

Implementasi Algoritma FP-GROWTH untuk Sistem Rekomendasi Buku di Perpustakaan UKDW

Rama Novta Miraldi
r.miraldi@gmail.com

Antonius Rachmat
anton@ti.ukdw.ac.id

Budi Susanto
budsus@ti.ukdw.ac.id

Abstract

The number of collections in a library will be increase along with the growth of the library. The Library of Universitas Kristen Duta Wacana (UKDW) also has a large number of books collection in order to provide literature and resources service to students. When the number of collection is very large, we need a retrieval system that will help anyone to find the needed resources. This system has a recommendation feature to provide related books or resources. This feature will help users to find similar literature/resources or related topic with what they need. This paper will describe about the main design and implementation of the book recommendation based on FP-Growth algorithm. Based on the experiments, this system can provide 60.78% of accuracy for related books recommendation

Kata Kunci : *perpustakaan, buku, data mining, fp-growth, fp-tree.*

1. Pendahuluan

Perpustakaan adalah sebuah ruangan, bagian sebuah gedung, ataupun gedung itu sendiri yang digunakan untuk menyimpan buku dan terbitan lainnya menurut tata susunan tertentu untuk digunakan pembaca, bukan untuk dijual (Sulistyo, 1991). Seiring perkembangan zaman koleksi buku suatu perpustakaan bertambah banyak, tidak terkecuali perpustakaan Universitas Kristen Duta Wacana (UKDW). Bertambahnya koleksi buku akan menyulitkan pengunjung perpustakaan untuk mencari buku yang diinginkan. Oleh karena itu dibutuhkan alat bantu berupa sistem pencarian untuk mempermudah pengunjung perpustakaan mencari buku.

Peminjam buku pada perpustakaan membutuhkan alternatif informasi buku lain ketika buku yang hendak dipinjam sedang dipinjam. Referensi peminjam buku dapat menjadi lebih luas jika sistem pencarian ditambahkan fitur rekomendasi buku. Fitur rekomendasi buku ini dapat dilakukan dengan cara mengimplementasikan salah satu algoritma *Association Analysis* yaitu Algoritma *Frequent Pattern Growth (FP-Growth)*. Algoritma *FP-Growth* ini adalah algoritma yang membutuhkan pelatihan sistem sebelum dapat dipakai untuk memberikan rekomendasi. Dengan menggunakan data transaksi sebagai data untuk pelatihan, sistem dapat menghasilkan rekomendasi.

Oleh karena itu dengan mengimplementasikan algoritma *FP-Growth* untuk fitur rekomendasi pada perpustakaan Universitas Kristen Duta Wacana dapat referensi yang akurat terkait dengan buku yang akan dipinjam. Selain itu juga dapat membantu perpustakaan Universitas Kristen Duta Wacana untuk mengembangkan sistem rekomendasi untuk lebih baik dikemudian hari.

2. Teori Pendukung

2.1 Association Rule

Association Rule merupakan tugas *data mining* yang mendasar (L. Bing, 2007). Menurut “*Data Preparation for Data Mining*” dalam *Applied Artificial Intelligence* (Zhang, Zhang, & Yang, 2003) *Association Rule* merupakan salah satu metode yang bertujuan untuk mencari pola yang sering muncul pada banyak transaksi, dimana setiap transaksi terdiri dari beberapa *item*.

Salah satu pengaplikasian dari *Association Rule* adalah *Market Basket Analysis*, dimana bertujuan untuk menemukan bagaimana *item* yang dibeli oleh pelanggan dalam supermarket atau toko saling berhubungan. Pencarian *Association Rules* dilakukan melalui dua tahap yaitu pencarian *frequent itemset* dan penyusunan *rules*. Penting tidaknya suatu *Association Rules* dapat diketahui dengan dua parameter, yaitu *support* (nilai penunjang) dan *confidence* (nilai kepastian). *Support* adalah ukuran yang menunjukkan tingkat dominasi *itemset* dari keseluruhan transaksi. Persamaan 1 untuk menentukan nilai suatu *support*

$$\text{Support} = \frac{(X \cup Y)\text{count}}{n} \quad [1]$$

dimana $(X \cup Y)$ count adalah jumlah transaksi yang mengandung x dan y , dan n adalah total transaksi. Parameter lain nya adalah *confidence* yaitu nilai ukuran seberapa besar valid tidaknya suatu *Association Rules*. Sebuah *Association Rules* dengan *confidence* sama atau lebih besar dari minimum *confidence* dapat dikatakan sebagai *valid association rule*. Persamaan 2 untuk menentukan nilai *confidence*, dimana X count adalah jumlah kemunculan *item X* pada seluruh transaksi.

$$\text{Confidence} = \frac{(X \cup Y)\text{count}}{X.\text{count}} \quad [2]$$

2.2 Lift

Korelasi pada suatu *rule* tidak hanya diukur pada nilai *support* dan *confidence* tetapi juga oleh korelasi antara kedua *itemsets*. Ada banyak ukuran korelasi yang dapat dipilih, salah satu nya adalah *lift*. *Lift* adalah salah satu contoh ukuran korelasi sederhana. Nilai *lift* antara A dan B dapat diukur dengan menggunakan komputasi sebagai berikut (Tan, Steinbach, & Kumar, 2004)

$$\text{Lift}(A,B) = \frac{P(A \cup B)}{P(A).P(B)} \quad [3]$$

dimana nilai $P(A \cup B)$ adalah nilai *confidence* dari *itemset A* dengan B . Dan $P(A)$ adalah jumlah transaksi yang mengandung A . Jika hasil nilai dari perhitungan pada rumus [2.3] kurang dari 1 maka korelasi dari A adalah *negative correlated* dengan *itemset B*, yang berarti bahwa tidak adanya keterkaitan antar satu sama lain. Dan jika hasil yang didapat adalah lebih dari 1, maka antara A dan B adalah *positively correlated*. Sedangkan bila hasil yang didapat adalah sama dengan 1 maka A dan B adalah *independent*.

2.3 FP-Tree

Frequent Patern Tree (FP-Tree) adalah representasi pemasukan data yang di padatkan (Kumar, 2004). *FP-Tree* dibentuk dengan membaca kumpulan data pada suatu transaksi dalam suatu waktu dan memetakan transaksi tersebut ke dalam lintasan *FP-Tree*. Transaksi yang berbeda dapat memiliki *items* yang sama, sehingga memungkinkan lintasannya saling menimpa. Semakin banyak lintasan yang menimpa satu sama lain, dengan menggunakan struktur *FP-Tree* maka proses pemadatan akan semakin baik.

Dalam “*Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach*” (Han, Pei, Yin, & Mao, 2004) menjelaskan misal $I = \{a_1, a_2, \dots\}$

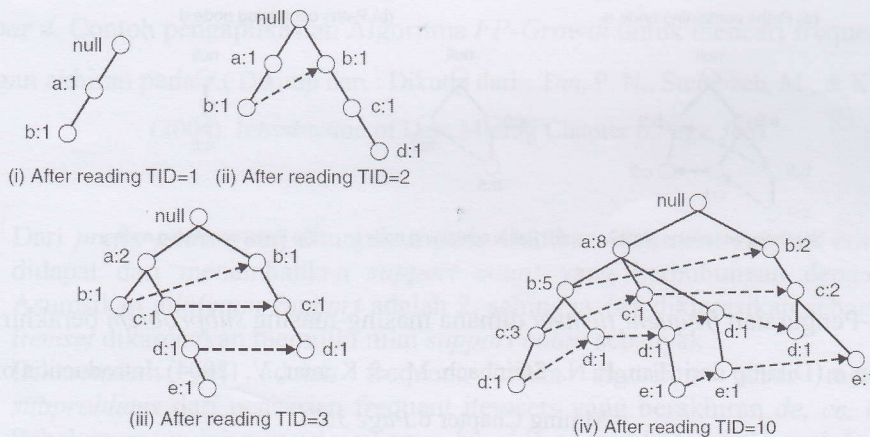
\dots, a_n adalah kumpulan dari item, dan database transaksi DB = $\{T_1, T_2, \dots, T_n\}$, dimana T_i ($i \in [1..n]$) adalah sekumpulan transaksi yang mengandung item di I. Sedangkan *support* adalah penghitung (*counter*) frekuensi kemunculan transaksi yang mengandung suatu pola. Suatu pola dikatakan sering muncul (*frequent pattern*) apabila *support* dari pola tersebut tidak kurang dari suatu konstanta minimum *support* (batas ambang minimum *support*) yang telah di definisikan sebelumnya. Permasalahan mencari pola frequent dengan batas ambang minimum *support count*, inilah yang dicoba untuk dipecahkan oleh FP-Growth dengan bantuan struktur FP-tree.

Berikut salah satu contoh kasus dengan menggunakan 10 transaksi dengan minimum support count = 2

Tabel 1.
Tabel Data Transaksi Mentah
(Diambil dari : Tan, P. N., Steinbach, M., & Kumar, V. (2004). Introduction of Data Mining Chapter 6. Page 364)

| TID | Items |
|-----|-----------|
| 1 | {a,b} |
| 2 | {b,c,d} |
| 3 | {a,c,d,e} |
| 4 | {a,d,e} |
| 5 | {a,b,c} |
| 6 | {a,b,c,d} |
| 7 | {a} |
| 8 | {a,b,c} |
| 9 | {a,b,d} |
| 10 | {b,c,e} |

Gambar di bawah ini memberikan ilustrasi mengenai langkah pembentukan FP-tree.



Gambar 1. Pembentukan FP-tree

(Diambil Dari: Tan, P. N., Steinbach, M., & Kumar, V. (2004). Introduction of Data Mining Chapter 6. Page 364)

2.4 FP-Growth

FP-Growth adalah algoritma pencarian frequent itemsets yang didapat dari FP-tree dengan menjelajahi tree dari bawah menuju ke atas (Tan, Steinbach, & Kumar, 2004) Algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. Sehingga kekurangan dari algoritma Apriori diperbaiki di algoritma FP-Growth. Algoritma ini menentukan frequent itemset yang berakhiran suffix tertentu dengan menggunakan metode divide and conquer untuk memecah problem menjadi subproblem yang lebih kecil (Han, Kamber, & Pei, Data Mining : Concepts and Techniques, 2011). FP-Growth menggunakan

konsep pembangunan *tree* dalam pencarian *frequent itemset*, tidak menggunakan *generate candidate* seperti Algoritma *Apriori*. Hal ini lah yang menyebabkan Algoritma *FP-Growth* lebih cepat dari Algoritma *Apriori*.

Adapun *pseudocode* dari algoritma *FP-Growth* sebagai berikut

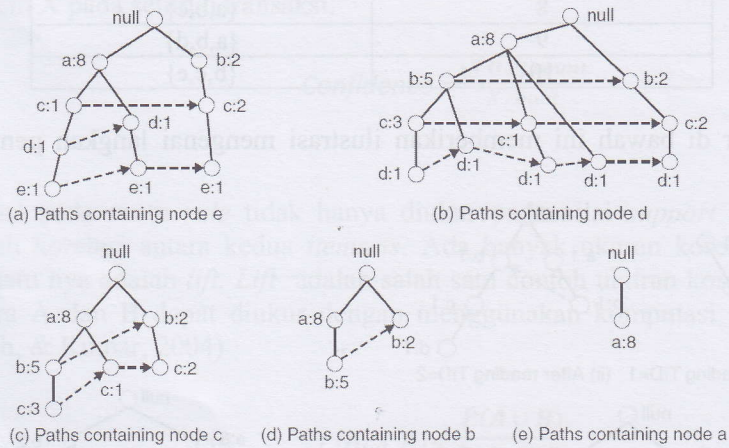
```

procedure FP_growth(Tree,  $\alpha$ )
(1)  if Tree contains a single path P then
(2)    for each combination (denoted as  $\beta$ ) of the nodes in the path P
(3)      generate pattern  $\beta \cup \alpha$  with support_count = minimum support count of nodes in  $\beta$ ;
(4)  else for each  $a_i$  in the header of Tree {
(5)    generate pattern  $\beta = a_i \cup \alpha$  with support_count =  $a_i$ .support_count;
(6)    construct  $\beta$ 's conditional pattern base and then  $\beta$ 's conditional FP-tree Tree $\beta$ ;
(7)    if Tree $\beta$   $\neq \emptyset$  then
(8)      call FP_growth(Tree $\beta$ ,  $\beta$ ); }
    
```

Gambar 2. FP-Growth.

(Dikutip dari: Han, J., Kamber, M., Pei, J. (2011). *Data Mining: Concepts and techniques*)

Untuk menemukan *Frequent itemset* dari tabel 1, maka perlu di tentukan terlebih dahulu lintasan yang berakhir dengan *support count* terkecil, yaitu *e* yang di ikuti dengan *d*, *c*, *b*, dan di akhiri *a*. Proses pembentukan masing-masing node dapat dilihat pada gambar berikut:



Gambar 3. Penguraian *frequent itemset* dimana masing-masing *subproblem* berakhir di *e*, *d*, *c*, *b* dan *a*. (Dikutip dari: Tan, P. N., Steinbach, M., & Kumar, V. (2004). *Introduction of Data Mining Chapter 6. Page 367*)

Setelah memeriksa *frequent itemset* untuk beberapa akhiran (*suffix*), maka didapat hasil yang dirangkum dalam tabel berikut:

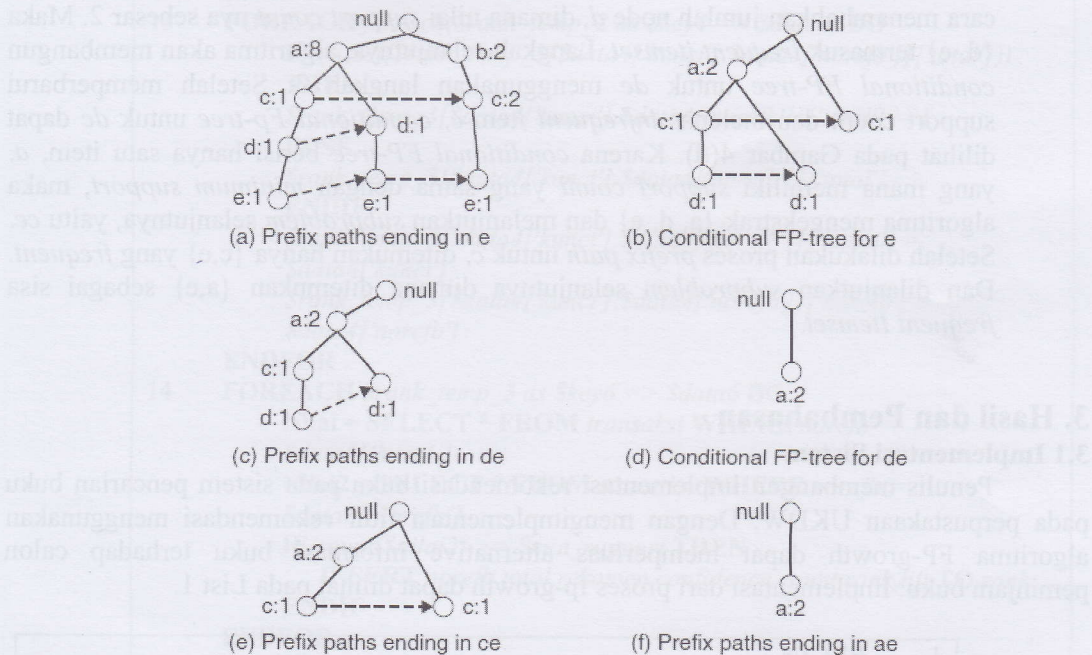
Tabel 3.

Daftar *Frequent Itemsets* diurutkan berdasarkan hubungan akhiran (Dikutip dari : Tan, P. N., Steinbach, M., & Kumar, V. (2004). *Introduction of Data Mining Chapter 6. Page 367*)

| Suffix | Frequent Itemset |
|--------|---|
| e | {e}, {d,e}, {a,d,e}, {c,e}, {a,e} |
| d | {d}, {c,d}, {b,c,d}, {a,c,d}, {b,d}, {a,b,d}, {a,d} |
| c | {c}, {b,c}, {a,b,c}, {a,c} |
| b | {b}, {a,b} |
| a | {a} |

Berikut contoh yang lebih detail tentang langkah-langkah menemukan frequent itemset, dimana itemsets yang dicari adalah itemsets berakhiran e , berdasarkan *Introduction of Data Mining*.

- Langkah pertama adalah mengumpulkan semua lintasan yang mengandung *node e*. *Initial paths* ini disebut *Prefix paths*. Berikut gambar dari lintasan yang mengandung *node e* beserta tahapan-tahapannya



Gambar 4. Contoh pengaplikasian Algoritma *FP-Growth* untuk mencari frequent itemset dengan akhiran pada e . (Dikutip dari : Dikutip dari : Tan, P. N., Steinbach, M., & Kumar, V. (2004). *Introduction of Data Mining Chapter 6*.Page 368)

- Dari *prefix paths* yang ditunjukkan pada Gambar 4(a), nilai *support count* untuk e didapat dari menambahkan *support count* yang berhubungan dengan *node e*. Asumsikan *minimum support* adalah 2, sehingga $\{e\}$ diklarasikan sebagai *frequent itemset* dikarenakan memiliki nilai *support count* sebanyak 3.
- Dikarenakan $\{e\}$ adalah frequent, maka algoritma harus menyelesaikan *subproblems* dari pencarian frequent itemsets yang berakhiran de , ce , be , dan ae . Sebelum memulai menyelesaikan *subproblems* tersebut, *prefix path* harus diubah terlebih dahulu menjadi *conditional FP-tree*. *Conditional FP-tree* dapat diperoleh dengan langkah berikut :
 - Support count* beserta *prefix path* harus diperbaharui terlebih dahulu karena beberapa transaksi tidak mengandung item e . Sebagai contoh pada gambar 2.7(a), $null \rightarrow b: 2 \rightarrow c: 2 \rightarrow e: 1$, pada $\{b, c\}$ mengandung transaksi yang tidak memiliki e .
 - Prefix paths* dipotong , dengan cara melepas *node e*. *Node* ini dapat di lepas karena *support counts* berhubungan dengan *prefix paths* telah diperbaharui untuk mencerminkan hanya transaksi yang mengandung e dan *subproblems* dari pencarian *frequent itemsets* yang berakhiran de , ce , be , dan ae tidak lagi membutuhkan informasi *node e*.
 - Setelah pembaharuan *support count* bersama dengan *prefix path*, beberapa item tidak lagi frequent. Sebagai contoh, *node b* hanya muncul sekali dan

memiliki nilai *support count* 1, yang berarti hanya ada 1 transaksi yang mengandung *b* dan *e*. Maka *b* dapat diabaikan dari *subsequent analysis* karena semua itemset yang berakhiran *be* pasti tidak *infrequent*. Gambar 2.7(b) menunjukkan Conditional FP-tree untuk *e*, dimana nilai *b* dan *e* telah dieliminasi.

4. FP-Growth menggunakan *conditional FP-tree* dari *e* untuk menyelesaikan subproblem dari pencarian *frequent itemset* yang berakhiran *de*, *ce*, dan *ae*. Untuk menentukan {*d*, *e*} termasuk *frequent itemset* atau tidak, dapat diketahui dengan cara menambahkan jumlah node *d*, dimana nilai *support count* nya sebesar 2. Maka {*d*, *e*} termasuk *frequent itemset*. Langkah selanjutnya, algoritma akan membangun *conditional FP-tree* untuk *de* menggunakan langkah 3. Setelah memperbarui *support count* dan melepas *infrequent* item *c*, *conditional Fp-tree* untuk *de* dapat dilihat pada Gambar 4(d). Karena *conditional FP-tree* berisi hanya satu item, *a*, yang mana memiliki *support count* yang sama dengan *minimum support*, maka algoritma mengekstrak {*a*, *d*, *e*} dan melanjutkan *subproblem* selanjutnya, yaitu *ce*. Setelah dilakukan proses *prefix path* untuk *c*, ditemukan hanya {*c,e*} yang *frequent*. Dan dilanjutkan *subproblem* selanjutnya dimana ditemukan {*a,e*} sebagai sisa *frequent Itemset*.

3. Hasil dan Pembahasan

3.1 Implementasi Sistem

Penulis membangun implementasi rekomendasi buku pada sistem pencarian buku pada perpustakaan UKDW. Dengan mengimplementasi fitur rekomendasi menggunakan algoritma FP-growth dapat memperluas alternative informasi buku terhadap calon peminjam buku. Implementasi dari proses fp-growth dapat dilihat pada List 1.

```

1  BEGIN
2  SET min support
3  Truncate Table Rank
4  utik = SELECT norefb, count (norefb) as total FROM transaksi
5  FOREACH utik as row1 DO
      IF total ≥ min support THEN
          Insert norefb, total to rank
      ENDIF
    ENDFOR
6  query_test = SELECT nim,tgl_pinjam,petugas,norefb FROM `transaksi`
      ORDER BY `transaksi`.`nim` ASC'
7  last_query = query_test->result_array()
8  FOREACH last_query as key => data DO
      transaksi[data['nim'].date('dmy',strtotime(data['tgl_pinjam']))].data['
      petugas']][$i] = $data;
      $i++;
9  Total_trans = count(transaksi)
10 rank = SELECT norefb, total FROM rank
11 Truncate table muncul and rank
12 FOREACH rank as row2 =>data2 DO
      Nampung = SELECT nim,petugas,norefb,tgl_pinjam
      FROM transaksi WHERE norefb = rank → norefb, petugas =
      nampung → petugas, tgl_pinjam = nampung → tgl_pinjam
      Rank_temp[data→norefb] → total = $data → total
      FOREACH nampung as row3 => data2 DO
          Tgl = date('dmy',strtotime($data2->tgl_pinjam))
          petugas = data2→petugas
          IF (!empty($transaksi[$data2->nim.$tgl.$petugas])) THEN
              rank_temp[data→norefb]→hasil_utik[$data2→nim.$tgl.
              $petugas] = $transaksi[$data2->nim.$tgl.$petugas]
          FOREACH $transaksi[$data2→nim.$tgl.$petugas] as

```

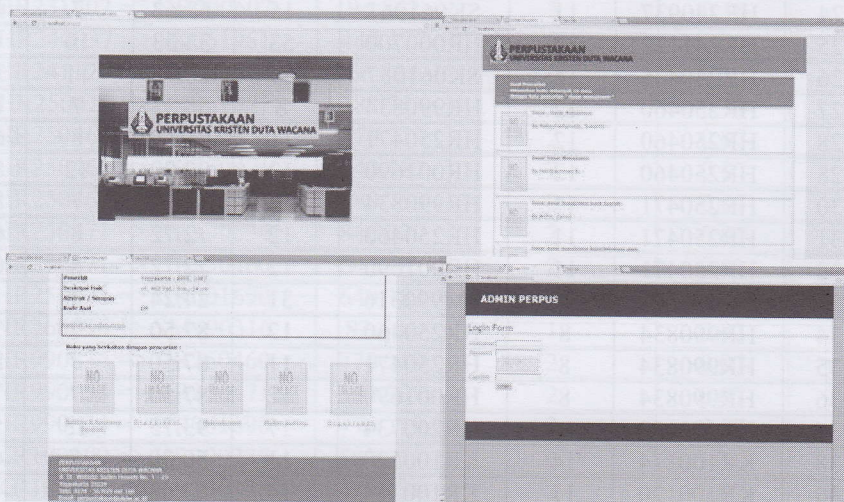
```

row3 => $data3) DO
    IF($data3['norefb'] != $data->norefb) THEN
        $rank_temp_2[$j]['kunci']=$data->norefb;
        $rank_temp_2[$j]['norefb'] = $data3['norefb'];
    ENDIF
ENDFOR
ENDIF
ENDFOR
ENDIF
13 FOREACH foreach($rank_temp_2 as $key4 => $data4) DO
    IF (!isset($rank_temp_3[$data4['kunci'].$data4['norefb']]['total']))
    THEN
        $rank_temp_3[$data4['kunci'].$data4['norefb']]['total'] = 1
    ELSE
        $rank_temp_3[$data4['kunci'].$data4['norefb']]['total']++
    ENDIF
    $rank_temp_3[$data4['kunci'].$data4['norefb']]['kunci'] =
    $data4['kunci'];
    $rank_temp_3[$data4['kunci'].$data4['norefb']]['norefb'] =
    $data4['norefb'];
ENDFOR
14 FOREACH $rank_temp_3 as $key6 => $data6 DO
    Nilai = SELECT * FROM transaksi WHERE norefb =
    $data6['kunci']
    Nilai2 = SELECT * FROM transaksi WHERE norefb =
    $data6['norefb']
    IF count($nilai2) >= $min_support THEN
        INSERT norefb,total,rekomen,confidence,count,rank,lift TO rank
    ENDIF
ENDFOR
15 END

```

List 1. Pseudo-code Kalkulasi FP-Growth

Gambar 5 merupakan hasil akhir dari antarmuka implementasi fitur rekomendasi pada sistem pencarian di perpustakaan UKDW. Dimana *form* utama terdiri dari *form* menu pencarian, *form* hasil pencarian, *form* detail buku, dan *form* admin. Pada *form* detail buku bagian bawah terdapat hasil rekomendasi buku yang dihasilkan oleh algoritma fp-growth yang di batasi jumlah buku yang ditampilkan paling banyak 5 buah buku rekomendasi.



Gambar 5. Form Menu pencarian, Hasil Pencarian, Detail buku, dan Admin

3.2 Metode Pengujian

Pengujian program dilakukan untuk mengetahui seberapa kuat rekomendasi yang dihasilkan oleh sistem. Kekuatan suatu rekomendasi diukur menggunakan nilai *lift ratio*. Pengujian dilakukan dengan menghitung nilai lift ratio dari rule yang dihasilkan. Data yang digunakan dalam pengujian ini adalah seluruh buku yang memiliki rekomendasi yang dihasilkan oleh sistem dengan $\text{min support} = 7$. Data transaksi yang digunakan untuk data training sebanyak 80% data transaksi peminjaman buku perpustakaan yang di ambil secara acak. Sedangkan sisa nya, 20% data digunakan untuk pengujian selanjutnya.

Tabel 4.1.

Hasil Lift Ratio

| Rule | A | Count (A) | C | Count (C) | Conf (%) | Count (A U C) | Lift Ratio |
|------|----------|-----------|----------|-----------|----------|---------------|------------|
| 1 | HR000607 | 33 | HR231502 | 54 | 48.48 | 16 | 94.842 |
| 2 | HR000700 | 53 | HR241122 | 30 | 18.86 | 10 | 66.434 |
| 3 | HR001203 | 38 | HR001208 | 49 | 18.42 | 7 | 39.711 |
| 4 | HR001208 | 49 | HR001203 | 38 | 14.28 | 7 | 39.711 |
| 5 | HR001690 | 12 | HR990834 | 8 | 58.33 | 7 | 770.219 |
| 6 | HR001690 | 12 | HR250460 | 12 | 99.99 | 12 | 880.250 |
| 7 | HR001690 | 12 | HR250471 | 11 | 66.66 | 8 | 640.182 |
| 8 | HR200734 | 17 | KR100109 | 16 | 88.23 | 15 | 582.518 |
| 9 | HR200934 | 14 | PS005859 | 14 | 50.00 | 7 | 377.250 |
| 10 | HR200934 | 14 | KR100433 | 15 | 50.00 | 7 | 352.100 |
| 11 | HR200934 | 14 | HR230852 | 14 | 50.00 | 7 | 377.250 |
| 12 | HR200934 | 14 | PS015247 | 13 | 50.00 | 7 | 406.269 |
| 13 | HR201076 | 8 | SK061095 | 22 | 87.50 | 7 | 420.119 |
| 14 | HR201761 | 15 | PS010565 | 22 | 86.66 | 13 | 416.118 |
| 15 | HR201835 | 13 | PS000717 | 14 | 76.92 | 10 | 580.385 |
| 16 | HR201835 | 13 | PS000625 | 14 | 76.92 | 10 | 580.385 |
| 17 | HR210671 | 12 | PS007571 | 10 | 58.33 | 7 | 616.175 |
| 18 | HR220141 | 17 | HR220142 | 56 | 41.17 | 7 | 77.669 |
| 19 | HR220142 | 56 | HR220141 | 17 | 12.50 | 7 | 77.669 |
| 20 | HR221056 | 27 | PS009067 | 17 | 33.33 | 9 | 207.118 |
| 21 | HR230852 | 14 | HR200934 | 14 | 50.00 | 7 | 377.250 |
| 22 | HR230852 | 14 | PS015247 | 13 | 50.00 | 7 | 406.269 |
| 23 | HR231502 | 54 | HR000607 | 33 | 29.63 | 16 | 94.842 |
| 24 | HR240937 | 11 | SK061961 | 15 | 63.63 | 7 | 448.127 |
| 25 | HR241122 | 30 | HR000700 | 53 | 33.33 | 10 | 66.434 |
| 26 | HR250425 | 16 | SK061057 | 19 | 68.75 | 11 | 382.214 |
| 27 | HR250460 | 12 | HR990834 | 8 | 58.33 | 7 | 770.219 |
| 28 | HR250460 | 12 | HR250471 | 11 | 66.66 | 8 | 640.182 |
| 28 | HR250460 | 12 | HR001690 | 12 | 99.99 | 12 | 880.250 |
| 30 | HR250471 | 11 | HR990834 | 8 | 63.63 | 7 | 840.239 |
| 31 | HR250471 | 11 | HR250460 | 12 | 72.72 | 8 | 640.182 |
| 32 | HR250471 | 11 | HR001690 | 12 | 72.72 | 8 | 640.182 |
| 33 | HR260036 | 33 | SM990316 | 31 | 24.24 | 8 | 82.604 |
| 34 | HR990834 | 8 | HR250460 | 12 | 87.50 | 7 | 770.219 |
| 35 | HR990834 | 8 | HR250471 | 11 | 87.50 | 7 | 840.239 |
| 36 | HR990834 | 8 | HR001690 | 12 | 87.50 | 7 | 770.219 |
| 37 | KR100109 | 16 | HR200734 | 17 | 93.75 | 15 | 582.518 |
| 38 | KR100114 | 14 | KR100120 | 11 | 50.00 | 7 | 480.136 |
| 39 | KR100120 | 11 | KR100114 | 14 | 63.63 | 7 | 480.136 |
| 40 | KR100433 | 15 | HR200934 | 14 | 46.66 | 7 | 352.100 |
| 41 | KR100433 | 15 | PS005859 | 14 | 46.66 | 7 | 352.100 |

Tabel 4.1 (sambungan)

| | | | | | | | |
|----|----------|----|----------|----|-------|----|---------|
| 42 | KR100433 | 15 | HR230852 | 14 | 46.66 | 7 | 352.100 |
| 43 | KR100433 | 15 | PS015247 | 13 | 46.66 | 7 | 379.185 |
| 44 | PS000625 | 14 | PS000717 | 14 | 71.42 | 10 | 538.929 |
| 45 | PS000625 | 14 | HR201835 | 13 | 71.42 | 10 | 580.385 |
| 46 | PS000717 | 14 | HR201835 | 13 | 71.42 | 10 | 580.385 |
| 47 | PS000717 | 14 | PS000625 | 14 | 71.42 | 10 | 538.929 |
| 48 | PS002891 | 9 | PS003022 | 12 | 77.77 | 7 | 684.639 |
| 49 | PS003022 | 12 | PS014922 | 16 | 58.33 | 7 | 855.815 |
| 50 | PS003022 | 12 | PS002891 | 9 | 58.33 | 7 | 481.396 |
| 51 | PS005854 | 8 | HR210697 | 10 | 99.99 | 9 | 855.815 |

| No | Rule | Lift | Kemunculan pada data 20% | Support | Memenuhi |
|----|----------------------|---------|--------------------------|---------|----------|
| 1 | HR000607 => HR231502 | 94.842 | 5 | 0.18 | Ya |
| 2 | HR000700 => HR241122 | 66.434 | 0 | 0 | Tidak |
| 3 | HR001203 => HR001208 | 39.711 | 1 | 0.03 | Ya |
| 4 | HR001208 => HR001203 | 39.711 | 1 | 0.03 | Ya |
| 5 | HR001690 => HR990834 | 770.219 | 2 | 0.07 | Ya |
| 6 | HR001690 => HR250460 | 880.250 | 2 | 0.07 | Ya |
| 7 | HR001690 => HR250471 | 640.182 | 2 | 0.07 | Ya |
| 8 | HR200734 => KR100109 | 582.518 | 0 | 0 | Tidak |
| 9 | HR200934 => PS005859 | 377.250 | 0 | 0 | Tidak |
| 10 | HR200934 => KR100433 | 352.100 | 0 | 0 | Tidak |
| 11 | HR200934 => HR230852 | 377.250 | 0 | 0 | Tidak |
| 12 | HR200934 => PS015247 | 406.269 | 0 | 0 | Tidak |
| 13 | HR201076 => SK061095 | 420.119 | 0 | 0 | Tidak |
| 14 | HR201761 => PS010565 | 416.118 | 4 | 0.15 | Ya |
| 15 | HR201835 => PS000717 | 580.385 | 1 | 0.03 | Ya |
| 16 | HR201835 => PS000625 | 580.385 | 1 | 0.03 | Ya |
| 17 | HR210671 => PS007571 | 616.175 | 4 | 0.15 | Ya |
| 18 | HR220141 => HR220142 | 77.669 | 1 | 0.03 | Ya |
| 19 | HR220142 => HR220141 | 77.669 | 1 | 0.03 | Ya |
| 20 | HR221056 => PS009067 | 207.118 | 3 | 0.11 | Ya |
| 21 | HR230852 => HR200934 | 377.250 | 0 | 0 | Tidak |
| 22 | HR230852 => PS015247 | 406.269 | 0 | 0 | Tidak |
| 23 | HR231502 => HR000607 | 94.842 | 5 | 0.18 | Ya |
| 24 | HR240937 => SK061961 | 448.127 | 0 | 0 | Tidak |
| 25 | HR241122 => HR000700 | 66.434 | 0 | 0 | Tidak |
| 26 | HR250425 => SK061057 | 382.214 | 3 | 0.11 | Ya |
| 27 | HR250460 => HR990834 | 770.219 | 2 | 0.07 | Ya |
| 28 | HR250460 => HR250471 | 640.182 | 2 | 0.07 | Ya |
| 29 | HR250460 => HR001690 | 880.250 | 2 | 0.07 | Ya |
| 30 | HR250471 => HR990834 | 840.239 | 2 | 0.07 | Ya |
| 31 | HR250471 => HR250460 | 640.182 | 2 | 0.07 | Ya |
| 32 | HR250471 => HR001690 | 640.182 | 2 | 0.07 | Ya |
| 33 | HR260036 => SM990316 | 82.604 | 1 | 0.03 | Ya |
| 34 | HR990834 => HR250460 | 770.219 | 2 | 0.07 | Ya |
| 35 | HR990834 => HR250471 | 840.239 | 2 | 0.07 | Ya |
| 36 | HR990834 => HR001690 | 770.219 | 2 | 0.07 | Ya |
| 37 | KR100109 => HR200734 | 582.518 | 0 | 0 | Tidak |
| 38 | KR100114 => KR100120 | 480.136 | 2 | 0.07 | Ya |
| 39 | KR100120 => KR100114 | 480.136 | 2 | 0.07 | Ya |
| 40 | KR100433 => HR200934 | 352.100 | 0 | 0 | Tidak |

| | | | | | |
|----|----------------------|---------|---|------|-------|
| 41 | KR100433 => PS005859 | 352.100 | 0 | 0 | Tidak |
| 42 | KR100433 => HR230852 | 352.100 | 0 | 0 | Tidak |
| 43 | KR100433 => PS015247 | 379.185 | 0 | 0 | Tidak |
| 44 | PS000625 => PS000717 | 538.929 | 1 | 0.03 | Ya |
| 45 | PS000625 => HR201835 | 580.385 | 2 | 0.07 | Ya |
| 46 | PS000717 => HR201835 | 580.385 | 1 | 0.03 | Ya |
| 47 | PS000717 => PS000625 | 538.929 | 1 | 0.03 | Ya |
| 48 | PS002891 => PS003022 | 684.639 | 0 | 0 | Tidak |
| 49 | PS003022 => PS002891 | 684.639 | 0 | 0 | Tidak |
| 50 | PS003022 => PS014922 | 385.109 | 0 | 0 | Tidak |
| 51 | PS005859 => HR200934 | 377.250 | 0 | 0 | Tidak |

Tabel diatas adalah tabel hasil lift ratio dari rule yang dihasilkan oleh algoritma FP-Growth. Setelah nilai lift dari masing-masing rule telah didapat, di filter terlebih dahulu dengan nilai lift lebih besar daripada 1. Dari 51 rule pada data pada tabel diatas, tidak ada nilai lift yang dibawah dari 1. Hal tersebut dikarenakan perbandingan antara pembagi dan penyebut yang terlampau jauh, dan nilai penyebut yang sangat besar (22570), sehingga mengakibatkan nilai lift melebihi dari 1. Dengan melihat hasil nilai lift ratio diatas dapat disimpulkan, rule yang di hasilkan oleh algoritma *FP-Growth* terhadap 80% data transaksi peminjaman menghasilkan rule yang *positif correlated*. Hal selanjut nya yaitu menguji rule pada tabel diatas dengan 20% data transaksi peminjaman yang berbeda dari data 80% transaksi peminjaman yang digunakan untuk menghasilkan rule. Berikut Tabel pengujian rule terhadap data transaksi sebanyak 2641 data yang sebelumnya tidak dimasukkan ke dalam sistem.

Tabel 4.2.

Tabel Pengujian

Dari tabel diatas dapat dilihat bahwa rule yang memiliki nilai support tertinggi adalah rule yang memiliki jumlah kemunculan pada data 20% sebanyak 5. Sedangkan rule yang memiliki nilai kemunculan terendah adalah 1. Dan terdapat 20 rule yang tidak memenuhi kemunculan pada transaksi 20%. Pengujian dilakukan dengan melihat adakah transaksi pada 20% data yang memenuhi masing – masing rule. Rule yang memenuhi 20% data transaksi sebanyak 31 rule. Dengan diketahuinya banyak nya rule yang memenuhi pada 20% data transaksi yang telah ditentukan, dapat di hitung tingkat keakuratan algoritma, dengan cara menghitung rata-rata keberhasilan.

$$\text{Tingkat keakuratan algoritma} = \frac{\text{Jumlah Rule yang berhasil}}{\text{Total Rule}} \times 100\% \quad [4]$$

Sehingga nilai tingkat keakuratan algoritma sebesar = $0.6078 \times 100\% = 60.78 \%$

4. Kesimpulan

1. Algoritma *FP-Growth* dapat digunakan didalam kasus perekomendasiian buku dan mendapatkan hasil keluaran berupa rekomendasi buku yang terkait.
2. Tingkat keakuratan FP-Growth dalam memberikan rekomendasi buku adalah sebesar 60, 78%.

DAFTAR PUSTAKA

- Alamsyah, N., Khotimah, B, K., & Cahyani, A, D. (2012). *Penetapan Pola Konsumen Menggunakan Algoritma Frequent Pattern Growth Pada Cross Market Analysis*. Madura : Universitas Trunojoyo Madura.
- Basuki, S.(1991). *Pengantar Ilmu Perpustakaan*. Jakarta : Gramedia.
- Fowler, M. (2000). *UML Distilled (2nd Edition)*. Canada : Addison Wesley.
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining : Concepts and techniques (3rd Edition)*. Morgan Kaufmann.
- Han, J., Pei, J., & Yin, Yiwen. (2004). *Mining Frequent patterns without candidate generation*. Simon Fraser University. Netherlands: Kluwer Academic.
- Schneider, G., Winter, J, P. (2001). *Applying Use Case: A Practical Use Case (2nd Edition)*. California : Addison-Wesley Profesional.
- Tan, Pang-Ning., Steinbach, Michael., & Kumar, Vipin.(2004). *Introduction of Data Mining*. <http://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf>(diakses November 2012).
- Yongmei, L., Yong ,G. (2009). Application in Market Basket Analysis Based on FP-Growth Algorithm. *Dalam World Congress on Computer Science and Information Engineering*. China: Capital Normal University.
- Zhang, S., Zhang, C., & Yang, Q. (2003). *Data Preparation For Data Mining*. Applied Artificial Intelligence , 375-381.
- Zhang, W., Liao, Hongzhi., & Zhao, N. (2008). Research on the FP Growth Algorithm about Association Rule Mining. *Dalam International Seminar on Business and Information Management*.

Keywords: algoritma, frequent pattern growth

1. Pendahuluan

Seiring berkembangnya teknologi komunikasi, cara berkomunikasi pun beralih menggunakan media internet sebagai sarana komunikasi. Dengan internet komunikasi dapat dilakukan dengan lebih mudah, cepat dan dapat dilakukan dimana saja. Akan tetapi, komunikasi melalui internet mempunyai kelemahan yaitu penyadapan atau pencurian informasi. Aspek keamanan informasi dalam komunikasi sangat penting untuk diperhatikan. Untuk hal itu dibutuhkan metode untuk pengamanan informasi tersebut.

Kriptografi dan Steganografi merupakan metode pengamanan data untuk menjaga kerahasiaan dan keaslian data serta dapat melindungi aspek keamanan suatu informasi. Metode ini bertujuan agar informasi yang dikomunikasikan melalui internet tidak dapat disadap oleh pihak-pihak yang tidak bertanggung jawab pada sistem tersebut.

Oleh karena itu pada dibangun sebuah sistem yang memadukan algoritma kriptografi *twofish* dan steganografi *least significant bit (lsb)* yang dapat mengamankan informasi lalu menyisipkannya ke dalam sebuah file citra. Dengan adanya sistem pengamanan tersebut, tingkat keamanan informasi akan menjadi lebih tinggi.

¹ Mahasiswa Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Widyadarmas

² Dosen Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Widyadarmas

³ Dosen Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Widyadarmas