1. Pendahuluan

1.1 Latar belakang masalah

Clustering adalah salah satu proses dari *data mining* yang merupakan bagian dari proses KDD (*Knowledge Discovery of Data*). Tujuan dari *clustering* adalah untuk mengelompokkan objek-objek data ke dalam *cluster-cluster* berdasarkan persamaan dan perbedaan yang dibawa oleh masing-masing atribut objek. *Cluster* yang baik akan memiliki persamaan (*similiarity*) intra *cluster* yang tinggi dan perbedaan (*dissimilarity*) antar *cluster* yang tinggi [3].

Ada beberapa pendekatan yang digunakan dalam mengembangkan metode *clustering*. Dua pendekatan utama adalah *clustering* dengan pendekatan partisi dan *clustering* dengan pendekatan hirarki. *Clustering* dengan pendekatan partisi atau sering disebut dengan *partition-based clustering* mengelompokkan data dengan memilah-milah data yang dianalisa ke dalam *cluster-cluster* yang ada. *Clustering* dengan pendekatan hirarki atau sering disebut dengan *hierarchical clustering* mengelompokkan data dengan membuat suatu hirarki berupa dendogram dimana data yang mirip akan ditempatkan pada hirarki yang berdekatan dan yang tidak pada hirarki yang berjauhan [3].

Algoritma *agglomerative hierarchical clustering* yang ada pada saat ini seperti CURE dan ROCK memiliki kekurangan utama dalam hal keputusan *merge* pasangan *cluster*. Dalam penggabungan *cluster* nya, algoritma CURE hanya mempertimbangkan *closeness* diantara representative point dari dua *cluster* dan mengabaikan informasi tentang *internal closeness* dari masing-masing *cluster*. Selain itu, CURE juga mengabaikan informasi *relative interconnectivity* dari dua *cluster* yang akan dilakukan *merge.* Sedangkan pada algoritma ROCK, hanya mempertimbangkan *interconnectivity* saja dan mengabaikan informasi tentang *internal interconnectivity* dari masing-masing *cluster* tersebut. Selain itu algoritma ROCK juga mengabaikan informasi *relative closeness* dari dua *cluster* yang akan dilakukan *merge*. Hal ini dapat berakibat pada kesalahan pengambilan keputusan dalam melakukan *merge* pasangan *cluster* [4]. Jika hal ini terjadi, maka kualitas *cluster* yang ditunjukkan pada nilai *cohesion* dan *cohesion/separation* yang dihasilkan oleh kedua algoritma tersebut akan kurang baik. Objek-objek yang berada pada *cluster* yang sama adalah objek-objek yang tidak memiliki kemiripan tinggi sehingga nilai *cohesion* yang dihasilkan adalah rendah. Oleh karena itu dengan semakin rendah nilai *cohesion* maka kualitas *cluster* yang ditunjukkan pada nilai *cohesion/separation* akan mengalami penurunan. Jika hal ini terjadi, maka kualitas *cluster* yang dihasilkan oleh kedua algoritma tersebut akan kurang bagus.

Salah satu pendekatan yang diterapkan untuk menangani masalah di atas adalah dengan menggunakan algoritma CHAMELEON. Algoritma CHAMELEON juga termasuk dalam *agglomerative hierarchical clustering.* Algoritma CHAMELEON sendiri mengelompokkan data dengan menggunakan dua buah fase yang berbeda yaitu fase *graph partitioning* dan fase *hierarchical agglomerative*. Proses *clustering* dimulai dengan memodelkan *similiarity* antar *cluster* dengan

membangun *graph* menggunakan pendekatan *K-nearest neighbor*. Fase *graph partitioning* dilakukan untuk menemukan kelompok-kelompok *cluster* yang saling terhubung. Kemudian fase *hierarchical agglomerative clustering* dilakukan untuk menggabungkan pasangan *subcluster* berdasarkan nilai *relative interconnectivity* dan *relative closeness* yang dimiliki oleh pasangan *subcluster* tersebut. Algoritma CHAMELEON hanya akan menggabungkan pasangan *subcluster* yang mempunyai nilai *relative interconnectivity* dan *relative closeness* yang nilainya diatas nilai *threshold RI* (*threshold relative interconnectivity*) dan *threshold RC* (*threshold relative closeness*) yang diinputkan oleh *user* [4].

Oleh karena itu, hipotesa awal yang diperoleh adalah hasil klasterisasi yang dibangun dengan algoritma CHAMELEON dapat menghasilkan kualitas *cluster* yang baik dimana *cluster* yang dihasilkan memiliki kesamaan karakteristik yang tinggi dalam satu *cluster* dan memiliki perbedaan yang tinggi antar *cluster*nya.

1.2 Perumusan masalah

Berdasarkan latar belakang tersebut, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana pengaruh perubahan nilai *k* (jumlah tetangga terdekat pada *k nearest neighbor graph*), *threshold RI, threshold RC, nlevel* yang ditetapkan oleh user terhadap akurasi hasil *cluster* pada algoritma CHAMELEON.
2. Bagaimana kualitas hasil *cluster* yang dihasilkan algoritma CHAMELEON dengan melihat hasil evaluasi *cluster* menggunakan perhitungan *cohesion* dan *separation*.

Adapun batasan masalah tugas akhir ini adalah sebagai berikut :

1. Dataset yang digunakan adalah dataset yang ada pada database UCI Machine Learning Repository yang mempunyai atribut kategoris.
2. Evaluasi kualitas hasil *cluster* dilakukan dengan mengukur nilai *cohesion* (kemiripan objek intra *cluster)* dan *separation* (ketidakmiripan objek antar *cluster)*).

1.3 Tujuan

Tujuan dari penelitian tugas akhir ini adalah:

1. Menganalisis algoritma CHAMELEON terhadap perubahan parameter nilai *k* (jumlah tetangga terdekat pada *k nearest neighbor graph*), *threshold RI, threshold RC, nlevel* terkait dengan akurasi hasil *cluster*.
2. Menganalisis kualitas *cluster* yang dihasilkan dengan melihat nilai *cohesion* dan *separation*.

1.4 Metodologi penyelesaian masalah

Metodologi yang digunakan dan langkah-langkah dalam penyelesaian masalah yang telah dirumuskan di atas adalah:

1. Studi Literatur.

a. Pencarian referensi, mencari referensi dan sumber-sumber lain yang layak yang berhubungan dengan *data mining*, *clustering*, algoritma CHAMELEON
b. Pendalaman materi, mempelajari dan memahami materi yang berhubungan dengan tugas akhir.

2. Pengumpulan data
   Mencari data kategoris untuk keperluan analisis algoritma CHAMELEON
3. Implementasi perangkat lunak
   a. Analisis dan design perangkat lunak
      Melakukan analisis dan desain perangkat lunak, mengenai kebutuhan sistem serta fungsionalitas – fungsionalitas yang dibutuhkan dalam sistem.
   b. Implementasi (*coding*)
      Pembuatan program berdasarkan analisis dan desain program yang telah ditentukan pada tahap sebelumnya.
   c. Pengujian
      Menguji aplikasi yang telah dibuat.
4. Analisis hasil
   Menganalisis hasil *cluster* yang terbentuk terkait dengan pengubahan parameter *k* (pada *k nearest neighbor*), *threshold RI, threshold RC, nlevel*
5. Pembuatan laporan Tugas Akhir
   Mengambil kesimpulan dari hasil analisis dan pembuatan laporan tugas akhir.