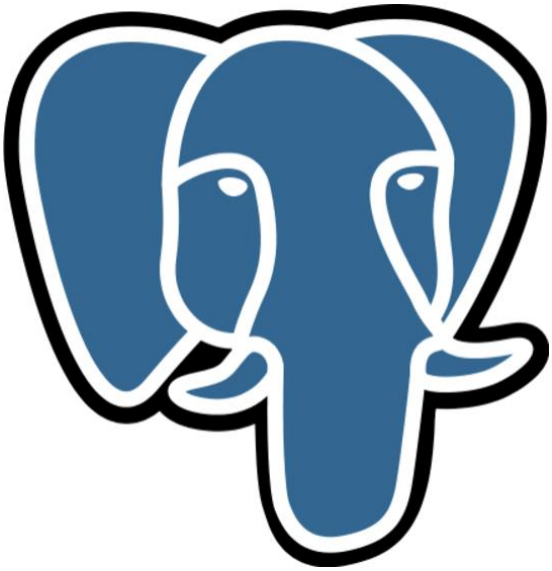
BAB III

TINJAUAN PUSTAKA

3.1. ETL Pipeline

ETL pipeline, yang merupakan singkatan dari *Extraction, Transform,* dan *Loading,* adalah kumpulan proses yang digunakan untuk mengambil data, memproses data, dan memuat data dari beberapa macam database yang berbeda ke satu database universal secara otomatis [1]. Proses ETL menjadi konsep yang kerap digunakan untuk memindahkan data ke *data warehouse* mulai dari tahun 1970 [2]. *ETL pipeline* diperlukan karena banyak sekali data yang didapatkan oleh perusahaan setiap hari. *Extraction* merupakan tahap dimana data diambil dari suatu sumber. Sumber data dapat disimpan di dalam file seperti csv, excel, dan txt, ataupun di dalam database server seperti melalui postgreSQL, mySQL, dan mongoDB. *Transform* dilakukan dengan melakukan normalisasi pada data yang telah didapatkan. Normalisasi dilakukan agar setiap data yang diambil mempunyai format yang sama. Proses yang dilakukan dapat berupa, mengubah nilai dengan arti yang sama menjadi satu nilai general, mengganti data dengan nilai kosong menjadi suatu nilai default, dan memastikan semua data pada setiap kolom menggunakan tipe data yang sama. *Loading* merupakan tahap dimana data dimuat ke target untuk langsung digunakan atau untuk disimpan.

3.2. PostgreSQL



Gambar III.1 Logo PostgreSQL [3]

PostgreSQL merupakan sistem manajemen database relasional (RDBMS) yang bersifat *open source*. PostgreSQL dikembangkan pada tahun 1996 dengan tujuan untuk membuat sistem database yang dapat mendukung tipe data, mendukung properti ACID, dan dapat dengan penuh mendeskripsikan hubungan antar tabel dengan menggunakan fitur sesedikit mungkin.

PostgreSQL biasa digunakan karena kemampuannya untuk menyimpan banyak sekali data pada suatu tabel dengan aman. Hal ini disebabkan karena, tidak seperti banyak database relasional lainnya seperti MySQL dan MariaDB, PostgreSQL merupakan database objek-relasional. Hal ini berarti, PostgreSQL dapat mendukung lebih banyak tipe data, fungsi, operator, dan lainnya dibandingkan dengan database lain. Akan tetapi, karena PostgreSQL merupakan database berbasis objek-relasional, PostgreSQL mempunyai kinerja yang lebih lambat dibandingkan dengan database berbasis noSQL seperti mongoDB. Hal ini disebabkan karena PostgreSQL melakukan query dengan membaca setiap baris pada sebuah tabel.

3.3. Python

Gambar III-2 Logo Python

Python adalah bahasa pemrograman *interpreted*, dimana suatu bahasa pemrograman akan menggunakan interpreter untuk menerjemahkan kode menjadi bahasa mesin secara bersamaan sewaktu program berjalan, yang bersifat tingkat tinggi. Python dirancang oleh Guido van Rossum pada awal tahun 1990 di Stichting Mathematisch Centrum (CW). Salah satu keuntungan dari Python adalah sebuah program dapat ditulis dengan sedikit mungkin. Hal ini disebabkan karena Python menggunakan data struktur tingkat tinggi dan simpel yang juga dapat melakukan pemrograman berbasis *object-oriented programming* secara efektif [4]. Keuntungan lain yang dimiliki Python dibandingkan dengan bahasa pemrograman lain yang kerap digunakan dalam bidang teknik adalah sintaksnya yang lebih jelas dan fungsionalitas yang lebih mudah dimengerti [5]. Salah satu kekurangan Python adalah Python terkenal mempunyai kecepatan yang lebih lambat dibandingkan dengan bahasa pemrograman lainnya yang dikompilasi seperti *C++* dan *Java*, terutama pada aplikasi yang menuntut banyak komputasi [6]. Python biasa digunakan pada bidang pekerjaan yang berhubungan dengan teknik. Salah satu alasannya adalah karena Python mempunyai banyak sekali *library* yang memudahkan pemrogram dalam melakukan pengolahan data, seperti dalam pembuatan model *machine learning*, perapian data dalam suatu database, dan analisa database [7].

3.4. Pandas

Gambar III-3 Logo Pandas [8]

Pandas, yang merupakan singkatan dari Panel Data, adalah salah satu *library* yang disediakan di Python. Dirancang pada tahun 2008 oleh Wes McKinney, Pandas dibuat dengan tujuan agar Python dapat digunakan sebagai aplikasi data analysis. Hal ini dilakukan dengan mendesain data struktur yang gampang digunakan, terintegrasi dengan library lain seperti matplotlib, dan beberapa fungsi lainnya yang dapat membantu melakukan data analisis seperti melakukan *indexing* secara hirearki dan meng-*align* data-data secara otomatis [9]. Pada tahun 2009, *Pandas* dibuat menjadi *open-source* sehingga lebih banyak orang yang dapat melakukan kontribusi untuk library tersebut [10].

Pandas biasa digunakan untuk membuat Objek DataFrame. DataFrame digunakan untuk membuat data tabular dari suatu variabel, seperti list atau dict, ataupun dari suatu file, seperti excel dan csv. Dibandingkan dengan tabel SQL ataupun *spreadsheet*, yang biasa digunakan untuk menampilkan data dalam bentuk dua dimensi seperti Pandas, Pandas dapat digunakan untuk memanipulasi data, seperti mengganti semua data dengan nilai NULL dengan nilai tertentu, ataupun menghapus semua data duplikat dari seluruh kolom atau beberapa kolom tertentu. Hal lain yang dapat dilakukan dengan menggunakan Pandas adalah menganalisis data dengan cepat dengan menggunakan beberapa baris kode. Salah satu hal penting lainnya yang dapat dilakukan oleh pandas adalah memvisualisasi data. Pandas memberikan opsi pada pengguna untuk dengan mudah membuat graf sesuai dengan yang diinginkan.

3.5. *Levenshtein Distance*

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Gambar III-4 Rumus Levenshtein Distance Algorithm [11]

Levenshtein Distance merupakan sebuah algoritma yang dikemukakan oleh matematikawan Soviet bernama Vladimir Levenshtein. Algoritma tersebut digunakan untuk mengetahui berapa banyak perubahan yang diperlukan pada setiap huruf dalam sebuah kalimat agar kalimat tersebut dapat mempunyai nilai yang sama dengan suatu kalimat lainnya. Hal ini berarti semakin besar nilai yang dihasilkan oleh algoritma *Levenshtein Distance,* semakin besar jarak perubahan antar 2 kalimat. Algoritma *Levenshtein Distance* biasa digunakan pada *Data Science* untuk berbagai macam contoh kasus, seperti untuk mengukur perbedaan dalam pelafalan dialek [12] dan deteksi plagiarisme [13].

3.6. Apache NiFi

Gambar III-5 Logo Apache NiFi [14]

NiFi merupakan program berbasis Java yang dikembangkan oleh *NSA*, dan kemudian diserahkan kepada Apache Foundation [15], untuk melakukan proses *Extract-Transform-Load* (ETL). NiFi dibuat dengan tujuan agar *ETL pipeline* dapat dikembangkan dengan cepat, gampang, dan dapat diandalkan. NiFi menyediakan interface GUI *drag and drop* yang dapat diakses melalui browser.

Terdapat lima komponen utama NiFi. Yang pertama adalah *Flowfile Processor*. *Flowfile Processor* adalah komponen utama NiFi yang digunakan

untuk melakukan suatu operasi. Secara default, NiFi menyediakan 293 processor. Setiap *Processor* mempunyai fungsi masing-masing, seperti melakukan query pada database, menjalankan script python, javascript, dan ruby, dan memuat file csv. Yang kedua adalah *FlowFile*. *Flowfile* adalah data yang bergerak dari satu prosesor ke prosesor lainnya. Setiap flowfile terbagi menjadi dua bagian, atribut dan konten. Atribut memberikan informasi mengenai nama file, ukuran file, durasi file dalam antrian, dan lainnya. Konten bersisi informasi mengenai isi data yang dimuat dalam *FlowFile* tersebut. Yang ketiga adalah *Connection*. *Connection* digunakan untuk menghubungkan satu processor dengan yang lainnya, sehingga jika suatu *FlowFile* selesai di proses, *FlowFile* dapat dikirim ke *processor* selanjutnya Yang keempat adalah *Process Group. Process Group* biasa digunakan untuk mengelompokkan beberapa *processor* dalam satu grup yang sama. Hal ini dilakukan agar alur *ETL* dapat dibaca dengan mudah dan gampang dimengerti. Yang terakhir adalah *Flow Controller*. *Flow Controller* digunakan untuk mengontrol alokasi memori pada setiap *Processor*, menyimpan setiap credential yang digunakan untuk mengakses sebuah servis, dan lainnya [16].