# LAPORAN TUGAS BESAR AKHIR SEMESTER

**Struktur Data dan Algoritma**

#### IMPLEMENTASI Asociation Rules/Market Basis Analysis



**Disusun oleh**

**Azzam Badrus Zaman – 1915240**

**Wafi Khoerun Nashirin – 1915240**

**Program Studi D-IV Teknik Informatika Departemen Teknik Komputer dan Informatika Politeknik Negeri Bandung**

**2019**

**1 DESKRIPSI APLIKASI**

* 1. **Teori Kompresi**

Market basket analysis adalah suatu analisa atas perilaku konsumen secara spesifik dari suatu golongan / kelompok tertentu. Sumber data dari market basket analysis antara lain dapat bersumber dari transaksi kartu kredit, kartu lotere, kupon diskon, panggilan keluhan pelanggan. Market basket analysis umumnya dimanfaatkan sebagai titik awal pencarian pengetahuan dari suatu transaksi data ketika kita tidak mengetahui pola spesifik apa yang kita cari. Kebutuhan market basket analysis berawal dari keakuratan dan manfaat yang dihasilkannya dalam wujud aturan assosiasi (association rules). Yang dimaksud dengan association rules adalah pola-pola keterkaitan data dalam basis data.

Proses market basket analysis dimulai dengan transaksi yang terdiri dari satu/lebih penawaran produk/jasa dan beberapa informasi dasar suatu transaksi. Hasil dari market basket analysis adalah berwujud aturan assosiasi (association rules).

**2.1 Abstraksi**

Abstraksi Kombinasi Item

Diagrams kombinasi level 0 : mamiliki 1 itemset

Level 1 : memiliki 2 itemset

Level 2 : memiliki 3 itemset

Level 3 : memiliki 4 itemset

Rumus kombinasi 2^n-1

Tree dan list yang akan dibuat juga berukuran 2^n-1

|  |  |
| --- | --- |
| Item | Support |
| Susu |  |
| Roti |  |
| Keju |  |
| Coklat |  |
| Susu,keju |  |
| Susu,roti |  |
| Susu,coklat |  |
| Roti,keju |  |
| Roti,coklat |  |
| Keju,coklat |  |
| Susu,roti,keju |  |
| Susu,roti,coklat |  |
| Susu,coklat,keju |  |
| Coklat,keju,susu |  |
| susu,coklat,keju,roti |  |

Tahap kerja:

1. baca file (bertipe string) -> **pengerjaan terakhir setelah code berjalan dengan baik**

* file baris pertama akan berisi nama barang apa saja yang tersedia, baris selanjutnya adalah rincian setiap transaksi (baris 2 = T1, baris 3 = T2 dan seterusnya)
* nama barang yang dijual akan di tampbung di array of array of char (aray of string)

gambaran:

typedef struct {

char \* namaBarang;

} list;

//deklarasi

list barang[jml barang]

Sehingga barang[1] adalah barang pertama, barang[2] adalah nama barang ke2 dan seterusnaya.

* Transaksi juga sama seperti barang

1. Membuat kemungkinan kombinasi transaksi yang terjadi sesuai jumlah barang.

* Semua nama barang cukup di wakilkan dengan angka, angka akan mewakilkan indeks dari list barang agar memudahkan dalam membuat tree kombinasi.sehingga cukup generat(jml barang)
* Setiap node berisi: int info, queue kombinasi, dan int support (rencananya)

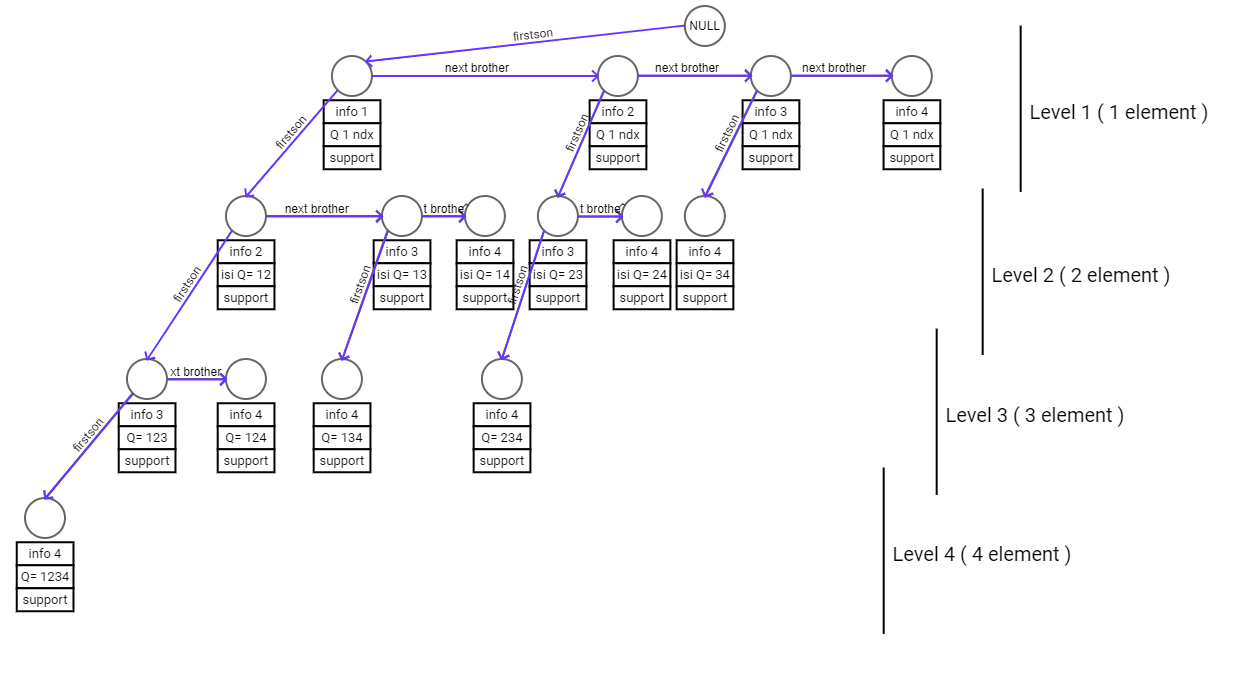
Typedef struct node {

Char info; // berisi char/int sebagai identitas/pengenal node

Queue kombinasi; //menyimpanurutan kombinasi

Int support;

} node;

* Gambaran tree:

1. Menghitung support

* Menghitung support dilakukan secara rekursif yaitu:
  + Search rekursif, ke kombinasi maks transaksi.

Misal T1 (124) maka search ke node yang bernilai 124.

* + Setelah ketemu, support node tersebut +1;
  + Rekursif FILO, maka semua node yang terlewati saat mencari node maks tersebut, bertambah satu supportnya. Alias semua kombinasi yang terjadi dalam T1(124) bertambah supportnya

1. Memindahkan ke string/ list

* String akan bertipe data array of array:

Gambaran:

Typedef struct {

Char kombinasi[lvl]; //kombinasi yang dikonversi Q->string

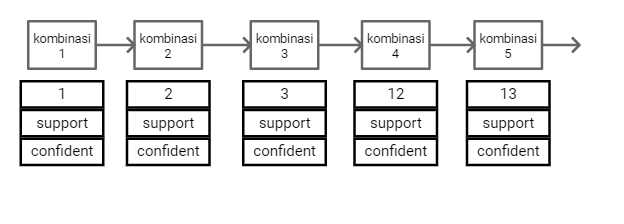
Int support;

} listKS; //list kombin support

//deklarasi

listKS kombinasi[2^n-1];

* untuk tipe data bisa jadi menggunakan linklist, jika diperlukan delete, yaitu kalau kombinasi yang supportnya dibawah standar dihapus.
* Gambaran:



* Proses Q->string menggunakan level order sehingga ukuran string yang dibuat untuk menampung kombinasi dari queue jumlah nya sesuai, tidak ada yang berlebih.

Gambaran:

* + Int k //level
  + Char kombin[k] //jumlah indekx string sesuai element
  + Selama i kurang dari k

Kombin[i]= dequeue //dari queu node yang bersangkutan.

1. Setelah terhitung supportnya,tentukan Min Support