

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan rumusan masalah dari hasil penelitian ini, maka diperoleh kesimpulan sebagai berikut:

1. Langkah-langkah yang dilakukan dalam pengujian *cosine* dengan *pearson similarity* berbasis *item based collaborative filtering* menggunakan model *K-Nearest Neighbor*, yaitu pengumpulan data, pembersihan data, mencari nilai *similarity*, pengelompokkan *K-Nearest-Neighbor*, *Generating missing rating*, dan pengujian evaluasi RMSE.
2. *Pearson* adalah perhitungan *similarity* yang jauh lebih baik daripada *cosine*. Dengan hasil pengujian RMSE pada dataset pertama *Good Books* 0,919560, dan 0,894753. lalu pada dataset kedua *Book Crossing* 1,805998 dan 1,798955.
3. Pada pengujian yang sudah dilakukan *tuning* parameter nilai K cukup memberikan perubahan. Dengan hasil pada dataset *Good Books* sebelum di *tuning* pada *pearson* 0,894753 setelah di *tuning* 0,886626 terpaut 0,1. dan pada *Book Crossing* sebelum di *tuning* 1,798955 setelah di *tuning* 1,720001 terpaut 0,7.
4. Dari evaluasi yang telah dilakukan performa sistem rekomendasi sangat dipengaruhi nilai k yang diberikan. Hasil evaluasi terhadap 2 dataset menggunakan metode RMSE, Dataset *Good Books* paling baik pada

$k=11$  dengan nilai 0.886626. Sebaliknya untuk dataset *Book Crossing* paling baik pada  $k=92$  dengan nilai 1.720001.

5. Dari penelitian yang sudah dilakukan ada beberapa faktor yang menyebabkan tingginya hasil skor evaluasi RMSE yaitu
  1. Jumlah banyaknya data.
  2. Tingkat sparsity atau banyaknya nilai kosong.
  3. Pemilihan parameter nilai  $K$  juga mempengaruhi hasil evaluasi tetapi tidak memberikan hasil yang tidak terlalu jauh.

## 5.2 Saran

Untuk mengembangkan penelitian ini agar hasil rekomendasi lebih maksimal, berikut saran penulis oleh penulis:

1. Pengujian dapat dilakukan dengan pengujian similarity yang lain seperti *KNN With Means*, *Z Score* maupun lainnya.
2. Mencari cara lain untuk pembersihan data selain pengurangan dimensi karena banyak kecilnya data sangat mempengaruhi hasil sistem rekomendasi.
3. Menguji perbandingan dengan library lain seperti Scikit Learn maupun lainnya.
4. Untuk menseiasati peningkatan jumlah komputasi yang semakin berat sehingga menyebabkan *memory leak*, maka salah satu solusi yang dapat diterapkan adalah membuat antrian proses komputasi menggunakan teknologi queue seperti redis/rabbitMq.