1. Pendahuluan

1.1 Latar belakang

Dengan berkembangnya teknologi yang ada, maka berkembang pula kebutuhan akan informasi. Begitu banyak dokumen informasi yang tersedia. Sistem pencarian yang konvensional akan memberikan hasil berupa daftar panjang dokumen yang memaksa user untuk menelusuri satu-persatu agar menemukan informasi yang relevan. Hal ini pun masih diterapkan oleh sebagian *search engine* yang ada.[5] Akibatnya, dibutuhkan waktu yang lama untuk menemukan informasi-informasi dengan topik tertentu dari sekian banyak kumpulan dokumen. Untuk itu, perlu dilakukan *clustering* atau pengelompokan dokumen. *Clustering* dokumen merupakan cara yang tepat untuk mempermudah search engine dalam melakukan query terhadap kumpulan dokumen yang besar. Dokumen-dokumen yang memiliki kesamaan akan dikelompokkan sehingga membentuk topik-topik atau subtopik yang berbeda.

Algoritma-algoritma *clustering* dokumen yang sering dipelajari adalah algoritma-algoritma *batch clustering*, di mana keseluruhan dokumen diperlukan sejak awal dan *clustering* dilakukan dengan banyak iterasi terhadap dokumen tersebut. Namun, dengan adanya publikasi *online* di web yang semakin berkembang seperti sekarang ini, terjadi ledakan jumlah informasi yang bertambah setiap harinya. Sebagai contoh, dokumen teks dalam situs-situs pemberitaan selalu bertambah setiap saat. Kemudian muncul ide untuk melakukan *batch clustering* terhadap dokumen yang dikumpulkan secara periodik, baru selanjutnya menggabungkan klaster-klaster yang sudah terbentuk. Metode seperti ini tentu menimbulkan adanya waktu tunda hingga suatu dokumen masuk ke dalam klaster tertentu. Pada kenyataannya, waktu tunda tidak dapat diterima di beberapa bidang tertentu, misalnya bisnis jasa finansial, di mana keputusan perdagangan sangat bergantung pada berita, sehingga akses berita yang cepat pun sangat diperlukan.[7] Agar proses *clustering* dapat dilakukan segera setelah dokumen masuk, maka *clustering* perlu dilakukan secara *incremental*.

Terdapat beberapa algoritma *incremental clustering* yang populer. Salah satunya adalah algoritma Cobweb. Algoritma ini menggunakan konsep hierarkhi dalam proses *clustering*nya. Konsep hierarkhi dalam *clustering* dokumen memberikan dua keuntungan. Pertama, dengan mendeskripsikan hubungan antar grup dokumen akan mempercepat pencarian topik yang spesifik. Kedua, dengan memberikan hierarkhi topik yang berisi dokumen kepada user, user dapat mencari informasi yang diperlukan sesuai level spesifikasi yang diinginkannya.[7] Cobweb melakukan *clustering* data dengan membangun pohon klasifikasi di mana tiap node dari pohon tersebut menggambarkan cluster yang berisi objek-objek data. Dalam membangun pohon, Cobweb menggunakan *category utility* (CU) untuk mengevaluasi tree dan mendapatkan pengelompokan data yang paling tepat.

Pada Tugas Akhir ini akan diteliti bagaimana proses *clustering* dokumen, khususnya dokumen berbentuk teks hasil pencarian *search engine*, menggunakan algoritma Cobweb sebagai algoritma *clustering* incremental dan hierarkhis. Dengan sistem clustering Cobweb yang kokoh, diharapkan dapat menghasilkan solusi klaster yang berkualitas baik dan menyajikan data sesuai kebutuhan user.

1.2 Perumusan masalah

Untuk analisis *incremental document clustering* menggunakan algoritma Cobweb, masalah yang akan diselesaikan dalam Tugas Akhir ini adalah sebagai berikut.

1. Bagaimana efektivitas metode *incremental hierarchical clustering* yang diterapkan algoritma Cobweb melakukan *clustering* dokumen teks?
2. Bagaimana pengaruh jumlah dokumen ter-*retrieve* yang muncul pada *hitlist* dan query inputan user terhadap kualitas klaster yang dihasilkan?

1.3 Batasan Masalah

Adapun batasan-batasan yang diberikan dalam penyelesaian masalah Tugas Akhir ini adalah sebagai berikut.

1. Penelitian pada Tugas Akhir ini hanya terfokus pada *clustering* dokumen menggunakan algoritma COBWEB.
2. Dokumen yang dijadikan objek *clustering* berupa dokumen berbentuk teks yang diambil dari database.
3. Dokumen teks yang digunakan adalah dokumen berbahasa Inggris.
4. Proses *clustering* menggunakan dataset yang berupa dokumen hasil pencarian oleh *search engine*.
5. Hanya membahas metode *clustering* menggunakan algoritma Cobweb, sedangkan text preprocessing menjadi batasan masalah.
6. Metode *clustering* yang digunakan tidak menangani pelabelan.

1.4 Tujuan

Tujuan Tugas Akhir ini adalah sebagai berikut.

1. Membangun aplikasi yang dapat melakukan *clustering* dokumen teks dengan tepat untuk mendapatkan hierarki hitlist hasil pencarian search engine menggunakan algoritma COBWEB.
2. Menguji kualitas klaster yang dibangun dalam memenuhi permintaan informasi dengan mengukur nilai *intrinsic similarity* serta melihat pengaruh jumlah dokumen ter-*retrieve* dan *query* inputan user terhadap perubahan nilainya.

1.5 Metodologi penyelesaian masalah

Metode yang digunakan untuk menyelesaikan permasalahan-permasalahan Tugas Akhir ini terdiri dari langkah-langkah sebagai berikut.

1. Studi Literatur
   Mempelajari dan memahami salah satu algoritma *incremental document clustering* yaitu algoritma COBWEB melalui literatur berupa buku, makalah, atau jurnal dari berbagai media terutama internet.
2. Pengumpulan Dataset
   Pengumpulan dataset yang akan dijadikan sebagai data latih untuk merancang model *clustering* menggunakan algoritma Cobweb.
3. Pembangunan Aplikasi
   Pembangunan aplikasi yang meliputi:
   a. Perancangan Sistem

Menyiapkan dokumen yang akan digunakan untuk *clustering* menggunakan algoritma Cobweb. Selanjutnya dilakukan perancangan sistem yang nantinya dapat mengelompokkan hasil pencarian search engine ke dalam *cluster-cluster* yang tepat menggunakan algoritma Cobweb.

b. Implementasi
   Di tahap ini dilakukan implementasi algoritma Cobweb pada dokumen yang diperoleh, yang meliputi:
   - Menggunakan search engine untuk mendapatkan *retrieved documents* sebagai inputan bagi *clustering system* yang akan dibangun.
   - Memproses *retrieved documents* sehingga menjadi data terstruktur dengan nilai atribut numerik.
   - Melakukan *clustering* data menggunakan algoritma Cobweb dengan mengimplementasikannya ke dalam kode program.

c. Pengujian
   Sistem aplikasi yang sudah dibangun kemudian diuji untuk mengetahui apakah sistem sudah berjalan seperti yang diharapkan.
   1) Untuk menganalisis efektivitas aplikasi, pengujian dilakukan dengan menjalankan aplikasi search engine, kemudian dilakukan beberapa kali percobaan dengan melakukan pencarian berita terkait kata kunci tertentu.
   2) Untuk menganalisis pengaruh jumlah dokumen dan jumlah atribut terhadap efektivitas aplikasi, pencarian dilakukan beberapa kali dengan memilih query-query dengan kombinasi panjang query dan jumlah dokumen pada *hitlist* yang representatif.

4. Analisis Hasil
   Pada tahap ini dilakukan analisis terhadap hasil *clustering* yang diperoleh dari perancangan sistem menggunakan algoritma Cobweb. Output pengujian aplikasi dianalisis berdasarkan parameter-parameter yang telah ditentukan yaitu *intrinsic similarity* menggunakan H-*score*. Dari hasil analisis tersebut akan diambil kesimpulan mengenai cara kerja algoritma Cobweb dalam *clustering* dokumen teks.
5. Penyusunan Laporan
   Pada tahap ini, akan dilakukan penyusunan laporan akhir sekaligus dokumentasi dengan mengikuti kaidah penulisan yang benar dan sesuai dengan ketentuan yang ditetapkan oleh institusi.