Anggota Kelompok

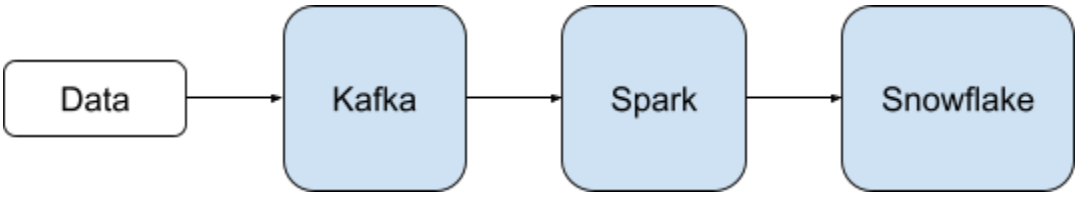
1. 13520007 - Nadia Mareta Putri Leiden
2. 13520050 - Felicia Sutandijo
3. 13520160 - Willy Wilsen

Penjelasan terkait data sumber dan data final hasil pemrosesan

Sumber data yang dipilih pada tugas ini adalah dataset "UK Flood Data" yang dapat dilihat pada pranala berikut https://environment.data.gov.uk/flood-monitoring/doc/reference. Dataset ini memuat data banjir area-area di Inggris secara *real-time.* Data diambil melalui API, dengan format *array of object* data banjir sebuah area. Data banjir sebuah area sendiri memiliki beberapa atribut, seperti *id, description, eaAreaName, eaRegionName, floodArea (id, county, notation, polygon,* dan *riverOrSea), floodAreaID, isTidal, message, severity, severityLevel, timeMessageChanged, timeRaised,* dan *timeSeverityChanged.* Atribut-atribut tersebut juga dapat difilter melalui API yang disediakan, namun kami memilih untuk melakukan *streaming* data secara keseluruhan. Yang ingin ditunjukkan dari dataset ini adalah jumlah masing-masing *flood severity level* untuk setiap area setiap 10 detik.

Penjelasan alur secara umum serta arsitektur yang digunakan

Berikut ilustrasi alur secara umum serta arsitektur yang digunakan pada tugas ini, dengan penjelasan di bawah.



Pertama-tama, data diambil dari API yang disediakan dengan request GET oleh Kafka Producer setiap 10 detik. Data yang dikirimkan dari Kafka Producer akan di-*consume* oleh Spark. Spark akan melakukan *streaming* pada data yang di-*consume*, mengolah data tersebut, dan melakukan *insert* ke Snowflake. Data yang telah dimasukkan ke dalam Snowflake akan divisualisasikan.

Penjelasan masing-masing komponen arsitektur dan alasan konfigurasinya

1. Kafka
   Pada Kafka Producer, akan dikirimkan data dari API yang disediakan dengan request GET setiap 10 detik. Alasan konfigurasinya adalah untuk menyediakan data yang akan di-*streaming* oleh Spark.

2. Spark
   Pada Spark, akan melakukan *streaming* pada data yang di-*consume* dari Kafka Consumer, kemudian mengolah data ke dalam bentuk yang diinginkan. Alasan konfigurasinya adalah untuk mendapatkan data sesuai kebutuhan yang kemudian disimpan dalam warehouse yang dalam hal ini adalah Snowflake.
3. Snowflake
   Pada Snowflake, akan dilakukan pengambilan data yang telah disimpan untuk digunakan kembali. Alasan konfigurasinya adalah untuk mendapatkan dan memvisualisasikan data yang telah diolah sesuai kebutuhan.

Lessons learned yang dipelajari selama implementasi dan eksplorasi

1. Pembuatan *flow* dari data mentah yang diproses secara real time hingga menjadi data yang siap divisualisasikan
2. Integrasi data mentah dari Kafka ke Spark dan data yang telah diolah dari Spark ke Snowflake
3. Visualisasi data yang diperoleh dari Snowflake