BAB 1
PENDAHULUAN

1.1. Latar Belakang

Dalam rangka mempercepat dan mempermudah pencarian sebuah informasi yang tersimpan pada sebuah basis data diperlukan nomor indeks[1]. Pemberian nomor indeks untuk baris data bersifat unik untuk membedakan satu dan lainnya. Bentuk nomor indeks dapat berupa barisan angka, atau kombinasi antara angka dengan huruf yang mewakili sebuah arti dan juga dapat memiliki pola tertentu. Contohnya penomoran akun prakiraan (*Chart of Account*) yang memiliki struktur **XYYYZZ** dimana **X** mewakili kode akun aset, **Y** mewakili kode kelompok dan **Z** mewakili detail akun (lihat lampiran 1). Dalam pembuatan penomoran untuk detail akun, bentuk **X** dan **Y** tidak akan berubah sedangkan **Z** akan berubah sesuai urutan pembuatan dengan kata lain **X** dan **Y** akan menjadi awalan (prefiks) untuk **Z**. Melihat kompleksitas dari bentuk nomor indeks, diperlukan sebuah fungsi yang menangani pemberian nomor indeks secara otomatis dapat menambahkan prefiks atau sufiks atau kombinasi prefiks dan sufiks.

Perangkat lunak *multi tenant* adalah perangkat lunak yang dirancang untuk melayani banyak pelanggan dengan kebutuhan yang berbeda. Menurut S. Aulbach Perangkat lunak *multi tenant* harus mampu memenuhi kebutuhan pelanggan yang beragam. Untuk memenuhi kebutuhan yang berbeda, perangkat lunak *multi tenant* harus dibangun dengan arsitektur yang fleksibel baik pada arsitektur perangkat lunaknya maupun pada skema data[2].

Permasalahan yang timbul jika dibuatkan suatu fungsi pada suatu aplikasi di perangkat lunak dengan arsitektur *multi tenant* adalah munculnya redundansi kode. Menurut Bayu Priyambadha redundansi kode atau duplikasi kode adalah sebuah kode dengan fungsi yang sama dalam peragkat lunak tanpa atau dengan disertai perubahan. Adanya redundansi pada kode perlu perhatian yang besar karena di satu sisi bisa menguntungkan namun di sisi lain dapat membahayakan. Contohnya apabila terjadi kesalahan pada sebuah blok kode namun kode tersebut

di salin ke

beberapa bagian file atau aplikasi, maka perbaikan harus dilakukan ke seluruh blok kode yang sama. Beberapa blok kode perlu disesuaikan dengan alur logika dimana blok kode itu ditempatkan. Blok kode yang tidak mengalami penyesuaian akan menimbulkan kesalahan proses[3].

Pada basis data terdapat fitur penomoran indeks otomatis. Karena fungsi ini melekat pada basis data, maka jika di implementasikan pada perangkat lunak multi-tenant tidak akan terjadi redundansi fungsi. Namun diperlukan basis data dengan performa yang baik. PostgreSQL yang kinerjanya lebih baik dibandingkan basis data open-source lainnya dapat melakukan proses *insert* 100.000 data dengan jenis yang sama PostgreSQL unggul 7,6 Kali lebih cepat dibandingkan MySQL. Sedangkan untuk proses *delete* hampir 2 kali lebih cepat[4]. Di tahun 2017 dan 2018 PostgreSQL dinobatkan sebagai "DBMS of The Year". Menurut survei yang dilakukan oleh DBEngineRank, PostgreSQL menunjukkan rata-rata pertumbuhan pengguna sekitar 16.5% pertahun dalam kurun waktu Januari 2016 hingga Januari 2019. Di tahun 2019 pengguna PostgreSQL naik 15.5 % dari tahun sebelumnya sedangkan pengguna MySQL turun sekitar 11.2% [5].

Penomoran indeks otomatis yang dapat ditangani oleh basis data PostgreSQL berbentuk auto-increment sedangkan berdasarkan pemaparan sebelumnya bentuk penomoran indeks memiliki prefiks atau bisa saja memiliki sufiks atau memiliki prefiks dan sufiks. Maka perlu dibuatkan sebuah fungsi tambahan pada basis data untuk menangai penomoran dengan penambahan prefiks atau sufiks atau prefiks dan sufiks.

Berdasarkan permasalahan yang timbul pada pemberian nomor indeks secara otomatis memiliki prefiks atau sufiks atau prefiks dan sufiks, maka akan dilakukan **"Pembangunan Extension ID Generator Dinamis"** yang dapat menyesuaikan kebutuhan.

1.2. Identifikasi Masalah

Berdasarkan dari uraian di bagian pendahuluan maka dapat di identifikasikan permasalahan yang terjadi, yaitu:

1. Fitur *auto-increment* untuk pemberian id secara otomatis pada basis data PostgreSQL tidak dapat menambahkan prefiks atau sufiks atau prefiks dan sufiks.
2. Redundansi blok kode yang digunakan untuk membuat fungsi penomoran pada perangkat lunak dengan arsitektur *multi tenant* dapat menimbulkan kesalahan proses apabila blok kode awal memiliki kesalahan sebelum di salin ke aplikasi-aplikasi yang berada pada arsitektur *multi tenant*.

1.3. Maksud dan Tujuan

Pada bagian ini dijelaskan maksud dan tujuan dilakukannya penelitian Pembangunan Extension ID Generator Dinamis.

1.3.1. Maksud

Untuk membangun Extension ID Generator Dinamis.

1.3.2. Tujuan

Adapun tujuan yang akan dicapai adalah:

1. Membuat fungsi yang dapat menambahkan prefiks atau sufiks atau prefiks dan sufiks pada penomoran secara otomatis.
2. Menggantikan fungsi penomoran yang ada pada setiap aplikasi pada arsitektur *multi tenant* dengan satu extension penomoran yang dipasang pada basis data.

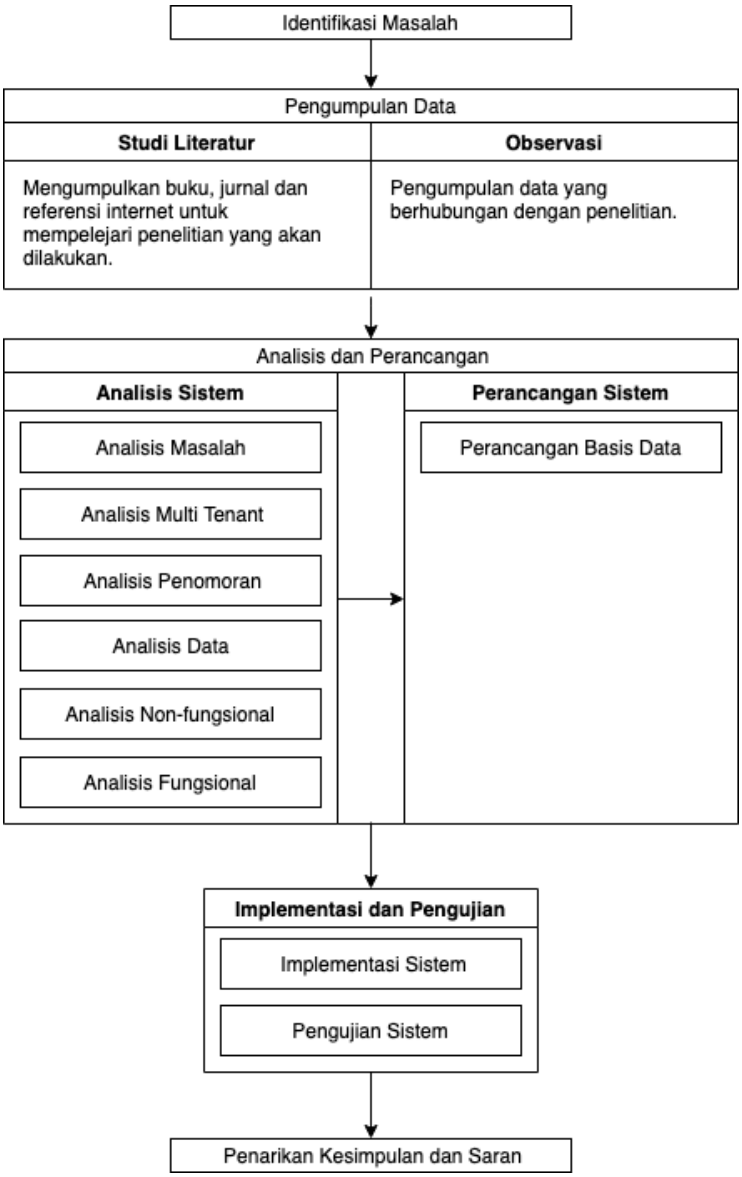
1.4. Batasan Masalah

1. Extension yang akan dibangun akan berfokus pada fungsi pemberian prefiks atau sufiks atau prefiks dan sufiks pada increment.
2. Studi kasus yang digunakan dalam pembangunan Extension ID Generator Dinamis adalah untuk pembuatan penomoran *Chart of Account* pada aplikasi Akunting dan pembuatan penomoran kontrak pegawai pada PT Mabra Technology Solution.
3. Implementasi extension akan dilakukan pada perangkat lunak dengan arsitektur *multi tenant*.
4. Pembangunan extension akan menggunakan bahasa GO.

5. Extension yang sudah diba'ngun di implementasikan pada basis data PostgreSQL versi 9.6.

1.5. Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini adalah metode penelitian terapan[6], dimana penelitian ini bertujuan untuk melihat faktor-faktor yang menjadi permasalahan lalu dibuatkan sebuah solusi yang merupakan hasil penelitian sebagai langkah pemecahan masalah[7].



Gambar 1.1 Alur kerja penelitian.

1.5.1. Metode Pengumpulan Data

1. Observasi

   Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan langsung terhadap permasalahan yang diambil. Pada kali ini penulis melakukan pengamatan terhadap sebuah domain aplikasi se-jenis yang berkitan dengan pemberian nomor indeks pada sistem akuntansi.

2. Studi Literatur

   Studi literatur yaitu metode pengumpulan data berupa literatur, jurnal, *paper*, dan bacaan lainnya yang berkaitan dengan penelitian. Pada kali ini penulis melakukan studi literatur pada dokumentasi, buku dan paper terkait proses akuntansi dan pembangunan extension pada PostgreSQL.

1.5.2. Metode Pembangunan Extension

Dalam pembuatan extension id generator, metode pembangunan yang digunakan adalah *waterfall* model sebagai tahanap pengembangan perangkat lunaknya. Proses *waterfall* model antara lain:

1. *Requirement analysis and definition*

   Tahap dimana pengumpulan kebutuhan telah terdefinisi secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

2. *System and software design*

   Tahap mendesain perangkat lunak yang dikerjakan setelah kebutuhan selesai secara lengkap.

3. *Implementation*

   Menerjemaahkan hasil tahap desain program kedalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan.

4. *Integration and system testing*

   Tahap integrasi atau penyatuan unit-unit program kemudian sistem diuji secara keseluruhan.

5. *Operation and maintenance*

   Tahap mengoprasikan program dilingkunganya dan melakukan pemeliharaan seperti penyesuaian atau perubahan karena adaptasi dengan situasi yang sebenarnya.

1.6. Sistematika Penulisan

Agar mencapai hasil yang baik dan terarah serta tidak menyimpang dari permasalahan yang ada maka penulis membuat sistematika laporan skripsi yang di uraikan sebagai berikut :

BAB 1 PENDAHULUAN

Pada bab ini penulis membahas tentang latar belakang masalah, identifikasi masalah, maksud dan tujuan. Betasan masalah, metodologi penelitian dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

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.

BAB 3 ANALISIS DAN PERANCANGAN

Bab ini berisi tentang analisis deskripsi sistem, analisis perancangan fungsional, analisis kebutuhan non fungsional dari perangkat lunak yang akan dibangun.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas implementasi perangkat keras dan perangkat lunak, pengujian perangkat lunak beserta kesimpulan dari hasil pengujian perangkat lunak yang dibangun.

BAB 5 KESIMPULAN

Bab ini berisi kesimpulan dan saran yang sudah diperoleh dari hasil penyusunan tugas akhir mengenai “PEMBANGUNAN EXTENSION ID GENERATOR DINAMIS PADA POSTGRESQL”.