BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam Data Science, sudah banyak metode untuk memprediksi data dengan model machine learning. Tetapi data yang biasa digunakan adalah data yang statis dan tidak bergerak. Bagaimana jika data dihasilkan secara *real-time*? Dapat terjadi penyimpangan perilaku data, atau juga disebut *concept drift* di mana interpretasi orang suatu data berubah seiring waktu, menyebabkan prediksi yang dilakukan model machine learning menjadi tidak akurat (Thesis and Zhang 2018). Dalam cloud computing, sangat penting untuk menjaga sumber daya agar tidak terbuang. Sebagai contoh dalam kasus Vertical Scaling, penggunaan CPU sangat penting diamati kaitannya dengan proses scaling yang dilakukan. Tetapi karena proses scaling terjadi secara real-time, maka perlu diidentifikasi apakah terjadi penyimpangan data saat scaling dilakukan. Peningkatan skalabilitas kemungkinan menyebabkan data yang drift karena meningkatnya atau menurunnya beban Cloudlet yang dikerjakan dalam VM.

Dalam *machine learning*, terdapat tiga tantangan fundamental (Khamassi et al. 2015). Yang pertama adalah banyaknya data yang tergenerasi seiring waktu. Data ini kemungkinan tidak terbatas, dan tidak praktis untuk menyimpan semuanya. Tantangan yang kedua adalah begitu cepatnya datangnya *data stream* dalam beberapa aplikasi yang membutuhkan pelayanan *real-time*. Kedua tantangan ini awalnya ditangani oleh algoritma *learning* online karena algoritma tersebut dapat

belajar dari *instance* yang tidak ada batasnya menggunakan memori dan waktu yang terbatas, dan algoritma tersebut tidak mengasumsi ketersediaan dari *training set* sebelum learning. Tetapi ada tantangan ketiga yang muncul saat *data stream* yang datang tidak stasioner dan berubah seiring waktu. Ini disebut penyimpangan perilaku data.

Secara umum, penyimpangan perilaku data atau *concept drift* dalam analisis data *stream* adalah perubahan pola data yang berubah seiring waktu karena perbedaan aktivitas user ataupun operasional. Ini menyebabkan berubahnya pola traffic data yang sebelumnya normal menjadi berbeda. Contohnya adalah preferensi pengguna yang berubah, kondisi cuaca yang bervariasi, dan perubahan ekonomi (Grulich et al. 2018). Ini menyebabkan teknik-teknik *machine learning* untuk analisis *data stream* menjadi tidak akurat. Dari kecepatannya, ada dua tipe penyimpangan perilaku data, yaitu penyimpangan tiba-tiba dan penyimpangan gradual (Sun et al. 2016). Penyimpangan tiba-tiba ditandai dengan banyaknya perubahan antara distribusi kelas yang mendasar dan *instance* yang masuk dalam waktu yang pendek. Penyimpangan gradual ditampilkan oleh banyaknya waktu untuk mengamati perubahan yang besar dalam perbedaan antara distribusi kelas yang mendasar dan *instance* yang masuk. Metode yang sudah ada mungkin hanya digunakan untuk salah satu dari dari kedua tipe ini, tetapi dalam dunia nyata, bisa ada lebih banyak dari satu tipe penyimpangan perilaku data dalam *data stream*. Maka dari itu, dapat mengikuti dan mengadaptasi ke berbagai tipe penyimpangan sangat berguna.

Untuk penelitian ini, data akan diambil dari simulasi Vertical VM Scaling dalam *CloudSim* di mana penggunaan CPU dalam simulasi akan dianalisis apakah terjadi driftnya dan mengapa. Data ini nanti akan diimplement dengan *scikitmultiflow* dalam Python, sebuah modul Python yang dibangun dari *SciPy* untuk machine learning dan dapat diinstal dengan *pip*. Atau data bisa juga diambil secara *real-time* menggunakan *OpenStack* dengan membaca data yang sedang di tengah proses transfer antara server. Tetapi untuk penelitian ini, CloudSim akan digunakan. Simulasi CloudSim dapat digunakan untuk mewakili lingkungan cloud yang nyata karena CloudSim memperbolehkan pengguna dan provider Cloud untuk melakukan modeling, simulasi, dan eksperimentasi infrastruktur dan layanan aplikasi yang baru secara mudah. CloudSim dapat melakukan modeling dan simulasi untuk data center cloud computing pada skala besar, simulasi sumber daya komputasi yang *energy-aware*, memasukkan elemen simulasi secara dinamis, memberhentikan dan melanjutkan simulasi, dan pengguna dapat menetapkan policy untuk alokasi jumlah host ke VM dan policy untuk alokasi sumber daya dalam host ke VM. Lalu penyimpangan perilaku data dapat dideteksi secara *real-time* dengan metode deteksi algoritma ADWIN atau *adaptive windowing* (Grulich et al. 2018). Algoritmanya bekerja dengan membuka *window* adaptif yang merupakan basis untuk komputasi model machine learning. ADWIN memperbesar *window* tersebut dan menambahkan *tuple* yang paling baru, asalkan tidak ada penyimpangan yang dideteksi. Tuple adalah sekumpulan data berisi object yang beraneka ragam terlampir dalam "( )". (Kanetkar and Kanetkar 2019). Oleh karena itu, model ini

dapat bergantung ke data training yang terus bertambah. ADWIN akan mengecilkan *window* tersebut dengan menghapus *tuple* lama saat penyimpangan dideteksi.

1.2. Rumusan Masalah

a. Bagaimana mengidentifikasi terjadinya penyimpangan data pada lingkungan operasi virtual?
b. Apa upaya yang dapat dilakukan untuk mengidentifikasi drifting yang terjadi dalam proses vertical scaling pada lingkungan virtual?
c. Apakah mempergunakan simulator CloudSim dapat menjelaskan proses skalabilitas dan bagaimana utilisasi terhadap elemen komputasi terjadi?

1.3. Batasan Masalah

a. Deteksi konsep penyimpangan dalam sistem virtual
b. Identifikasi konsep penyimpangan data dengan algoritma *ADWIN* dari modul *skmultiflow* menggunakan Python
c. Batasan lingkungan yang digunakan menggunakan data penggunaan CPU dalam simulasi scaling VM secara vertikal dalam CloudSim

1.4. Tujuan Penelitian

Tujuan penelitian tugas akhir ini adalah mengidentifikasi penyimpangan yang terjadi saat menyimulasikan skenario *scaling* VM secara vertikal menggunakan CloudSim. Data yang diperhatikan di sini adalah CPU atau processor,

di mana penggunaan CPU akan diambil data dari awal simulasi sampai akhir. Untuk menggenerasi lebih banyak data, beberapa nilai parameter simulasi diubah, tetapi tidak bisa hanya diganti satu parameter misalkan hanya parameter VM. Setiap parameter, VM, PEs, dan Cloudlet harus diubah untuk membuat simulasi yang tepat, karena sumber daya yang terlalu banyak tanpa Cloudlet yang diubah akan membuat simulasi hanya mengeluarkan data dengan nilai kecil, tetapi sumber daya yang kurang sedangkan Cloudlet yang terlalu banyak dapat mengeluarkan nilai yang terlalu tinggi dan bahkan bisa menyebabkan crash dalam simulasi. Maka dari itu, elemen VM, PEs, dan Cloudlet harus disesuaikan dengan satu sama lain agar simulasi dapat berjalan seimbang.

1.5. Metodologi

Untuk menyelesaikan masalah pada penelitian ini, metode-metode yang digunakan adalah:

1. Melakukan studi pustaka untuk mempelajari teori mengenai penyimpangan perilaku data dan *cloud computing*
2. Menjalankan simulasi CloudSim
3. Mengoleksi data dan *cleansing* nilai yang tidak diperlukan dari output data
4. Data yang sudah di*cleanse* difitting ke dalam algoritma ADWIN, menggunakan modul *skmultiflow* dalam Python

1.6. Sistematika Penulisan

Laporan tugas akhir ini disusun dengan menggunakan sistematika penulisan yang dapat dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Bab ini dimulai dengan penjelasan mengenai latar belakang penelitian yang berjudul IDENTIFIKASI TERJADINYA PENYIMPANGAN PADA LINGKUNGAN VIRTUAL DENGAN METODE *ADAPTIVE WINDOWING*. Kemudian dalam bab ini juga dibahas penentuan rumusan dan batasan masalah serta penjelasan tujuan penelitian dan metodologi yang digunakan pada penelitian ini. Pada akhir bab ini dijelaskan mengenai sistematika penulisan yang digunakan.

BAB II LANDASAN TEORI

Bab ini berisi tentang teori-teori yang digunakan sebagai acuan dalam merancang dan mengembangkan tugas akhir meliputi *Cloud Computing* dan Penyimpangan Perilaku Data (Concept Drift).

BAB III PERANCANGAN SISTEM

Bab ini berisi tentang aliran rencana kerja yang akan dilakukan, metode-metode yang digunakan dalam penelitian, seperti cara menggenerasi data, mengoleksi data, *cleansing* data, sampai *fitting* data untuk menjalankan algoritma ADWIN agar *drift* dapat terdeteksi.

BAB IV HASIL DAN ANALISA

Bab ini berisi tentang proses penelitian yang dilakukan, seperti output program dengan nilai *default*, serta analisa output tersebut, output program dengan parameter yang direncanakan, analisa outputnya, lalu lokasi *drift* dalam output, serta analisa *drift* yang terjadi dalam data.

BAB V KESIMPULAN DAN SARAN

Bab ini adalah bab terakhir yang berisi kesimpulan yang didapatkan dari penelitian ini, serta saran-saran yang dapat dilakukan untuk mengembangkan penelitian ini di masa depan.