

ABSTRAK

Penggalian kaidah asosiasi (*mining association rules*) merupakan salah satu proses *data mining* untuk menemukan pola dan aturan (*rule*) dari sekumpulan data yang besar. Pola-pola ini merupakan kumpulan item (*itemset*) yang sering muncul secara bersamaan (*frequent itemset*) dalam transaksi pada basis data. Proses pencarian *frequent itemset* membutuhkan waktu yang sangat lama, oleh karena itu diperlukan suatu algoritma yang bisa mengefisiensi waktu yang dibutuhkan.

Algoritma yang paling populer saat ini adalah algoritma apriori yang menggunakan *support base pruning* (membuang ruang pencarian dengan batasan nilai *support*). Algoritma ini memiliki kelemahan ketika kardinalitas pada *longest frequent itemset* berupa k , membutuhkan sebanyak k pembacaan basis data dan memiliki sifat *computation-intensive* dalam membangkitkan kandidat *itemset* dan penghitungan nilai *support*, khususnya untuk aplikasi yang memiliki nilai *support* yang sangat rendah dan atau item yang sangat banyak.

Algoritma *Cut Both Ways (CBW)* menggunakan gabungan beberapa teknik dan menggunakan *cutting level* (α) untuk membagi ruang pencarian menjadi dua bagian. Strategi *top-down* untuk menemukan *frequent itemset* yang berada dibawah *cutting level* dikombinasikan dengan strategi pencarian *breadth first search* dan *horizontal counting* untuk penghitungan nilai *support*. Sedangkan *bottom-up* untuk menemukan *frequent itemset* yang berada diatas *cutting level* dikombinasikan dengan *depth first search* dan *vertical intersection*. Nilai *cutting level* merupakan nilai rata-rata dari kardinalitas *frequent itemset*.

Pada tugas akhir ini akan mengimplementasikan proses pencarian *frequent itemset* dengan menggunakan algoritma Apriori dan CBW. Kemudian membandingkan kinerjanya dengan menggunakan beberapa parameter nilai *support*.

Kata Kunci : *mining association rules, itemset, frequent itemset, support, support base pruning, longest frequent itemset, computation-intensive, cutting level, top-down, bottom-up, breadth first search, dept first search, vertical intersection.*