

# 1. Pendahuluan

## 1.1 Latar belakang

Seiring dengan semakin majunya zaman dan semakin berkembangnya teknologi, jumlah dan jenis informasi yang ada dan dibutuhkan oleh manusia semakin banyak dan beragam. Yang menjadi permasalahan adalah bagaimana memilih informasi yang berguna dan relevan dengan yang dibutuhkan user. Pemilihan informasi ini mengalami kesulitan jika dilakukan secara manual karena kumpulan informasi yang sangat besar dan terus bertambah. Untuk mengatasi masalah tersebut, dibutuhkan sebuah *Information Retrieval System* atau sistem temu kembali informasi untuk membantu menemukan kembali informasi yang dibutuhkan user. *Information Retrieval System* adalah sistem yang digunakan untuk menemukan kembali dokumen-dokumen yang sesuai dengan kata kunci (*query*) yang diberikan user pada suatu koleksi dokumen. Dengan sistem ini, user mendapatkan informasi yang relevan/sesuai dengan kebutuhannya.

Menyusun ulang *query* (*query reformulation*) yang dimasukkan oleh user adalah hal yang sering dilakukan dalam *information retrieval*. Hal ini dilakukan untuk mengatasi ketidaksesuaian antara *query* yang dimasukkan oleh user dengan informasi yang ingin didapatkannya. Pada *Information Retrieval System*, terkadang dokumen yang di-*retrieve* dari *query* yang dimasukkan user tidak relevan. Hal ini bisa saja disebabkan salah ketik oleh user (*human error*), ataupun karena user tidak tahu *query* apa yang harus dimasukkan ke sistem. Oleh karena itu, diimplementasikan *relevance feedback* agar didapatkan *return hits* yang relevan dan sesuai dengan kebutuhan user.

*Relevance feedback* merupakan proses formulasi ulang *query* awal berdasarkan informasi umpan balik relevansi dari pengguna terhadap dokumen-dokumen hasil pencarian awal. Dengan menggunakan *relevance feedback*, pengguna dapat memilih informasi yang relevan terhadap kebutuhannya dan dari *feedback-feedback* user itulah yang digunakan untuk menentukan *query* baru.

Pada *Relevance Feedback* terdapat beberapa tipe yang bisa digunakan, antara lain: *Explicit Feedback*, *Implicit Feedback*, dan *Blind Feedback* (*pseudo*).

Dalam Tugas Akhir ini diimplementasikan salah satu tipe *Explicit Feedback* yakni dengan menggunakan algoritma *Rocchio*. Dengan menggunakan algoritma *Rocchio* ini diharapkan akan mendapatkan tingkat relevansi yang lebih baik dibandingkan jika tidak menggunakan *relevance feedback*.

## 1.2 Perumusan masalah

Dalam pengerjaan Tugas Akhir ini, ada beberapa rumusan masalah yang ingin diselesaikan, yaitu:

- a) Bagaimana mengimplementasikan *relevance feedback* menggunakan algoritma *Rocchio* pada *Information Retrieval System*?
- b) Bagaimana performansi dari *Information Retrieval System* yang dihasilkan dari algoritma *Rocchio*?

Adapun batasan masalah dalam Tugas Akhir ini, yaitu:

- a) Dokumen yang digunakan sudah ditentukan yaitu dokumen teks berbahasa Inggris standar yang telah memiliki *relevance judgemen* tyang diperoleh dari <ftp://ftp.cs.cornell.edu/pub/smart>
- b) *Query* yang diinputkan harus menggunakan bahasa Inggris.
- c) Performansi diukur berdasarkan *precision*, *recall* dan *Non Interpolated Average Precision* (NIAP).

### 1.3 Tujuan

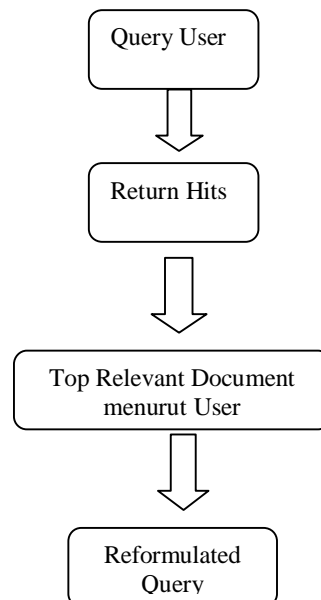
Tujuan yang ingin dicapai dari Tugas Akhir ini yaitu :

1. Membangun *Information Retrieval System* menggunakan algoritma *Rocchio* pada *Relevance Feedback*.
2. Menganalisis performansi sistem yang telah dibangun dengan mencari kondisi optimum bobot  $\alpha$ ,  $\beta$ , dan  $\gamma$  yang terdapat dalam *Algoritma Rocchio*.
3. Mencari jumlah term yang harus ditambahkan pada algoritma *Rocchio* agar mendapatkan hasil yang optimal.

### 1.4 Metodologi penyelesaian masalah

Dalam membuat Tugas Akhir ini, beberapa metode akan dilakukan yakni :

- a) Studi Literatur  
Studi literatur bertujuan untuk meningkatkan pemahaman tentang *Relevance Feedback* menggunakan algoritma *Rocchio*, dengan cara mencari referensi yang berkaitan dan kemudian mendalami materinya.
- b) Analisis dan Perancangan Perangkat Lunak  
Pada tahap ini akan dilakukan analisis terhadap perangkat lunak yang diperlukan dalam membangun implementasi dari aplikasi tersebut. Selain itu, melakukan pemilihan pemodelan perangkat lunak dan bahasa pemrograman saat melakukan perancangan.



- c) Implementasi dan *Prototyping*  
Setelah perancangan, dilakukan implementasi dan *Prototyping* untuk membangun aplikasi. Pada tahap ini akan dilakukan pengkodean.
- d) Testing  
Di tahap ini, dilakukan pengujian terhadap hasil implementasi dan *prototyping* untuk meminimalisir adanya *bug/error*. Jika *bug/error* ditemukan, maka dilakukan implementasi dan *prototyping* ulang.
- e) Analisis Hasil Implementasi  
Analisis hasil akhir aplikasi dilakukan dengan menghitung nilai *precision* dan *recall* yang dihasilkan dengan algoritma *Rocchio* pada aplikasi ini.
- f) Pembuatan Laporan Tugas Akhir.  
Pada tahap akhir, akan dilakukan pembuatan dokumentasi yang berupa laporan Tugas Akhir