Review Paper FP-Growth Algorithm

Nindyasari Dewi Utari 5113100039

Kinasih Nur Azizah 5113100178

Admiral Budi Arviansyah W 5113100189

*Frequent Patterns Tree* (FP-*Tree*) adalah struktur pohon atau pola yang mengandung informasi penting yang sering muncul. Fp-*Tree* yang dibahas pada paper ini adalah FP-*Tree* yang sekarang sedang dikembangkan. Selain itu, metode *frequent pattern* yang dibahas pada paper ini adalah FP-*growth*.

Efisiensi pengolahan data dicapai dengan 3 teknik. Pertama, dengan mengkompres database yang besar menjadi sangat padat dan kecil, sehingga mengurangi biaya dan perulangan *scannig*  database. Kedua, FP-*Tree* mengadopsi metode pola pertumbuhan fragmen untuk mengurangi biaya dari sejumlah calon dataset besar. Ketiga, *partitioning-based* metode *divide-and-conquer* digunakan untuk menguraikan tugas dari pengolahan data menjadi satu set tugas yang lebih kecil sehingga memperkecil ruang pencarian.

Struktur data yang tersusun rapat dapat dirancang dengan 4 cara, yang pertama adalah hanya *item frequent* yang akan memulai peran pada *pattern frequent*, hal itu perlu untuk menampilkan 1 scan dari DB untuk mengidentifikasi dataset dari *item frequent* tersebut. Kedua, jika kita akan menyimpan dataset dari item *frequent* dari beberapa transaksi pada struktur yang tersusun rapat, hal itu mungkin akan menghindari *scanning* berulang pada DB. Ketiga, banyak transaksi yang membagikan *dataset frequent* yang identic. Dan yang keempat, jika 2 transaksi membagi prefix umum, jika item yang sering diurutkan frekuensinya menurun, maka string yang lebih baiklah yang akan dibagikan.

*Output* dari metode FP-*growth* adalah kita dapat membuat struktur FP-*Tree*  yang sesuai dengan berbagai keunggulan yang ditawarkannya. Keunggulan dari menggunakan algoritma ini adalah (1) FP-*growth* dapat membuat sturktur FP-*Tree* yang lengkap dan biasanya lebih kecil dari database aslinya sehingga mengurangi biaya scanning database pada proses pengolahan data, (2) FP-*growth* menggunakan pola metode *growth* yang menghindari biaya generasi kandidat dan diuji dengan meng-*concat* 1-itemset secara berkala, (3) FP-*growth* menggunakan partitioning-based metode divide-and-conquer yang berdampakkan mengurangi sebagian besar ukuran basis pola kodisional berikutnya dan Fp-*Tree* kondisional.

Jadi, kesimpulan dari paper ini adalah mengajukan sebuah stuktur data baru untuk kompresi penyimpanan, informasi krusial mengenai *frequent* pattern dan mengembangkan metode *pattern growth* yaitu FP-*growth* untuk efisiensi pengolahan *frequent pattern* di sebuah database yang besar. Jika dibandingkan dengan metode *Apriori* dan *Tree Projection*, FP-*Tree* termasuk algoritma yang lebih cepat daripada kedua algoritma tersebut.