



Implementasi Foreign Data Wrapper Pada Aplikasi Nexchief Di PT Paramadaksa Teknologi Nusantara

Marlina Sentosa¹, Sebastian Nigel², Wasis Haryono^{3*}

^{1,2,3}Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Pamulang, Kota Tangerang Selatan, Indonesia

Email : ¹marlinasentosa88@gmail.com, ²sebastiannigel.utility@gmail.com, ^{3*}wasish@unpam.ac.id

(* : coresponding author)

Abstrak - Penelitian ini membahas implementasi teknologi Foreign Data Wrapper (FDW) pada aplikasi Nexchief di PT Paramadaksa Teknologi Nusantara untuk mengoptimalkan pengelolaan data lintas basis data. FDW memungkinkan integrasi data dari berbagai sumber secara efisien tanpa perlu replikasi data, sehingga mengurangi kompleksitas manajemen basis data. Hasil penelitian menunjukkan bahwa penerapan FDW memberikan kontribusi signifikan terhadap optimalisasi pengelolaan data, mempercepat proses pengambilan keputusan, dan meningkatkan kinerja aplikasi secara keseluruhan. Dengan FDW, proses integrasi data menjadi lebih fleksibel dan terstandardisasi, menjadikannya solusi yang efektif untuk kebutuhan pengelolaan data lintas platform.

Kata Kunci : Foreign Data Wrapper, Integrasi Data, Pengelolaan Data, Kinerja Aplikasi

Abstract - This study discusses the implementation of Foreign Data Wrapper (FDW) technology in the Nexchief application at PT Paramadaksa Teknologi Nusantara to optimize cross-database data management. FDW allows efficient data integration from various sources without the need for data replication, thereby reducing the complexity of database management. The results of the study show that the implementation of FDW makes a significant contribution to optimizing data management, accelerating the decision-making process, and improving overall application performance. With FDW, the data integration process becomes more flexible and standardized, making it an effective solution for cross-platform data management needs.

Keywords: Foreign Data Wrapper, Data Integration, Data Management, Application Performance.

1. PENDAHULUAN

Dalam era digital yang semakin maju, pengelolaan data menjadi salah satu aspek penting yang harus diperhatikan oleh perusahaan untuk mendukung operasional bisnis yang efisien dan efektif. Salah satu tantangan yang sering dihadapi adalah bagaimana mengintegrasikan data dari berbagai sumber yang berbeda ke dalam satu platform tanpa mengorbankan performa maupun keamanan data itu sendiri. PT Paramadaksa Teknologi Nusantara, sebagai perusahaan yang bergerak di bidang teknologi, berupaya untuk mengatasi tantangan ini.

Oleh karena itu, Untuk mengatasi permasalahan ini, perusahaan PT Paramadaksa Teknologi Nusantara memutuskan untuk mengimplementasikan teknologi Foreign Data Wrapper (FDW) pada aplikasi Nexchief, platform unggulan mereka yang berfungsi untuk memfasilitasi manajemen data dan informasi secara terpusat.

Nexchief adalah sebuah aplikasi yang dirancang untuk mengelola alur kerja (workflow), manajemen dokumen, dan proses bisnis dalam berbagai sektor industri. Namun, untuk mendukung kebutuhan yang semakin dinamis, Nexchief harus dapat mengakses data dari berbagai sumber eksternal, seperti sistem ERP, database cloud, atau sistem warisan (legacy systems), yang semuanya menggunakan teknologi dan struktur basis data yang berbeda. Proses integrasi lintas platform sering kali memerlukan upaya yang rumit, membutuhkan waktu panjang, dan dapat mempengaruhi kinerja aplikasi secara keseluruhan jika tidak dikelola dengan baik.

Foreign Data Wrapper (FDW) adalah teknologi yang memungkinkan sebuah sistem manajemen basis data untuk mengakses dan memanipulasi data yang ada di luar basis data itu sendiri seolah-olah data tersebut merupakan bagian dari sistem lokal. Dengan implementasi FDW pada aplikasi Nexchief, PT Paramadaksa Teknologi Nusantara dapat mengintegrasikan berbagai sumber data eksternal secara langsung ke dalam lingkungan aplikasi mereka tanpa harus memindahkan atau menduplikasi data tersebut, yang pada akhirnya meningkatkan efisiensi dan fleksibilitas dalam pengelolaan data.



Dalam proses implementasinya, PT Paramadaksa Teknologi Nusantara menghadapi sejumlah tantangan, seperti penyesuaian kinerja aplikasi saat mengakses data eksternal yang besar, pengelolaan latensi jaringan, serta memastikan bahwa integrasi dengan sistem eksternal berjalan lancar tanpa gangguan terhadap operasi harian perusahaan. Meskipun begitu, penerapan FDW diharapkan dapat memberikan peningkatan signifikan pada fleksibilitas, skalabilitas, dan efisiensi operasional aplikasi Nexchief.

2. METODE

Penelitian ini menggunakan metode studi kasus yang berfokus pada implementasi Foreign Data Wrapper (FDW) pada aplikasi Nexchief di PT Paramadaksa Teknologi Nusantara. Proses penelitian dilakukan melalui beberapa tahapan yang dirancang secara sistematis untuk memastikan implementasi berjalan sesuai dengan tujuan yang telah ditetapkan. Berikut adalah tahapan metode penelitian yang digunakan:

a. Studi Literatur

Penelitian diawali dengan melakukan kajian pustaka untuk memahami konsep dasar Foreign Data Wrapper (FDW), PostgreSQL sebagai platform yang mendukung FDW, serta teknik integrasi data lintas platform. Literatur yang dikaji meliputi Artikel dan jurnal terkait integrasi data antar basis data, Studi kasus dan laporan implementasi FDW pada sistem yang serupa.

b. Observasi

Penulis melakukan observasi yaitu dengan melihat secara langsung cara kerja bagian-bagian yang terkait dengan pencatatan hasil-hasil kegiatan yang dilakukan, setelah itu penulis diberikan kesempatan untuk melihatnya.

c. Interview

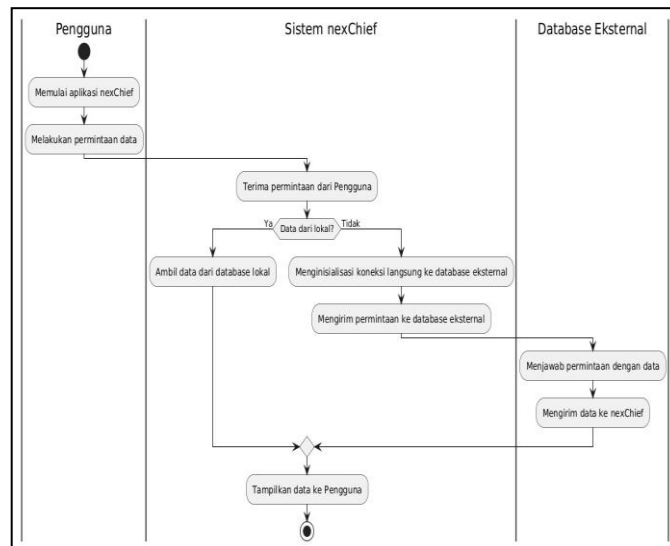
Wawancara dilakukan untuk mengetahui masalah yang timbul atau dialami langsung oleh setiap subjek yang bersangkutan. Dalam kegiatan ini diajukan pertanyaan lisan dalam usaha untuk melengkapi data-data yang akan diperoleh. Wawancara dilakukan kepada bagian-bagian yang terkait dalam sistem aplikasi Nexchief.

3. ANALISIS DAN PEMBAHASAN

3.1 Perancangan Sistem

Perancangan sistem adalah proses mendesain dan mengembangkan struktur serta fungsi dari sebuah sistem agar memenuhi kebutuhan pengguna dan tujuan organisasi. Dalam suatu perancangan proses terhadap sistem yang akan dirancang terdapat komponen-komponen yang dihasilkan dari analisis siklus pengembangan sistem yang dapat berupa perancangan, penggambaran suatu sistem yang akan dirancang.[4]

a. Activity Diagram sistem berjalan



Gambar 1. Activity Diagram Sistem Berjalan

Alur proses permintaan data di aplikasi NexChief:

1. Inisiasi oleh Pengguna:

Pengguna memulai aplikasi NexChief dan melakukan permintaan data. Ini adalah langkah pertama yang memulai proses permintaan data dalam sistem.

2. Penerimaan Permintaan oleh Sistem NexChief:

Sistem NexChief menerima permintaan data dari pengguna. Setelah menerima permintaan, sistem memeriksa apakah data tersebut tersedia di database lokal atau perlu diambil dari database eksternal.

3. Pengambilan Data dari Database Lokal:

Jika data yang diminta tersedia di database lokal, sistem mengambil data langsung dari sana. Setelah data diperoleh, sistem melanjutkan ke langkah akhir untuk menampilkan data kepada pengguna.

4. Inisialisasi Koneksi ke Database Eksternal:

Jika data yang diminta berada di database eksternal, sistem melakukan inisialisasi koneksi langsung ke database eksternal. Ini memungkinkan sistem untuk berkomunikasi dan mengirimkan permintaan data ke database eksternal.

5. Mengirim Permintaan ke Database Eksternal:

Setelah koneksi diinisialisasi, sistem NexChief mengirim permintaan data ke database eksternal. Permintaan ini akan diteruskan untuk diproses oleh database eksternal.

6. Proses di Database Eksternal:

Database eksternal menerima dan memproses permintaan data dari NexChief. Setelah data diperoleh, database eksternal mengirimkan respons yang berisi data yang diminta kembali ke sistem NexChief.

7. Pengembalian Data ke Sistem NexChief:

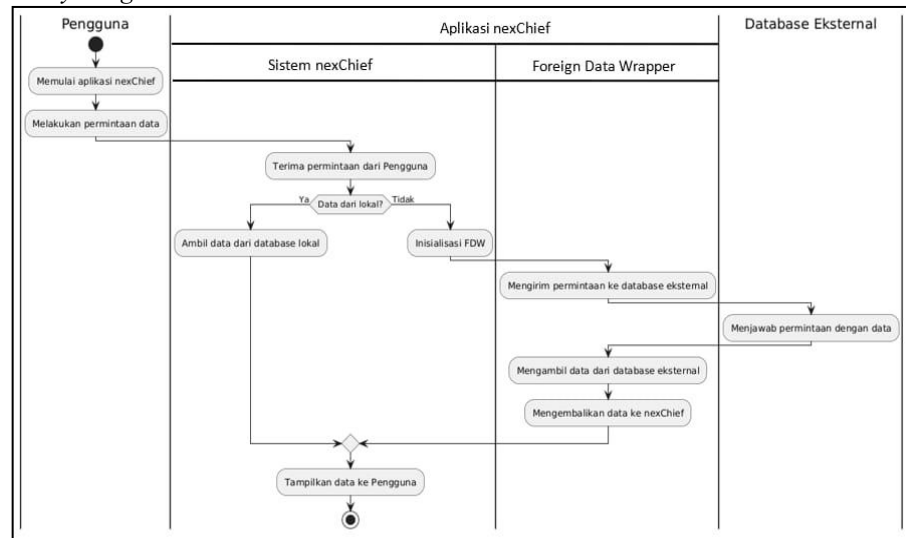
Setelah data diterima dari database eksternal, sistem NexChief menggabungkannya dengan data lainnya (jika ada) dan melanjutkan ke langkah akhir.

8. Menampilkan Data kepada Pengguna:

Sistem NexChief menampilkan data yang diperoleh kepada pengguna. Ini adalah langkah terakhir di mana pengguna dapat melihat hasil dari permintaan data yang telah diproses.

Diagram aktivitas ini menunjukkan alur kerja dari permintaan data yang berasal dari pengguna hingga data tersebut ditampilkan. Proses ini mencakup dua skenario: pengambilan data dari database lokal dan database eksternal. Alur ini menggambarkan fleksibilitas sistem NexChief dalam mengakses data dari berbagai sumber untuk memenuhi kebutuhan pengguna secara efektif.

b. *Activity Diagram* Sistem Usulan



Gambar 2. *Activity Diagram* Sistem Usulan

Alur proses permintaan data di aplikasi NexChief menggunakan Foreign Data Wrapper (FDW):

1. Inisiasi oleh Pengguna:

Diagram dimulai dengan Pengguna yang membuka aplikasi NexChief dan melakukan permintaan data. Ini merupakan langkah pertama yang menginisiasi seluruh proses permintaan data.

2. Penerimaan Permintaan di Sistem NexChief:

Aplikasi NexChief menerima permintaan data dari pengguna. Sistem kemudian memeriksa apakah data yang diminta berada di database lokal atau eksternal.

3. Pengambilan Data dari Database Lokal:

Jika data yang diminta berada di database lokal, sistem langsung mengambil data dari sana dan melanjutkan ke langkah terakhir untuk menampilkan data kepada pengguna.

4. Inisialisasi Foreign Data Wrapper (FDW) untuk Database Eksternal:

Jika data yang diminta berada di database eksternal, sistem melakukan inisialisasi FDW. FDW adalah mekanisme yang memungkinkan aplikasi NexChief untuk mengakses data dari database yang berbeda seolah-olah data tersebut ada di dalam database lokal.

5. Pengiriman Permintaan ke Database Eksternal:

Setelah FDW diinisialisasi, permintaan data dikirimkan ke database eksternal melalui FDW.

6. Proses di Database Eksternal:

Database eksternal menerima permintaan data tersebut dan memprosesnya. Setelah data ditemukan, database eksternal mengirimkan respons yang berisi data yang diminta kembali ke FDW.

7. Pengambilan dan Pengembalian Data oleh FDW:

FDW mengambil data dari database eksternal dan meneruskannya kembali ke sistem NexChief.

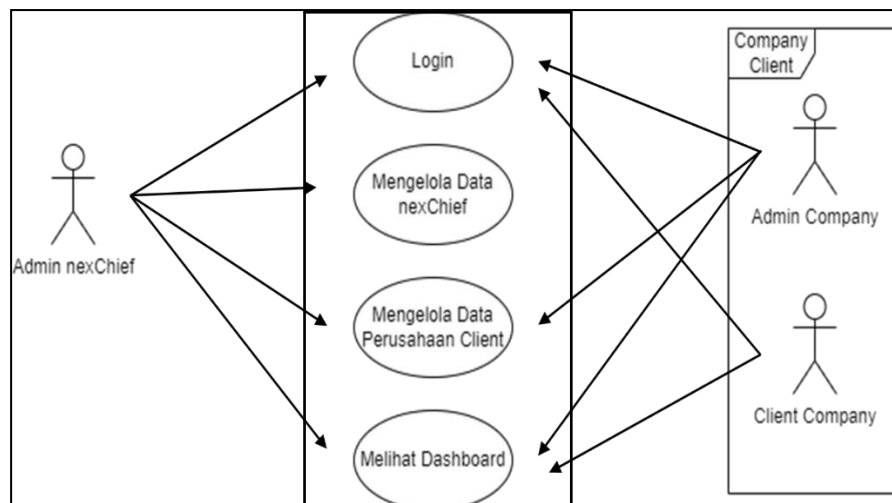
8. Penampilan Data kepada Pengguna:

Sistem NexChief menerima data dari FDW dan menampilkan hasilnya kepada pengguna. Ini merupakan langkah terakhir di mana pengguna akhirnya dapat melihat data yang diminta.

Diagram aktivitas ini menggambarkan proses yang efisien dalam pengelolaan permintaan data yang bersumber dari berbagai database, baik lokal maupun eksternal, dengan memanfaatkan FDW. Dengan FDW, aplikasi NexChief dapat mengakses data dari database yang berbeda tanpa harus melakukan replikasi data, sehingga memungkinkan integrasi data lintas platform secara fleksibel dan cepat.

c. *Use Case Diagram*

Use Case Diagram adalah satu jenis dari diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor.[5]



Gambar 3. Use Case Diagram

❖ **Aktor**

➤ **Admin Nexchief:**

- **Akses penuh:** login, melihat dashboard, mengelola data Nexchief, dan mengelola data perusahaan client.

➤ **Admin Client:**

- **Akses terbatas:** login, melihat dashboard, dan mengelola data perusahaan client.

➤ **End Client:**

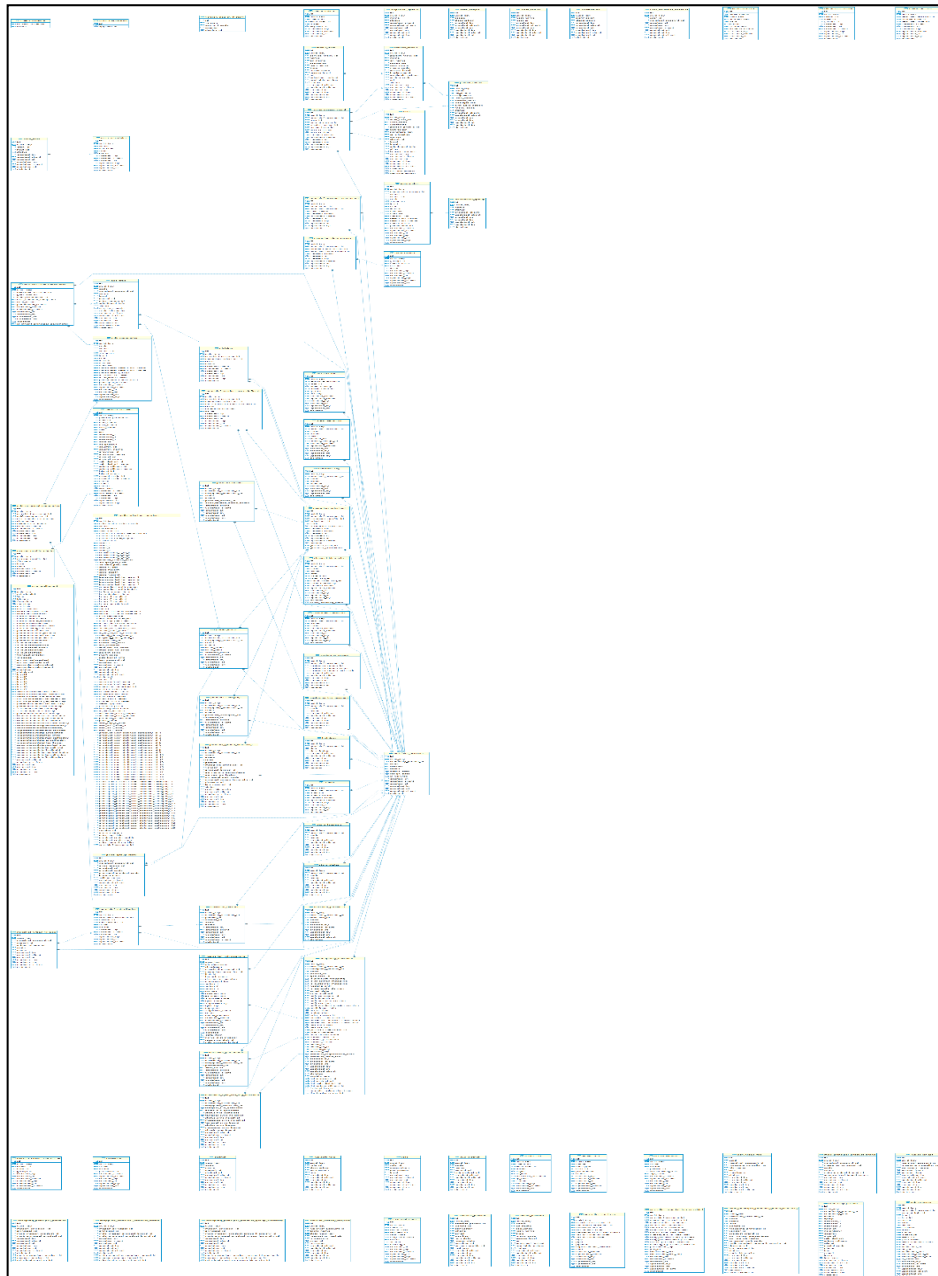
- **Akses minimal:** login dan melihat dashboard.

❖ **Use Cases**

- **LOGIN:** Semua aktor perlu login untuk mengakses sistem.
- **MELIHAT DASHBOARD:** Tersedia untuk semua aktor.

- **MENGELOLA DATA NEXCHIEF:** Hanya untuk Admin Nexchief.
- **MENGELOLA DATA PERUSAHAAN CLIENT:** Untuk Admin Nexchief dan Admin Client.

d. ERD



Gambar 4. Entity Relationship Diagram

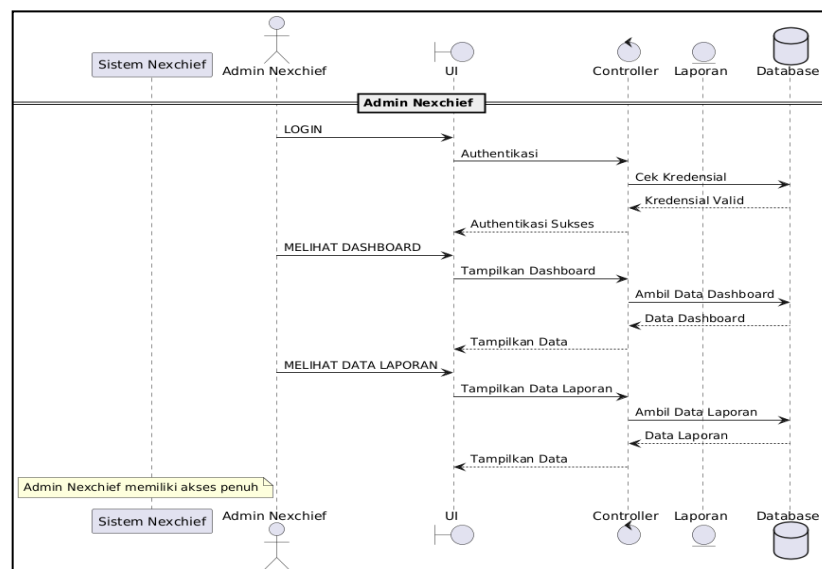
Diagram Entity-Relationship (ERD) atau model relasional database, yang menggambarkan hubungan antara tabel-tabel dalam sebuah sistem. Berikut adalah penjelasannya:

- Tabel-tabel di dalam database:
 - a) Setiap kotak mewakili tabel dalam database.
 - b) Nama tabel biasanya terletak di bagian atas kotak.

- c) Kolom di dalam tabel terdaftar di bawah nama tabel, dengan kolom kunci utama (primary key) sering kali ditandai secara khusus (misalnya, dengan garis bawah atau huruf tebal).
- Relasi antar tabel:
 - a) Garis yang menghubungkan tabel-tabel menunjukkan relasi antar tabel.
 - b) Garis ini biasanya dilengkapi dengan simbol di kedua ujungnya untuk menunjukkan jenis relasi, seperti one-to-one, one-to-many, atau many-to-many.
- Struktur relasi:
 - a) Dalam diagram ini, terlihat struktur database yang cukup kompleks dengan banyak tabel saling terkait.
 - b) Beberapa tabel terlihat menjadi pusat (hub table), menandakan bahwa tabel-tabel tersebut memiliki banyak hubungan dengan tabel lainnya.
- e. *Sequence Diagram*

Sequence diagram ialah menunjukkan proses jalannya sebuah sistem untuk melihat operasi apa saja yang dapat dilakukan oleh sebuah sistem tersebut.[6] Berikut adalah gambar sequence diagram yang di buat:

➤ **Sequence Diagram untuk Admin Nexchief**



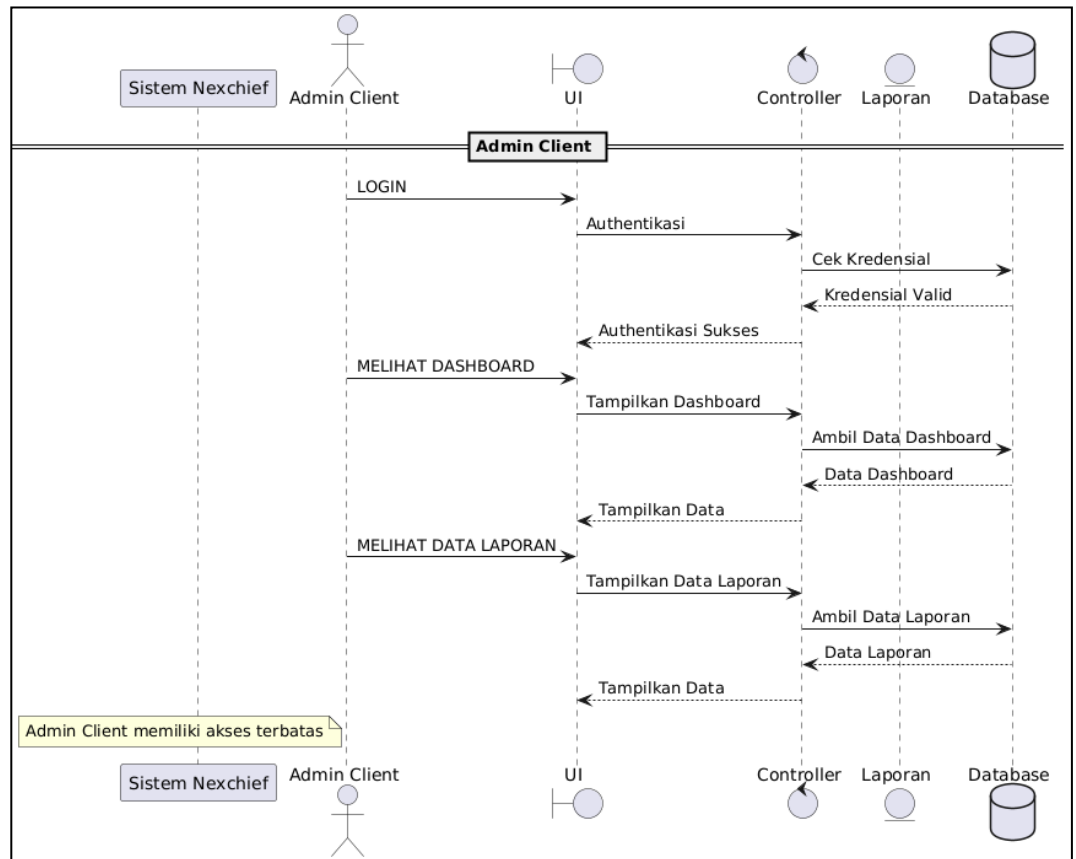
Gambar 5. Sequence Diagram Untuk Admin Nexchief

Penjelasan:

- **LOGIN:** Admin Nexchief memulai interaksi dengan sistem melalui antarmuka pengguna (UI) untuk melakukan login. UI kemudian meminta autentikasi ke Controller, yang memeriksa kredensial di Database. Jika kredensial valid, sistem mengonfirmasi autentikasi yang berhasil kepada UI.
- **MELIHAT DASHBOARD:** Setelah login, Admin Nexchief meminta untuk melihat dashboard. UI mengarahkan permintaan ke Controller, yang mengambil data dari Database dan mengirimkan data tersebut kembali ke UI untuk ditampilkan.
- **MELIHAT DATA LAPORAN:** Admin Nexchief meminta untuk melihat data laporan. Prosesnya mirip, di mana Controller mengambil data laporan dari Database dan mengembalikannya ke UI untuk ditampilkan.

- **Akses Penuh:** Admin Nexchief memiliki akses penuh untuk semua fitur dalam sistem.

➤ **Sequence Diagram untuk Admin Client**

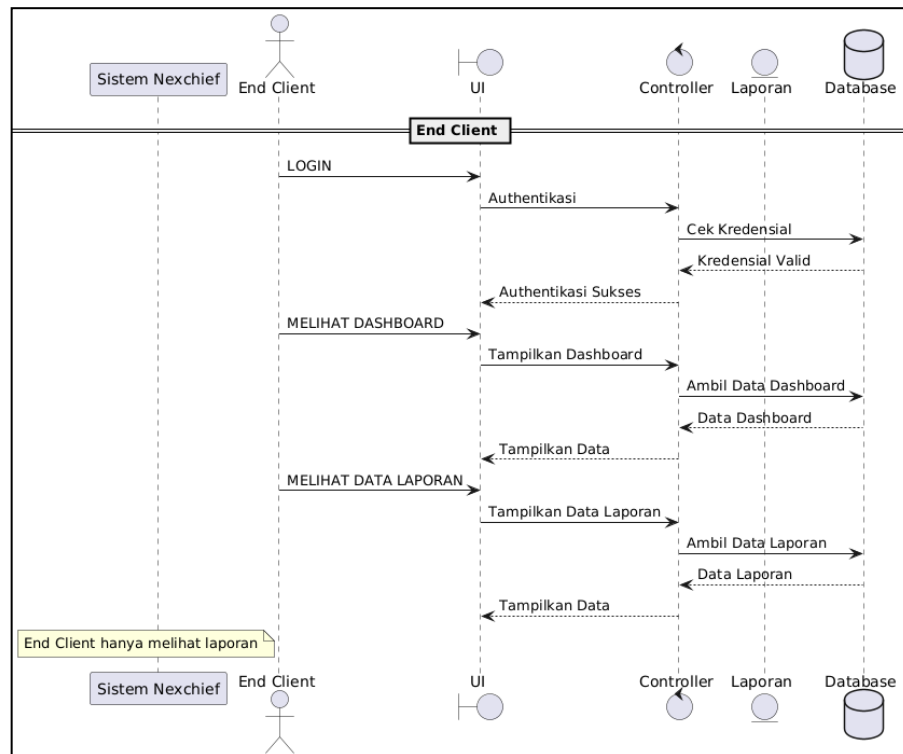


Gambar 3. 1 Sequence Diagram untuk Admin Client

Penjelasan:

- **LOGIN:** Proses login dilakukan oleh Admin Client sama seperti Admin Nexchief, di mana UI meminta autentikasi ke Controller dan memverifikasi kredensial di Database.
- **MELIHAT DASHBOARD:** Setelah berhasil login, Admin Client meminta untuk melihat dashboard, dan prosesnya sama seperti sebelumnya—Controller mengambil data dari Database dan mengembalikannya ke UI.
- **MELIHAT DATA LAPORAN:** Admin Client juga dapat melihat data laporan dengan cara yang sama. Namun, Admin Client tidak memiliki akses untuk mengedit atau menghapus data.
- **Akses Terbatas:** Admin Client memiliki akses terbatas, sehingga tidak dapat melakukan tindakan lain seperti membuat atau mengupdate data laporan.

➤ *Sequence Diagram untuk End Client*

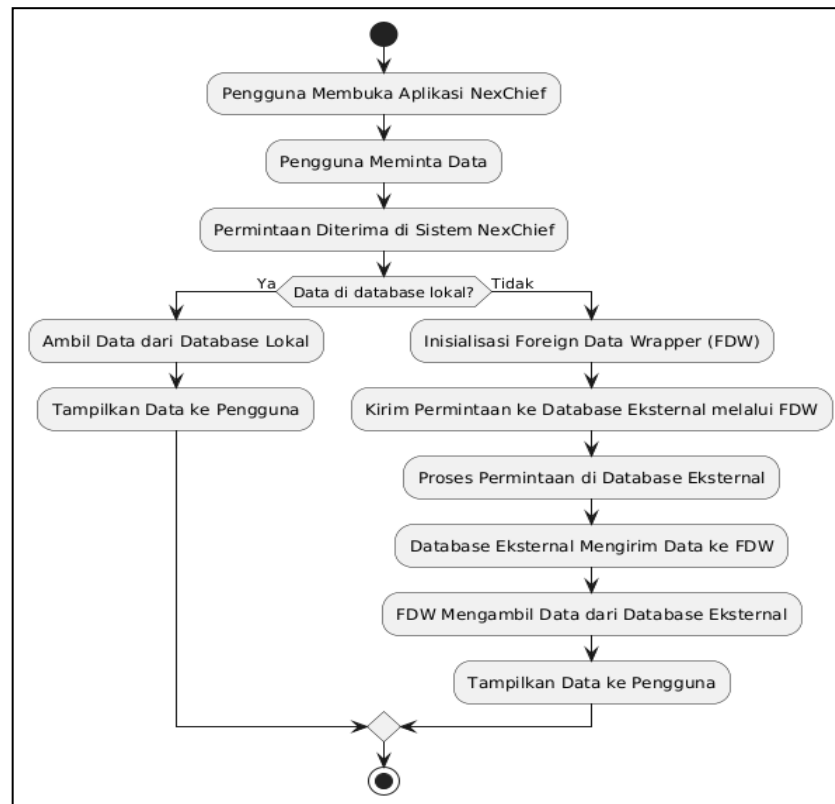


Gambar 7. *Sequence Diagram untuk End Client*

Penjelasan:

- **LOGIN:** End Client juga melakukan login dengan proses yang sama, di mana UI memverifikasi kredensial melalui Controller dan Database.
- **MELIHAT DASHBOARD:** Setelah login, End Client dapat melihat dashboard, dengan data diambil dari Database dan ditampilkan oleh UI.
- **MELIHAT DATA LAPORAN:** End Client dapat melihat laporan, tetapi tidak memiliki kemampuan untuk membuat, mengupdate, atau menghapus laporan.
- **Akses Terbatas:** End Client hanya memiliki akses untuk melihat informasi, tanpa kemampuan untuk mengubah data.

f. Perancangan Perangkat Lunak : Flowchart



Gambar 8. Flowchart

Penjelasan Flowchart

1. Inisiasi:

- Pengguna membuka aplikasi NexChief dan melakukan permintaan data.

2. Penerimaan Permintaan:

- Sistem NexChief menerima permintaan data.

3. Pemeriksaan Data:

- Sistem memeriksa apakah data yang diminta ada di database lokal.
- Jika Ya, data diambil dari database lokal dan ditampilkan kepada pengguna.
- Jika Tidak, sistem melanjutkan ke langkah berikutnya.

4. Inisialisasi FDW:

- Sistem menginisialisasi Foreign Data Wrapper (FDW) untuk mengakses database eksternal.

5. Kirim Permintaan ke Database Eksternal:

- Permintaan data dikirim ke database eksternal melalui FDW.

6. Proses di Database Eksternal:

- Database eksternal memproses permintaan dan mengirimkan respons yang berisi data kembali ke FDW.

7. Ambil dan Tampilkan Data:

- FDW mengambil data dari database eksternal dan mengembalikannya ke sistem NexChief, yang kemudian menampilkan data kepada pengguna.

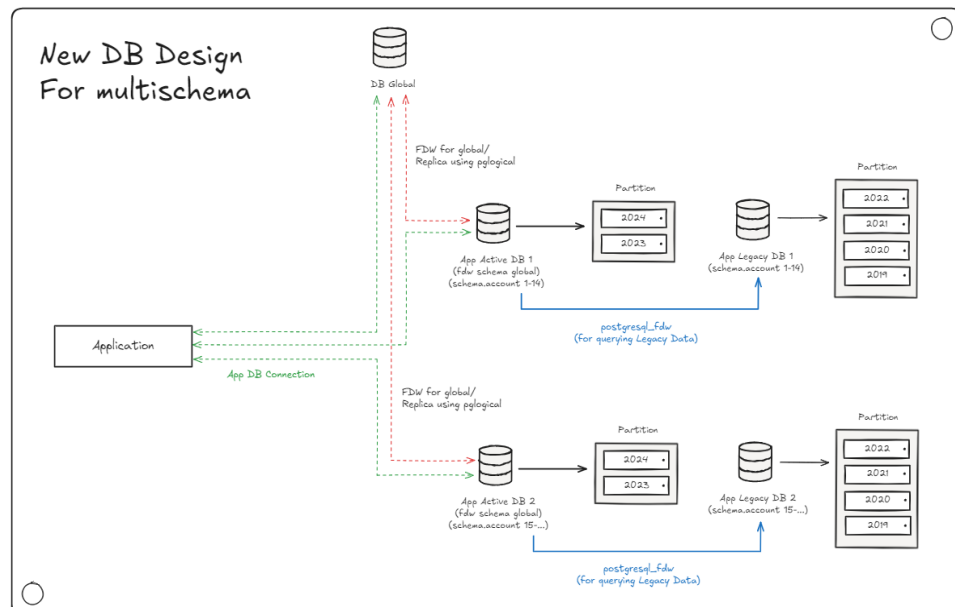
3.2 Analisa dan Pembahasan

3.2.1 Pembahasan Foreign Data Wrapper

Foreign Data Wrapper (FDW) adalah sebuah mekanisme dalam PostgreSQL yang memungkinkan integrasi data dari sumber eksternal (seperti database lain, file, atau API) ke dalam PostgreSQL. Dengan FDW, pengguna dapat melakukan query pada data eksternal seolah-olah data tersebut adalah tabel lokal dalam database.

FDW (Foreign Data Wrapper) di PostgreSQL memungkinkan instance PostgreSQL untuk mengakses tabel yang disimpan di database jarak jauh (baik itu PostgreSQL atau sistem database lainnya). Secara prinsip, FDW menciptakan cara bagi database lokal untuk berinteraksi dengan tabel dari database eksternal seolah-olah tabel tersebut berada di dalam database lokal, memungkinkan kueri dan integrasi lintas database.

3.2.2 Rancangan Database



Gambar 9. Database Multi-Skema Usulan

Rancangan dalam gambar menunjukkan desain database baru dengan pendekatan multischema yang mengintegrasikan beberapa komponen database menggunakan Foreign Data Wrapper (FDW) di PostgreSQL.

3.2.3 Implementasi dan Penjelasan Rancangan Database

1) Cara Kerja FDW di PostgreSQL:

- Definisi Server Jarak Jauh (Foreign Server Definition):** Menyusun koneksi ke database jarak jauh. Ini mencakup informasi tentang lokasi database eksternal dan cara mengaksesnya.
- Pemetaan Pengguna (User Mapping):** Memetakan pengguna lokal ke pengguna database jarak jauh. Ini memastikan bahwa ketika pengguna lokal mengakses data di database eksternal, hak akses yang tepat diterapkan.



- c) **Tabel Eksternal (Foreign Table):** Tabel yang mendefinisikan struktur tabel dari database eksternal. Ini memungkinkan PostgreSQL untuk mengakses dan berinteraksi dengan data dari tabel yang ada di database lain seolah-olah tabel tersebut ada di dalam database lokal.

Berikut adalah penjelasan komponen utama dalam rancangan tersebut:

(1) Komponen Utama

Aplikasi

- Fungsi: Aplikasi ini adalah titik akses utama yang terhubung ke database untuk melakukan query dan manipulasi data.
- Aplikasi menggunakan koneksi database untuk mengakses App Active DB (Database Aktif) dan data lainnya melalui FDW.

DB Global

- Fungsi: Sebagai database utama yang menyimpan data global dan dapat diakses oleh semua database aktif menggunakan FDW.
- FDW: Database ini diakses oleh App Active DB 1 dan App Active DB 2 untuk mendapatkan data global.

App Active DB 1 dan DB 2

- Fungsi:
 - a) Setiap database aktif dikelola secara terpisah untuk berbagai keperluan operasional.
 - b) Dibedakan berdasarkan skema, seperti:
 - App Active DB 1: Skema untuk akun 1-14.
 - App Active DB 2: Skema untuk akun 15+.
- Partisi:
 - a) Data pada database aktif dipartisi berdasarkan tahun (misalnya: 2024, 2023).
 - b) Hal ini dirancang untuk efisiensi penyimpanan dan performa query.

App Legacy DB 1 dan DB 2

- Fungsi:
 - a) Menyimpan data historis (legacy) dari akun yang sama.
 - b) Dikelola secara terpisah berdasarkan skema (akun 1-14 di Legacy DB 1, akun 15+ di Legacy DB 2).
 - c) Data juga dipartisi berdasarkan tahun (misalnya: 2022, 2021).

Alur Data dan FDW

FDW untuk DB Global

- Deskripsi: FDW digunakan untuk mengakses data dari DB Global ke database aktif (App Active DB 1 dan App Active DB 2).
- Implementasi: Menggunakan FDW seperti `postgres_fdw` atau replikasi dengan `pglogical` untuk sinkronisasi data global.
- Fungsi:
 - Data global dapat digunakan secara efisien oleh database aktif tanpa perlu replikasi fisik di masing-masing database.

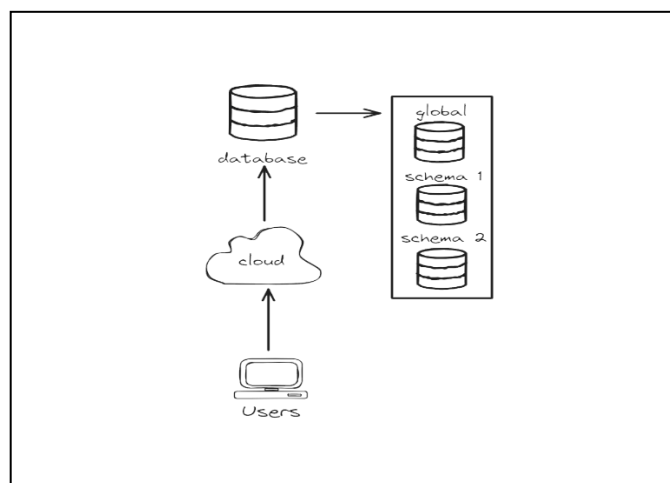
FDW untuk Data Legacy

- Deskripsi: FDW juga digunakan oleh App Active DB untuk mengakses data historis di App Legacy DB.
- Implementasi:
 - Data legacy tetap berada di database historis tetapi dapat di-query langsung dari database aktif.
 - Memungkinkan analisis data lintas waktu tanpa perlu duplikasi data.

Keunggulan Rancangan

1. Modularitas:
 - Setiap database aktif dan legacy dikelola terpisah berdasarkan skema, mempermudah pengelolaan data untuk kelompok akun tertentu.
 - Data global dipusatkan di satu tempat dan dapat diakses oleh semua database menggunakan FDW.
2. Efisiensi Performa:
 - Partisi data berdasarkan tahun meningkatkan performa query pada database aktif dan legacy.
 - Data yang sering digunakan (data aktif) berada di database terpisah dari data historis (legacy).
3. Fleksibilitas Query:
 - Dengan FDW, aplikasi dapat melakukan query lintas database dan skema tanpa perlu memindahkan data secara fisik.
4. Skalabilitas:
 - Jika diperlukan, lebih banyak database aktif atau legacy dapat ditambahkan tanpa mengganggu sistem yang ada.
5. Replikasi Global:
 - Menggunakan replikasi seperti pglogical, data global selalu tersedia dan sinkron untuk semua database.

3.2.4 Penggunaan Database Yang Sedang Berjalan



Gambar 10. Database multi-Skema Yang Sudah Ada

Pada gambar tersebut mengidentifikasi beberapa kelebihan (pros) dari pendekatan penyimpanan data dengan skema terpisah dalam satu basis data, yaitu:

➤ **Pemisahan Data Secara Logis**

Setiap perusahaan (atau *tenant*) memiliki skema sendiri, sehingga data setiap perusahaan tetap terisolasi meskipun berada di dalam basis data yang sama. Ini membantu menjaga privasi dan keamanan data.

➤ **Manajemen yang Lebih Sederhana**

Dengan semua skema berada dalam satu basis data, proses pencadangan (*backup*), pemeliharaan, dan pemantauan dapat dilakukan secara terpusat. Tidak perlu mengelola banyak basis data untuk setiap perusahaan, yang menyederhanakan infrastruktur.

➤ **Data Global Terpusat**

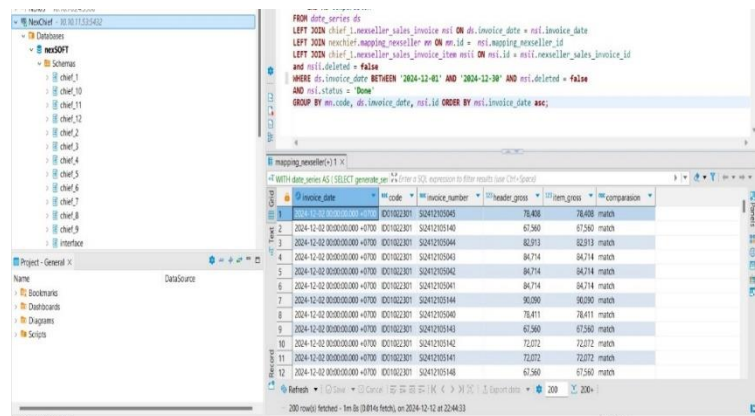
Kehadiran skema global memungkinkan penyimpanan data yang dapat digunakan bersama oleh beberapa perusahaan, seperti konfigurasi umum atau data referensi. Hal ini menghindari duplikasi data global pada skema perusahaan yang berbeda.

➤ **Penerapan yang Lebih Mudah**

Satu basis data membuat arsitektur penerapan lebih sederhana, tanpa logika kompleks untuk mengelola koneksi ke banyak basis data atau pengaturan server yang rumit.

3.2.5 Uji Coba Program Dengan Contoh Data

- Berikut hasil eksekusinya dalam sebuah tool database (DBeaver) sebelum implementasi FDW, Sharding, dan Partitioning:



Waktu yang ditempuh oleh query untuk menampilkan hasil adalah 1 menit 8 detik (dapat dilihat pada bagian bawah layar "1m 8s").

Durasi ini menunjukkan waktu eksekusi query untuk:

- Membuat data tanggal menggunakan generate_series.
- Melakukan beberapa LEFT JOIN ke tabel yang relevan.
- Memproses filter (WHERE), pengelompokan (GROUP BY), dan pengurutan (ORDER BY).
- Menampilkan 200 baris hasil.

Query SQL ini mengambil data faktur dari beberapa tabel dengan langkah berikut:

- WITH Clause: Membuat daftar tanggal (generate_series) sebagai referensi.
- LEFT JOIN: Menggabungkan tabel faktur (nexseller_sales_invoice), pemetaan (mapping_nexseller), dan detail faktur (sales_invoice_item).

- c) WHERE: Menyaring data berdasarkan tanggal faktur (1–30 Desember 2024), status "Done", dan tidak dihapus.
- d) GROUP BY & ORDER BY: Mengelompokkan data berdasarkan kode, tanggal faktur, dan ID faktur, lalu mengurutkannya berdasarkan tanggal faktur.

Hasil: Menampilkan invoice_date, code, invoice_number, header_gross, item_gross, dan comparison. Semua data comparison menunjukkan "match", artinya nilai header_gross dan item_gross sama.

- Berikut hasil eksekusinya dalam sebuah tool database (DBeaver) setelah implementasi FDW, Sharding, dan Partitioning:

	invoice_date	code	invoice_number	header_gross	item_gross	comparison
1	2024-12-02 00:00:00.000+0700	IC0005586	524120866	46,757	46,757	match
2	2024-12-02 00:00:00.000+0700	IC0005586	524120867	28,378	28,378	match
3	2024-12-02 00:00:00.000+0700	IC0005586	524120868	26,126	26,126	match
4	2024-12-02 00:00:00.000+0700	IC0005586	524120869	17,117	17,117	match
5	2024-12-02 00:00:00.000+0700	IC0005586	524120872	113,313	113,313	match
6	2024-12-02 00:00:00.000+0700	IC0005586	524120883	163,963	163,963	match
7	2024-12-02 00:00:00.000+0700	IC0005586	524120884	157,657	157,657	match
8	2024-12-02 00:00:00.000+0700	IC0005586	524120885	157,657	157,657	match
9	2024-12-02 00:00:00.000+0700	IC0005586	524120880	91,114	91,114	match
10	2024-12-02 00:00:00.000+0700	IC0005586	524120881	91,114	91,114	match
11	2024-12-02 00:00:00.000+0700	IC0005586	524120899	16,620	16,620	match
12	2024-12-02 00:00:00.000+0700	IC0005586	524120897	28,378	28,378	match
13	2024-12-02 00:00:00.000+0700	IC0005586	524120898	11,261	11,261	match

Waktu yang ditempuh untuk mengeksekusi query pada gambar ini adalah 50 detik, seperti terlihat di bagian bawah layar ("50s").

Proses yang mempengaruhi waktu eksekusi:

- a) Generate Series: Membuat rentang tanggal menggunakan fungsi generate_series untuk referensi data.
- b) LEFT JOIN: Melakukan beberapa gabungan (JOIN) tabel besar, seperti tabel faktur, pemetaan, dan item faktur.
- c) Filter Data: Menyaring data dengan kriteria tertentu (tanggal, status, dan deleted). Grouping dan Sorting: Mengelompokkan data (GROUP BY) dan mengurutkan berdasarkan tanggal faktur (ORDER BY).

Hasilnya: Query menghasilkan 400 baris data dengan waktu eksekusi 50 detik, yang mencakup waktu untuk mengambil data, memproses filter, dan menampilkan hasil.

4. KESIMPULAN

Setelah menyelesaikan kerja praktik di PT Paramadaksa Teknologi Nusantara, khususnya dalam mengimplementasikan *Foreign Data Wrapper* (FDW) pada aplikasi Nexchief, dapat disimpulkan bahwa teknologi FDW memberikan kontribusi signifikan terhadap optimalisasi pengelolaan data lintas basis data. FDW memungkinkan integrasi data dari berbagai sumber dengan lebih efisien, sehingga mempercepat proses pengambilan keputusan dan meningkatkan kinerja aplikasi.

Selama kerja praktik, penulis mendapatkan pengalaman berharga dalam memahami cara kerja FDW, konfigurasi, dan implementasinya dalam lingkungan nyata. Selain itu, penulis juga belajar untuk bekerja secara kolaboratif dalam tim dan menghadapi tantangan yang muncul selama proses implementasi.

Adapun kendala yang dihadapi selama pengerjaan, seperti kompatibilitas antar versi basis data atau penyesuaian query lintas sumber, telah berhasil diatasi melalui diskusi dengan pembimbing.



dan pengujian sistem yang berkelanjutan. Dengan selesainya tugas ini, penulis berharap hasil kerja ini dapat memberikan manfaat bagi perusahaan dalam meningkatkan efisiensi sistem aplikasi Nexchief.

Penulis mengucapkan terima kasih kepada PT Paramadaksa Teknologi Nusantara atas kesempatan yang diberikan untuk belajar dan berkembang. Semoga pengalaman ini menjadi awal yang baik untuk karier profesional penulis di masa depan.

REFERENCES

- [1] S. Samidi, R. Y. Suladi, and A. B. Lesmana, "Implementation of Database Distributed Sharding Horizontal Partition in MySQL. Case Study of Application of Food Serving On Kemkes," *J. Sisfotek Glob.*, vol. 12, no. 1, p. 50, 2022, doi: 10.38101/sisfotek.v12i1.477.
- [2] A. Qushay, A. L. Barra, and E. F. Dharmawan, "' PostgreSQL ' Database," pp. 1–18, 2017.
- [3] J. M. Hellerstein and M. Stonebraker, *Readings in database systems*, 4th ed. Cambridge, Mass. SE - xi, 865 pages : illustrations ; 28 cm: MIT Press, 2005. doi: LK - <https://worldcat.org/title/57546479>.
- [4] A. Selva Noviriliya, M. Andini, S. Bela Hutabarat, and W. Haryono, "Analisa Dan Pengembangan Aplikasi Buku Tamu Berbasis Web Pada Kelurahan Pondok Cabe Udik," *J. Ilmu Komput. dan Sci.*, vol. 1, no. 08, pp. 1088–1094, 2022, [Online]. Available: <https://journal.mediapublikasi.id/index.php/oktal>
- [5] R. Maulana, M. Farid, and W. Haryono, "OKTAL : Jurnal Ilmu Komputer dan Science PERANCANGAN SISTEM INFORMASI PADA PT. PRASTY MITRA SOLUSINDO BERBASIS WEB," vol. 2, no. 9, pp. 2564–2576, 2023, [Online]. Available: <https://journal.mediapublikasi.id/index.php/oktal>
- [6] M. Farhan Ramadhan, R. Cahya Gumilang, S. Zakki Galbi, and W. Haryono, "OKTAL : Jurnal Ilmu Komputer dan Science Perancangan Sistem Inventory Berbasis Dekstop (Studi Kasus : Toko Beras Pusaka Rama)," *OKTAL J. Ilmu Komput. dan Sci. Peranc. Sist. Invent. Berbas. Dekstop (Studi Kasus Toko Beras Pusaka Rama)*, vol. 1, no. 08, pp. 1187–1192, 2022, [Online]. Available: <https://journal.mediapublikasi.id/index.php/oktal>