

# 1. Pendahuluan

## 1.1. Latar Belakang Masalah

*Recommender system* telah digunakan oleh berbagai situs e-commerce untuk merekomendasikan produk, barang, film, musik dll. *Recommender system* adalah alat penting dalam e-commerce di Web [1]. Saat ini, *Recommender system* sedang digunakan oleh banyak data pelanggan di database komersial yang ada, dan tersedia di situs-situs jejaring sosial. *Recommender system* yang paling sukses untuk merekomendasikan adalah pendekatan berbasis *collaborative filtering*. Dalam rangka untuk menghasilkan rekomendasi, sistem penyaringan kolaboratif pada dasarnya perlu membandingkan objek yang berbeda seperti item terhadap pengguna. Ada dua pendekatan utama untuk membantu perbandingan dalam *collaborative filtering*, *neighborhood approach* dan *latent factor models*. Metode *neighborhood* didasarkan pada komputasi hubungan antara pengguna atau item. Pendekatan *Single Value Decomposition* (SVD), *latent factor models*, yang mengkonversi pengguna dan item untuk faktor laten yang sama yang sebanding dengan satu sama lain. Pertumbuhan yang luar biasa dari pelanggan dan produk karena web dan e-commerce merupakan situs sosial yang menciptakan dua tantangan utama untuk sistem recommender.

Tantangan pertama adalah bahwa bagaimana meningkatkan kualitas rekomendasi bagi pelanggan. Jika kualitas *recommender system* baik, maka pelanggan dapat mempercayai, membeli produk, jika kualitas *recommender system* buruk, maka pelanggan tidak akan mempercayai *recommender system* lagi. Tantangan kedua adalah bahwa bagaimana meningkatkan skalabilitas dari algoritma *collaborative filtering*. Entah bagaimana ada konflik dalam dua tantangan ini. Jika algoritma menghabiskan lebih sedikit waktu untuk mencari sebuah *neighbors*, hal ini akan lebih *scalable* tetapi akan mendapatkan kualitas buruk. Hal ini perlu mempertimbangkan dua tantangan ini secara bersamaan sehingga akan menemukan solusi yang praktis. Oleh karena itu membutuhkan teknologi baru yang berguna dan secara dramatis meningkatkan skalabilitas dari sistem recommender. Banyak peneliti telah menyarankan bahwa *Singular Value*

*Decomposition* (SVD) [2] merupakan teknologi di sebagian besar kasus. Pendekatan berbasis SVD dapat menghasilkan hasil yang jauh lebih baik daripada sebagian besar algoritma penyaringan kolaboratif tradisional saat diuji pada dataset MovieLens. Tapi ada beberapa keterbatasan yang serius ketika menerapkan pendekatan berbasis SVD untuk merekomendasikan yang membuatnya kurang cocok untuk diterapkan dalam sistem e-commerce skala besar. Langkah penghitungan metode faktorisasi matrik sangat mahal karena memakan banyak memori dan waktu. Ini adalah masalah besar untuk mencapai skalabilitas yang tinggi sambil menghasilkan akurasi prediksi yang baik. Dalam dunia nyata aplikasi e-commerce, sejumlah besar pelanggan hanya membeli atau menilai persentase yang sangat kecil dari produk, yang merupakan masalah nyata. Pengurangan dimensi dalam *recommender system* digunakan karena dua masalah ini. Yang akan membantu untuk meningkatkan ketepatan rekomendasi, dan mengurangi kompleksitas perhitungan *real time*.

Tugas akhir ini membahas algoritma pengurangan dimensi, menjelaskan tentang konsep popularitas sosial dengan model yang terintegrasi, menyajikan prosedur evaluasi eksperimental dan hasil prediksi.

## **1.2. Perumusan Masalah**

Permasalahan yang akan dibahas pada tugas akhir ini adalah analisis dan implementasi algoritma SVD++ pada *recommender system* berbasis collaborative filtering. Adapun *research questions* dalam tugas akhir ini yaitu:

- a. Bagaimana perbedaan algoritma SVD dan SVD++ dalam implementasi pada *recommender system* berbasis *collaborative filtering*.
- b. Bagaimana *implicit feedback* dapat mempengaruhi hasil rating yang lebih baik pada algoritma SVD++.
- c. Bagaimana pengaruh faktor pada algoritma dan SVD++.

## **1.3. Batasan Masalah**

Batasan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Batasan masalah Tugas Akhir ini untuk pengujian *collaborative filtering* pada dataset filmtrust menggunakan metode SVD dan SVD++.

#### **1.4. Tujuan**

Berdasarkan rumusan masalah di atas, maka tujuan akhir tugas akhir ini adalah:

- a. Membandingkan dan menganalisis algoritma SVD dan SVD++.
- b. Mengetahui pengaruh *implicit feedback* pada algoritma SVD++.
- c. Mengetahui pengaruh parameter faktor pada algoritma SVD++.

#### **1.5. Metodologi Penyelesaian Masalah**

- a. Studi Literatur

Melakukan pencarian terhadap literatur yang sesuai dengan permasalahan meliputi perbandingan terhadap berbagai metode pembobotan term.

- b. Pengumpulan data

Melakukan pengumpulan data untuk mendukung pengimplementasian berbagai metode *term weighting*.

- c. Analisis dan Perancangan

Menganalisis kebutuhan sistem, dan proses-proses yang terjadi pada sistem yang dibangun.

- d. Implementasi

Membangun perangkat lunak sesuai dengan hasil perancangan pada bahasa pemrograman java dan melakukan pengujian terhadap data yang didapatkan.

- e. Analisis

Menganalisis performansi metode pembobotan term terhadap pengkategorian teks.

- f. Kesimpulan

Melakukan pengambilan kesimpulan dari hasil testing yang didapatkan.

- g. Pembuatan Laporan Tugas Akhir

Mendokumentasikan kegiatan yang dilakukan pada pengerjaan tugas akhir ke dalam sebuah laporan.

## **1.6. Sistematika Penulisan**

Tugas akhir ini akan dilaksanakan dengan memiliki sistematika penulisan sebagai berikut:

### **BAB 1 Pendahuluan**

Bab ini berisi mengenai penjelasan secara umum dalam tugas akhir, yang meliputi latar belakang masalah, perumusan masalah, tujuan, dan metodologi penyelesaian masalah.

### **BAB 2 Dasar Teori**

Bab ini akan menjelaskan landasan teori yang berkaitan dengan pembangunan perangkat lunak.

### **BAB 3 Perancangan dan Implementasi**

Bab ini berisi rancangan sistem sesuai dengan kasus yang dipilih, dan dilanjutkan dengan pengimplementasiannya.

### **BAB 4 Pengujian dan Analisis**

Bab ini akan menjelaskan tentang pengujian terhadap perangkat lunak dan melakukan analisis terhadap *output* pengujian sebelumnya.

### **BAB 5 Kesimpulan dan Saran**

Bab ini berisi kesimpulan dari tugas akhir mengenai perangkat lunak yang dibangun dan berisi juga saran bagi pengguna yang memakai atau yang ingin mengembangkan penelitian lebih lanjut.