

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Data warehouse atau gudang data adalah repositori data gabungan semua data yang dikumpulkan dari berbagai sistem operasional, baik fisik maupun logis. *Data warehouse* menekankan pada pengambilan data dari berbagai sumber data operasional untuk kemudian diakses dan dianalisis menjadi informasi [1]. Sementara *data mart* adalah subset dari *data warehouse* yang berorientasi pada lini bisnis tertentu. Pemanfaatan *data warehouse* dapat memberi manfaat bagi penggunaannya dari sudut pandang bisnis atau pun teknis, dengan memisahkan data atau pun proses operasional dengan proses analisis. Hal ini dapat meningkatkan kinerja sistem operasional dan memungkinkan pengguna bisnis untuk dapat mengakses data analisis dengan lebih relevan, konsisten, dan lebih cepat karena tidak bergantung pada sistem operasional [2].

Dalam mengimplementasikan *data warehouse* atau *data mart* terdapat proses ETL atau *Extract Transform Load*, merupakan prosedur umum menyalin data dari satu atau lebih sumber(*extract*) lalu mengubahnya ke dalam bentuk sesuai dengan yang diharapkan(*transform*) dan menyimpannya ke dalam *data warehouse* atau *data mart* (*load*). Umumnya, ETL merupakan proses *batch* dimana sistem akan melakukan ekstraksi terhadap keseluruhan data sumber dan mengubahnya [3], [4].

Proses ETL yang merupakan proses *batch* adalah proses yang melihat pada keseluruhan data sebagai proses ekstraksi. Proses ini memakan waktu yang lama dan membutuhkan sumber daya komputasi yang besar, sehingga ETL hanya dapat berjalan pada waktu-waktu tertentu agar tidak mengganggu sistem operasional yang berjalan. Hal ini menjadi permasalahan apabila sistem berjalan secara terus menerus tanpa henti dan data yang diproses diperlukan secara *real-time* terutama apabila data tersebut merupakan data yang *real-time* seperti data *time-series* [9, 10].

Berdasarkan permasalahan umum yang ada pada proses ETL tersebut, maka akan dilakukan “**Optimasi ETL Perangkat Lunak Data Mart**” dengan menggunakan metode *Change Data Capture*. *Change Data Capture* atau CDC akan melihat log pada basis data sumber dan melaporkan setiap perubahan data yang

terjadi pada waktu yang bersamaan dengan ukuran data yang kecil. Dengan demikian proses ekstraksi pada ETL yang pada umumnya merupakan *batch-processing* dapat dilakukan secara *real-time* berbentuk *streaming* data dari sumber datanya [6, 7]. Dengan penelitian ini, diharapkan dapat menghilangkan gangguan yang terjadi pada sistem operasional dan dapat memperkecil kebutuhan sumber daya komputasi dan waktu yang digunakan.

1.2 Identifikasi Masalah

Proses ETL *batch-processing* berjalan pada waktu-waktu tertentu sering mengganggu sistem operasional yang berjalan juga memakan sumber daya yang besar dan waktu yang lama sehingga menjadi permasalahan apabila data yang diproses diperlukan secara *real-time*.

1.3 Maksud dan Tujuan

1.3.1 Maksud

Melakukan optimasi pada proses ETL pada perangkat lunak *data mart* untuk menghemat waktu dan sumber daya yang digunakan dengan menggunakan teknik *Change Data Capture* dengan memanfaatkan teknologi *message broker*.

1.3.2 Tujuan

1. Membuat proses ETL *realtime* menggunakan metode *log based Change Data Capture* yang mengekstrak data melalui *data-streaming* dalam bentuk *event-log*, demi memastikan keterbaruan data pada *data mart*.
2. Mengukur kinerja dan keefektifan dari optimasi yang dilakukan.

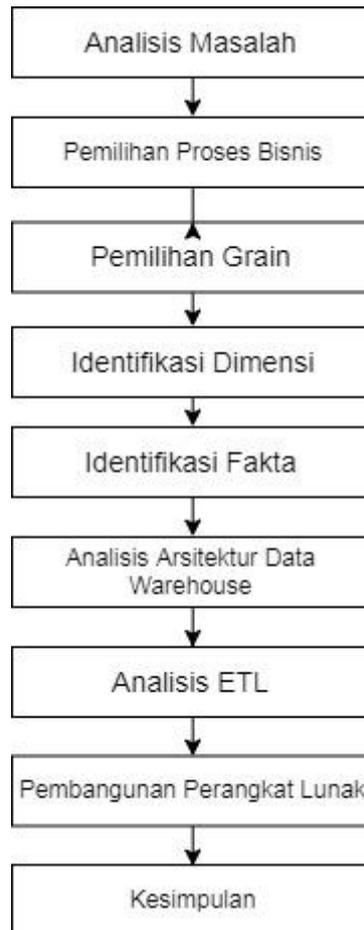
1.4 Batasan Masalah

1. Sistem yang akan dibangun akan berfokus pada proses ETL pada *data warehouse*.
2. Sistem operasional yang akan dijadikan sumber data pada penelitian adalah aplikasi operasional Personalia dan Payroll.
3. Sistem basis data yang digunakan pada implementasi sistem menggunakan PostgreSQL versi 9.6.

4. Metode *Change Data Capture* atau CDC digunakan untuk melakukan deteksi perubahan atau penambahan data pada data sumber lalu mengirimkannya ke *message broker*.
5. Proses CDC pada PostgreSQL akan dilakukan dengan memanfaatkan Debezium.
6. Sistem menggunakan Apache Kafka sebagai *message broker* yang digunakan untuk menyampaikan data operasional ke sistem *data warehouse*.
7. Pemodelan perangkat lunak menggunakan *Unified Modelling Language* (UML).

1.5 Metodologi Penelitian

Metodologi penelitian merupakan sebuah cara untuk mengetahui hasil dari sebuah permasalahan yang spesifik, dimana permasalahan tersebut yakni permasalahan penelitian. Adapun alur dalam melakukan penelitian adalah sebagai berikut:



Gambar 1.1 Metode Penelitian.

1.6 Metode Pengumpulan Data

1. Dokumen

Pengambilan data melalui dokumen tertulis maupun elektronik dari lembaga/institusi.

2. Studi Literatur

Pengumpulan data dengan mempelajari buku-buku karya ilmiah, jurnal, internet dan koleksi perpustakaan yang berkaitan erat dengan materi pembahasan dalam penulisan penelitian ini.

1.7 Metode Pembangunan Perangkat Lunak

Metode pembangunan perangkat lunak yang akan digunakan adalah *Extreme Programming*. Extreme programming (XP) adalah metodologi pengembangan perangkat lunak yang dimaksudkan untuk meningkatkan kualitas perangkat lunak dan responsif terhadap perubahan kebutuhan pelanggan, *Extreme Programming*

menekankan rilis dalam siklus pengembangan singkat, yang dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos-pos pemeriksaan di mana persyaratan pelanggan baru dapat diadopsi [8]. *Extreme Programming* memiliki 4 aktifitas dasar dalam prosesnya, yaitu:

1. *Designing*

Proses ini diperlukan untuk menghindari proses implementasi yang rumit dan sulit. Proses perancangan ini akan dilakukan secara terus menerus dengan aktivitas lainnya, terutama *coding* dan *testing*, dikarenakan adanya kemungkinan akan kebutuhan perubahan rancangan seiring proses pengembangan.

2. *Coding*

Pada *Extreme Programming* aktivitas ini dapat dilakukan bersamaan dengan aktivitas perancangan. Pada tahap ini akan dilakukan proses implementasi berdasarkan perancangan ke dalam bentuk kode program.

3. *Testing*

Proses pengujian akan dilakukan terhadap perangkat lunak yang dibangun, tes akan dilakukan secara otomatis dengan membuat *unit test* yang akan menguji bagian-bagian yang ada pada perangkat lunak.

4. *Listening*

Proses ini akan diisi dengan mendengarkan masukan atau umpan balik dari pengguna perangkat lunak untuk proses penyempurnaan perangkat lunak yang dibangun.

1.8 Sistematika Penulisan

Sistematika penulisan tugas akhir ini dibagi dalam beberapa bab dengan pokok pembahasan secara umum sebagai berikut.

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang, identifikasi masalah, maksud dan tujuan, batasan masalah, metodologi penelitian dan sistematika penelitian. Sebagai penunjang yang menjelaskan pokok bahasan mengenai penelitian yang dilakukan.

BAB 2 LANDASAN TEORI

Bab ini membahas berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan serta tinjauan terhadap penelitian-penelitian serupa yang telah atau pernah dilakukan sebelumnya termasuk sintesisnya.

BAB 3 ANALISIS DAN PERANCANGAN

Bab ini berisi tentang kebutuhan nonfungsional, analisis kebutuhan fungsional dan perancangan sistem yang diharapkan dapat menjelaskan keseluruhan dari apa yang dibangun pada penelitian ini.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini menjelaskan implementasi yang terdiri atas implementasi perangkat keras, implementasi perangkat lunak, implementasi aplikasi dan pengujian, yang terdiri atas pengujian alpha dan pengujian beta serta kesimpulan hasil pengujian.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yaitu apakah tujuan penelitian sudah terpenuhi atau belum, dan berdasarkan kesimpulan tersebut penulis mencoba memberi saran yang berguna untuk melengkapi dan menyempurnakan pengembangan aplikasi perangkat lunak untuk masa yang akan datang.