

BAB I

PENDAHULUAN

1.1 Latar Belakang

Collaborative filtering adalah algoritme yang populer digunakan untuk membangun sistem rekomendasi. Sistem rekomendasi dibuat untuk membantu pengguna dalam memilah informasi dan memberi saran-saran untuk mengidentifikasi *item-item* menarik yang sesuai dengan preferensi pengguna (Martín-Vicente dkk, 2014). Sistem rekomendasi sangat bermanfaat dalam pencarian dan pemilihan informasi bagi pengguna yang tidak memiliki waktu dan pengetahuan yang cukup untuk memilah informasi. Dengan pertimbangan berbagai manfaat yang dimilikinya, sistem rekomendasi kemudian diaplikasikan secara luas di berbagai bidang mulai dari *e-commerce* (Linden dkk, 2003), *social network* (Xu dkk., 2014), *e-tourism* (Gavalas dkk, 2013), rekomendasi dokumen (Wiranto, 2014), rekomendasi film (Leben, 2008), rekomendasi musik (Sobhanam dan Mariappan, 2013), rekomendasi berita (Claypool dkk, 1999), dan lain-lain.

Collaborative filtering bekerja dengan cara mempelajari data nilai rating yang telah diberikan pengguna untuk sejumlah *item*, mengenali kemiripan *item* atau kemiripan pengguna berdasarkan data nilai rating, untuk kemudian menghasilkan rekomendasi baru (Candillier dkk, 2007). Pada metode *collaborative filtering*, pengguna memberikan penilaian untuk suatu *item* yang dinyatakan dalam bentuk nilai rating. Contohnya adalah sistem rekomendasi film, seorang pengguna memberikan nilai rating untuk film-film yang telah dilihat. Pengguna bebas menentukan kriteria penilaian untuk suatu film, namun hanya satu nilai rating yang diberikan untuk sebuah film. Sistem rekomendasi kemudian akan memprediksi nilai rating untuk film-film lain yang belum dilihat oleh pengguna tersebut dengan memanfaatkan data nilai rating dari pengguna-pengguna lain yang memiliki kemiripan preferensi (Adomavicius dkk, 2011).

Metode *collaborative filtering* kemudian dikembangkan dengan menggunakan lebih dari satu kriteria untuk menilai suatu *item*. Sebagai contoh dalam rekomendasi film, pengguna menilai sebuah film dalam berbagai kriteria. Pengguna dapat menilai naskah cerita, akting para aktor, sutradara, dan efek visual yang terdapat pada suatu film. Setiap kriteria tersebut diberikan nilai rating, sehingga untuk sebuah film terdapat banyak nilai rating sesuai dengan jumlah kriteria penilaian. Metode *collaborative filtering* yang melibatkan banyak kriteria ini disebut sebagai *multi-criteria collaborative filtering*. *Multi-criteria collaborative filtering* dinilai mempunyai kualitas rekomendasi yang baik karena informasi rating multikriteria yang digunakan dapat mewakili preferensi pengguna dengan lebih baik daripada *collaborative filtering* klasik yang menggunakan kriteria tunggal (Adomavicius dkk, 2011). Walaupun memiliki kelebihan dalam hal kualitas rekomendasi, *multi-criteria collaborative filtering* khususnya yang menggunakan pendekatan *memory-based* juga memiliki kekurangan yaitu masalah skalabilitas. Ketika jumlah data yang diolah semakin meningkat, kinerja *multi-criteria collaborative filtering* menjadi kurang maksimal dibandingkan ketika jumlah data yang diolah masih sedikit (Pagare dan Shinde, 2013). Salah satu indikasi adanya permasalahan skalabilitas ini adalah meningkatnya waktu yang dibutuhkan untuk memberikan rekomendasi kepada pengguna ketika *volume* data sistem rekomendasi bertambah.

Skalabilitas sistem rekomendasi *collaborative filtering* menjadi tema yang banyak dibahas dalam berbagai penelitian dengan berbagai metode dan pendekatan yang diusulkan, salah satu di antaranya adalah pendekatan *scale-out*. Pada pendekatan *scale-out*, dilakukan penambahan *node* komputer untuk menjalankan sistem rekomendasi agar diperoleh skalabilitas yang baik. Implementasi metode *scale-out* yang banyak dilakukan dalam penelitian sebelumnya adalah dengan menggunakan Hadoop MapReduce seperti yang dilakukan oleh Zhao dan Shang (2010), Jiang dkk (2011), Schelter dkk, (2012), Ghuli dkk (2014), dan Shang dkk (2014) untuk memperoleh skalabilitas yang

bagus dari sistem rekomendasi *collaborative filtering* kriteria tunggal. Penelitian lain dilakukan oleh Casey (2014) yang menggunakan Apache Spark untuk mengimplementasikan pendekatan *scale-out* guna mengatasi masalah skalabilitas sistem rekomendasi *collaborative filtering* kriteria tunggal. Penggunaan Apache Spark oleh Casey (2014) dilatarbelakangi penilaian terhadap Hadoop MapReduce yang banyak menggunakan proses *read* dan *write* ke *harddisk* dinilai kurang cocok untuk implementasi algoritme *collaborative filtering* yang memiliki banyak langkah iteratif.

Penelitian ini berusaha untuk mengimplementasikan algoritme *multi-criteria collaborative filtering* di *cluster* Apache Spark guna mengatasi masalah skalabilitas yang dihadapi algoritme *multi-criteria collaborative filtering*.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah sebagai berikut :

Bagaimana kinerja yang diperoleh dari implementasi sistem rekomendasi *multi-criteria collaborative filtering* di *cluster* menggunakan *framework* Apache Spark.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Pendekatan perhitungan similaritas antar hotel yang digunakan adalah *item-based similarity*.
2. Data yang digunakan pada penelitian ini adalah rating multikriteria hotel dari situs TripAdvisor.
3. Ukuran kinerja yang digunakan pada penelitian ini adalah skalabilitas yang didasarkan pada perbandingan waktu yang dibutuhkan algoritme untuk menghasilkan rekomendasi di *cluster* terhadap peningkatan jumlah *resource* (jumlah *node worker*).

1.4 Keaslian Penelitian

Pada penelitian-penelitian sebelumnya untuk mengatasi masalah skalabilitas sistem rekomendasi, algoritme yang digunakan adalah *collaborative filtering* kriteria tunggal. Sedangkan pada penelitian ini digunakan algoritme *multi-criteria collaborative filtering*. Implementasi pendekatan *scale-out* untuk mengatasi masalah skalabilitas sistem rekomendasi pada penelitian-penelitian sebelumnya dilakukan menggunakan Hadoop MapReduce, sedangkan pada penelitian ini digunakan Apache Spark. *Framework* Apache Spark dipilih karena dinilai lebih cocok untuk implementasi algoritme *collaborative filtering* yang memiliki banyak langkah iteratif (Casey, 2014).

1.5 Tujuan Penelitian

Tujuan penelitian ini adalah untuk membuat sistem rekomendasi dengan algoritme *multi-criteria collaborative filtering* yang dijalankan pada *cluster* Apache Spark.

1.6 Manfaat Penelitian

Penelitian ini diharapkan dapat mengatasi masalah skalabilitas yang dihadapi sistem rekomendasi yang menggunakan algoritme *multi-criteria collaborative filtering*.

1.7 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yaitu studi pustaka, pengumpulan data, perancangan sistem, implementasi, pengujian dan analisis hasil.

1. Studi pustaka.

Pada tahap studi pustaka dilakukan pencarian buku, *paper*, tesis, dan sumber tertulis lainnya yang terkait dengan topik penelitian, permasalahan yang dibahas, serta publikasi penelitian sebelumnya yang mengangkat permasalahan yang serupa. Sumber tertulis yang dikaji adalah yang membahas mengenai sistem rekomendasi, algoritme *collaborative filtering*, sistem rekomendasi *multi-criteria collaborative filtering*, skalabilitas sistem rekomendasi, Apache Spark, serta penelitian lain untuk mengatasi masalah skalabilitas sistem rekomendasi.

2. Pengumpulan data.

Pengumpulan data dilakukan untuk mendapatkan data yang dipakai dalam penelitian. Data yang digunakan berupa data rating multikriteria hotel yang diambil dari situs tripadvisor.com menggunakan teknik *web crawling*.

3. Perancangan sistem.

Dalam penelitian ini diusulkan penggunaan *framework* Apache Spark untuk mengatasi masalah skalabilitas sistem rekomendasi yang menggunakan algoritme *multi-criteria collaborative filtering*. Terdapat tiga tahap di dalam algoritme *multi-criteria collaborative filtering* yaitu menghitung nilai similaritas antar hotel, menghitung rating kriteria individual, dan pembangkitan rekomendasi. Ketiga tahap tersebut akan dijalankan di *cluster* Apache Spark. Selain itu ditentukan pula rancangan jaringan komputer *cluster* yang digunakan untuk implementasi sistem.

4. Implementasi.

Hasil perancangan algoritme *multi-criteria collaborative filtering* diwujudkan dalam bentuk program sistem rekomendasi yang akan dijalankan di *cluster* Apache Spark.

5. Pengujian dan analisis hasil.

Pengujian dilakukan untuk mengetahui skalabilitas algoritme *multi-criteria collaborative filtering* di *cluster* Apache Spark. Untuk keperluan pengujian

digunakan beberapa *cluster* dengan jumlah *node worker* yang berbeda. Program sistem rekomendasi kemudian dijalankan di masing-masing *cluster* untuk menghasilkan rekomendasi bagi seorang pengguna yang dipilih. Waktu yang dibutuhkan program untuk menghasilkan rekomendasi kemudian dibandingkan untuk menghitung nilai *speedup*.

1.8 Sistematika Penulisan

Tesis ini berisikan 7 (tujuh) bab dengan rincian setiap bab adalah sebagai berikut :

Bab I Pendahuluan

Bab ini berisi uraian tentang latar belakang penelitian, rumusan masalah, batasan masalah, keaslian penelitian, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

Bab II Tinjauan Pustaka

Bab ini berisi pemaparan penelitian-penelitian yang berkaitan dengan masalah skalabilitas sistem rekomendasi menggunakan berbagai metode yang telah dipublikasikan sebelumnya yang digunakan sebagai referensi dalam penelitian ini.

Bab III Landasan Teori

Bab ini berisi uraian teori-teori dasar yang berkaitan dengan topik sistem rekomendasi menggunakan algoritme *collaborative filtering*, *multi-criteria collaborative filtering*, permasalahan skalabilitas sistem rekomendasi, serta metode yang digunakan untuk menyelesaikan permasalahan skalabilitas.

Bab IV Analisis dan Perancangan

Bab ini membahas mengenai analisis data yang digunakan dalam penelitian, serta perancangan algoritme *multi-criteria collaborative filtering* untuk dijalankan pada *cluster* Apache Spark.

Bab V Implementasi

Pada bab ini terdapat penjelasan mengenai implementasi sistem rekomendasi *multi-criteria collaborative filtering* dari hasil analisis ke dalam bahasa pemrograman untuk kemudian hasilnya dijalankan menggunakan *framework* Apache Spark.

Bab VI Hasil dan Pembahasan

Bab ini menjelaskan hasil penelitian yang diperoleh, yang berupa waktu yang dibutuhkan oleh program sistem rekomendasi *multi-criteria collaborative filtering* untuk membangkitkan rekomendasi, disertai dengan pembahasan mengenai skalabilitas sistem rekomendasi *multi-criteria collaborative filtering* berdasarkan hasil yang diperoleh.

Bab VII Penutup

Bab ini berisi kesimpulan yang diambil dari hasil penelitian yang dilakukan beserta saran-saran untuk penelitian selanjutnya.