsyst*, vol. 98, Apr. 2023, doi: 10.1016/j.micpro.2023.104782.

[14] A. Helinda, Z. Musliyana, D. R. Y. Tb, M. Dwipayana, J. Suanda, and A. N. Johari, "Performance analysis of heterogeneous database management system (DBMS) synchronization using message digest 5," in *AIP Conference Proceedings*, American Institute of Physics Inc., Nov. 2020. doi: 10.1063/5.0027970.

[15] P. F. Tanaem, A. F. Wijaya, A. D. Manuputty, and G. N. Huwae, "Penerapan RESTFul Web Service Pada Disain Arsitektur Sistem Informasi Pada Perguruan Tinggi (Studi Kasus: STARS UKSW)," *JASIEK (Jurnal Aplikasi Sains, Informasi, Elektronika dan Komputer)*, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:225730519

[16] T. Wahyuningsih, A. Iriani, H. Purnomo, and I. Sembiring, "Predicting students' success level in an examination using advanced linear regression and extreme gradient boosting," *Computer Science and Information Technologies*, vol. 5, pp. 23–31, Apr. 2024, doi: 10.11591/csit.v5i1.p23-31.

[17] T. U. A, S. Jayaram, and S. G. Hegde, "A Brief Study of Computer Network Security Technologies." 2024.

[18] H. S. Lella, K. Manasa, R. Chattaraj, and S. Chimalakonda, "DBJoules: An Energy Measurement Tool for Database Management Systems," *ArXiv*, vol. abs/2311.08961, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:265212964

[19] E. Berger, S. Stern, and J. A. Pizzorno, "Triangulating Python Performance Issues with Scalene," *ArXiv*, vol. abs/2212.07597, 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:254685810

[20] M. Katsaragakis, L. Papadopoulos, M. Konijnenburg, F. Catthoor, and D. Soudris, "A memory footprint optimization framework for Python applications targeting edge devices," *Journal of Systems Architecture*, vol. 142, p. 102936, 2023, doi: https://doi.org/10.1016/j.sysarc.2023.102936.