BAB II
TINJAUAN PUSTAKA

II.1 Profil Perpustakaan

Kantor perpustakaan umum dan kearsipan daerah Kabupaten Ciamis dibentuk berdasarkan peraturan daerah Kabupaten Ciamis nomor 17 tahun 2008 tentang organisasi perangkat daerah yang dijabarkan dalam peraturan Bupati Ciamis nomor 61 tahun 2008 tentang tugas, fungsi dan tata kerja unsur organisasi kantor perpustakaan umum dan kearsipan daerah.

Saat ini kantor perpustakaan umum dan kearsipan daerah Kabupaten Ciamis menempati tanah dan bangunan milik Pemerintah Kabupaten Ciamis dengan luas tanah 2.555 m2 dan luas bangunan 866,5 m2 dengan jumlah pegawai sebanyak 25 orang PNS dengan 4 orang jabatan fungsional (2 orang Pustakawan dan 2 orang Arsiparis). Untuk keberhasilan pengelolaan perpustakaan dan kearsipan yang optimal dituntut adanya peningkatan kualitas sumber daya manusia pengelola. Berkaitan hal tersebut kantor perpustakaan umum dan kearsipan daerah Kabupaten Ciamis terus memotivasi para pegawainya untuk mengikuti diklat Pustakawan dan Arsiparis serta Bintek bidang perpustakaan dan kearsipan yang diselenggarakan baik di tingkat Provinsi maupun tingkat nasional.

II.1.1 Visi dan Misi Perpustakaan

Berikut ini adalah visi dan misi dari perpustakaan umum dan kearsipan daerah Kabupaten Ciamis :

1 Visi

Visi dari perpustakaan umum dan kearsipan daerah Kabupaten Ciamis yaitu “Dengan iman dan takwa terwujudnya masyarakat Kabupaten Ciamis yang gemar membaca, berpengetahuan, terampil dan produktif serta berperilaku tertib arsip”.

2 Misi

Misi dari perpustakaan umum dan kearsipan daerah Kabupaten Ciamis adalah :

1. Mendorong masyarakat belajar seumur hidup (*long life education*).

2. Meningkatkan kualitas pengelolaan dan pelayanan perpustakaan dan kearsipan sesuai kebutuhan pengguna jasa / pemakai (*user oriented).*
3. Menyelamatkan arsip sebagai sumber informasi dan bahan bukti pertanggungjawaban penyelenggaraan kepemerintahan dan kemasyarakatan.
4. Meningkatkan pengetahuan dan pemahaman SDM aparatur dan masyarakat terhadap pentingnya arsip.

II.1.2 Strategi Perpustakaan

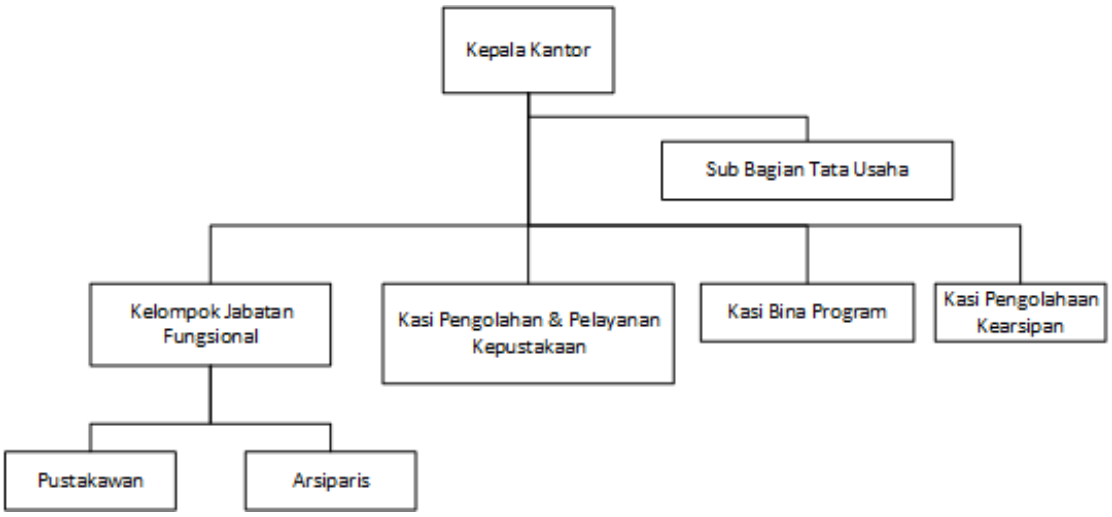
Adapun strategi dari Perpustakaan umum dan kearsipan daerah Kabupaten Ciamis adalah sebagai berikut :

1. Mengembangkan pembinaan dan pengembangan Perpustakaan Desa / Kelurahan,Perpustakaan Sekolah, Perpustakaan PKK, Perpustakaan RSUD, Perpustakaan, Pondok Pesantren, dan Perpustakaan Komunitas untuk meningkatkan minat baca dan budaya gemar membaca masyarakat
2. Mengoperasikan Mobil unit perpustakaan keliling ke objek yang tepat.
3. Mengusulkan pelaksanaan diklat fungsional daerah.
4. Mengusulkan peningkatan anggaran untuk pengadaan bahan pustaka.
5. Meningkatkan kualitas dan kuantitas sarana dan prasarana perpustakaan dan kearsipan.
6. Mengajukan anggaran pembangunan gedung Kantor Perpustakaan dan Kearsipan tahun 2014.
7. Mendayagunakan koleksi bahan pustaka yang ada untuk meningkatkan minat baca pelajar, dan mahasiswa melalui kegiatan lomba-lomba bedah buku dan silang layan.
8. Melakukan sosialisasi/ publikasi minat baca dan gemar baca masyarakat secara *continue* dan konsisten serta berkesinambungan melalui pameran, pemasangan baliho, spanduk, *banner*, media masa dan media cetak.
9. Melakukan penyuluhan kearsipan pada masyarakat.
10. Mengikutsertakan pegawai mengikuti diklat baik tingkat Nasional, Provinsi maupun di Kabupaten Ciamis.

11. Memanfaatkan kerja sama/ bantuan dari Badan Perpustakaan Daerah dan Badan Perpustakaan Daerah Provinsi Jabar serta PERPUSNAS RI dan Arsip Nasional RI.
12. Melakukan kerja sama dengan seluruh komponen masyarakat dalam pengembangan perpustakaan.
13. Mengembangkan standar pelayanan minimal dan meningkatkan kualitas pelayanan Perpustakaan dan Kearsipan.
14. Meningkatkan profesionalisme sumber daya aparatur Kantor Perpustakaan dan Kearsipan Daerah Kabupaten Ciamis, baik di setiap jenjang struktural, fungsional, maupun pelaksana.
15. Meningkatkan efektivitas pembinaan Kearsipan.
16. Meningkatkan pengumpulan arsip dari di Baleka, Kecamatan ,dan Desa/ Kelurahan.
17. Mengembankan kerja sama dengan lembaga terkait (penerbit, pengusaha, komunitas-komunitas baca) dalam pengembangan koleksi bahan pustaka.

II.1.3 Struktur Organisasi Perpustakaan

Struktur organisasi Perpustakaan umum dan kearsipan daerah Kabupaten Ciamis dapat dilihat pada Gambar II-1.



Gambar II-1 Struktur Organisasi

II.1.4 Koleksi Perpustakaan Umum

Berikut ini merupakan koleksi berdasarkan golongan dan jumlah koleksi perpustakaan umum dan kearsipan daerah Kabupaten Ciamis :

1. Jenis Koleksi

Bahan bacaan dan informasi yang disediakan meliputi berbagai ilmu pengetahuan dari mulai bacaan anak-anak, pelajar, mahasiswa, pegawai dan masyarakat umum. Selengkapnya mengenai jenis koleksi dapat dilihat pada Gambar II-2.

Golongan	Jenis Koleksi
000	Karya Umum
100	Filsafat dan Psikologi
200	Agama
300	Ilmu-ilmu Sosial
400	Bahasa dan Linguistik
500	Ilmu-ilmu Murni
600	Teknologi
700	Kesenian
800	Kesusastraan dan Fiksi
900	Sejarah dan Geografi

Gambar II-2 Jenis Koleksi Perpustakaan

2. Jumlah Koleksi

Pada saat ini kantor perpustakaan umum dan kearsipan daerah Kabupaten Ciamis mempunyai koleksi bahan pustaka sebanyak 28.448 eksemplar (23.706 judul).

II.1.5 Jenis Layanan

Adapun layanan yang tersedia di Kantor Perpustakaan dan Kearsipan Daerah Kabupaten Ciamis adalah sebagai berikut :

1. Layanan sirkulasi

   Adalah layanan berupa peminjaman dan pengembalian bahan pustaka, dengan sistem otomasi perpustakaan. Untuk meningkatkan kualitas layanan kepada pemustaka/ pengguna jasa, Kantor Perpustakaan Umum dan Kearsipan Daerah Kabupaten Ciamis membuka jam layanan sebagai berikut :

   Hari Senin – Jumat : 07.30 – 16.30 WIB

   Hari Sabtu 08.30 – 15.00 WIB

2. Layanan Referensi

   Layanan bantuan penelusuran *litearure*, kamus, ensiklopedia, majalah koleksi daerah dan sebagainya.

3. Layanan Internet

   Layanan akses informasi global melalui fasilitas *online* internet, baik diakses sendiri oleh pengguna jasa maupun dibantu oleh operator.

4. Layanan Mobil Unit Perpustakaan

   Layanan mobil unit perpustakaan keliling di 12 Kecamatan, 17 titik layanan.

II.2 Landasan Teori

Berikut adalah materi atau teori apa saja yang digunakan dalam penelitian.

II.2.1 Data Mining

Data mining merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, *statistic*, *database* dan visualisasi untuk penanganan permasalahan pengambilan informasi dari data yang besar [2]. *Data mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu :

1. Asosiasi (*Association*)

Tugas asosiasi adalah menemukan atribut yang sering muncul dalam waktu yang bersamaan. Atau dalam dunia bisnis dikenal dengan analisis keranjang belanja [7].

2. Deskriptif

Deskriptif bertujuan untuk menggambarkan pola dan kecenderungan di dalam data, kemudian memberikan penjelasan pola dan kecenderungan tersebut.

3. Klasifikasi (*Classification*)

Klasifikasi adalah proses penemuan model atau fungsi yang membedakan kelas data atau konsep untuk dapat memperkirakan kelas dari suatu objek.

4. Pengklusteran (*Clustering*)

Penglkusteran merupakan pengelompokan ,pengamatan atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan.

5. Prediksi (*Prediction*)

Tujuan dari prediksi adalah untuk memperkirakan nilai di masa mendatang menggunakan nilai dari hasil prediksi.

6. Estimasi

Tujuan dari estimasi adalah untuk menghasilkan suatu model yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi. Variabel target estimasi lebih ke arah *numeric* daripada ke arah kategori.

II.2.2 Tahapan Data Mining

Istilah *data mining* dan *knowledge discovery in database* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar [7]. Proses KDD secara garis besar dapat dijelaskan sebagai berikut.

1. *Data Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional [7].

2. *Cleaning*

Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus KDD. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak (*tipografi)*. Juga dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal [7].

3. *Data mining*

Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan [7].

4. *Interpretation/ Evaluation*

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut *interpretation.* Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya [7].

II.2.3 Association Rule

Association rule adalah teknik *data mining* yang berguna untuk menemukan suatu korelasi atau pola yang terpenting/ menarik dari sekumpulan data besar [3]. *Association rule* merupakan suatu proses untuk menemukan semua aturan asosiatif yang memenuhi syarat minimum untuk *support (minsup)* dan syarat minimum untuk *confidence (mincof)* pada sebuah *database* [8]. Dalam asosiasi terdapat istilah *antecedent* dan *consequent*, dimana niali A adalah *antecedent* dan B adalah *consequent.* Untuk mencari nilai *support* dari 2-*item* dapat diperoleh dari rumus berikut [8]

$$Support(A,B) = P\ (A \cap B) = \frac{\text{Jumlah transaksi yang mengandung A dan B}}{Total\ transaksi}$$

Setelah semua *frequent item* dan l*arge itemset* ditemukan, dapat dicari semua *association rules* yang memenuhi syarat minimum untuk *confidence (minconf)* dengan menggunakan rumus berikut ini [8]

$$Confidence\ (A \rightarrow B) = P\ (B|A) = \frac{\text{Jumlah transaksi yang mengandung A dan B}}{Jumlah\ transaksi\ yang\ mengandung\ A}$$

II.2.4 FP-Tree

Frequent Pattern Tree (FP-Tree) merupakan struktur penyimpanan data yang dimampatkan [9]. Pada algoritma *FP-Growth*, penelusuran *FP-Tree* merupakan langkah yang wajib dilakukan untuk memperoleh *conditional FP-Tree* [10]. *FP-Tree* dibentuk dengan cara membaca kumpulan data dalam sebuah transaksi dalam waktu yang bersamaan, kemudian memetakan transaksi tersebut ke dalam lintasan *FP-Tree.* Dalam sebuah transaksi yang berbeda, kemungkinan munculnya *item* yang sama sangatlah besar, sehingga memungkinkan lintasan dalam *FP-Tree* dapat

saling menimpa. Semakin banyak lintasan yang saling menimpa satu sama lain, maka proses pemadatan dalam *FP-Tree* akan semakin baik.

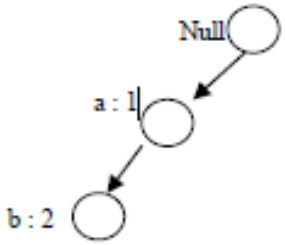
Dalam "Penerapan Struktur *FP-Tree* dan Algoritma *FP-Growth* dalam Optimasi Penentuan *Frequent Itemset*" (David Samuel, 2007) menjelaskan misal I = $\{a_1, a_2, \ldots, a_n\}$ adalah kumpulan dari item. Dan basis data trarnsaksi DB = $\{T_1, T_2, \ldots, T_n\}$, dimana T_i (i € [1 ... n]) adalah sekumpulan 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 (batas ambang minimum *support*) yang telah didefinisikan sebelumnya. Permasalahan mencari pola *frequent* dengan batas ambang minimum *support count* inilah yang coba dipecahkan oleh *FP-Growth* dengan bantuan struktur *FP-Tree.*

Contoh misalkan diberikan data transaksi sebagai berikut, dengan minimum *support count* = 2.

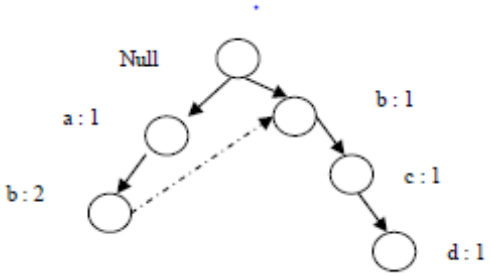
Tabel II-1 Tabel Transaksi Data Awal [9]

TID	Item
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}

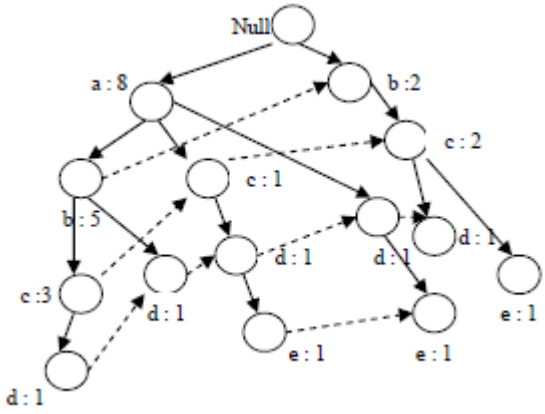
Gambar di bawah ini memberikan ilustrasi mengenai langkah pembentukan *FP-Tree.*



Gambar II-3 Hasil Pembentukan FP-Tree Setelah Pembacaan TID1 [9]



Gambar II-4 Hasil Pembentukan FP-Tree Setelah Pembacaan TID2 [9]



Gambar II-5 Hasil Pembentukan FP-Tree Setelah Pembacaan TID10 [9]

II.2.5 Algoritma Improved FP-Growth

FP-Growth adalah salah satu alternatif algoritma yang dapat digunakan untuk menetukan himpunan data yang paling sering muncul *(frequent itemset)* dalam sebuah kumpulan data [9]. Pada algoritma *FP-Growth* menggunakan konsep pembangunan *tree,* yang biasa disebut *FP-Tree,* dalam pencarian *frequent itemsets* bukan menggunakan *generate candidate* seperti yang dilakukan pada algoritma

Apriori [4]. *FP-Tree* dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam *FP-Tree* [9].

Algoritma *FP-Growth* dibagi menjadi tiga langkah utama, yaitu :

1. Tahap pembangkitan *Conditional Pattern Base*

Conditional Pattern Base merupakan *subdatabase* yang berisi *prefix path* (lintasan *prefix*) dan *suffix pattern* (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui *FP-Tree* yang telah dibangun sebelumnya [9].

2. Pembangkitan *Conditional FP-Tree*

Pada tahap ini, *support count* dari setiap item pada setiap *conditional pattern base* dijumlahkan, lalu setiap *item* yang memiliki jumlah *support count* lebih besar sama dengan *minimum support count* akan dibangkitkan dengan *conditional FP-Tree* [9].

3. Tahap pencarian *frequent itemset*

Apabila *conditional FP-Tree* merupakan lintasan tunggal *(single path)*, maka didapatkan *frequent itemset* dengan melakukan kombinasi *item* untuk setiap *conditional FP-Tree.* Jika bukan lintasan tunggal, maka dilakukan pembangkitan *FP-Growth* secara rekrusif [9].

Algoritma *Improved FP-Growth*, merupakan algoritma yang dikembangkan dari algoritma *FP-Growth.* Algoritma *Improved FP-Growth* memiliki kelebihan dari sisi kecepatan. Berikut ini merupakan contoh penerapan dari algoritma *improved FP-Growth* [5].

Data transaksi yang digunakan yaitu sebanyak 5 transaksi, dan ditentukan nilai *minimum support* = 3. Data transaksi awal dapat dilihat pada Tabel II-2.

Tabel II-2 Transaksi Awal [5]

TID	Items
T100	f, a, c, d, g, i, m, p
T200	a, b, c, f, l, m, o
T300	b, f, h, j, o
T400	b, c, k, s, p
T500	a, f, c, e, l, p, m, n

Langkah 1 : *Scan* seluruh transaksi dan hitung frekuensi kemunculan *item* di dalam seluruh transaksi, dan urutkan secara menaik (*ascending*) berdasarkan frekuensi kemunculannya dan masukan kedalam tabel All_item, pada saat yang bersamaan buat tabel *header* yang berisikan *itemid,*dan *support count.*. Hasil dari penghitungan frekuensi kemunculan dapat dilihat pada Tabel II-3.

Tabel II-3 Tabel All_item

Items	Support Count
d	1
e	1
g	1
h	1
i	1
j	1
k	1
n	1
s	1
l	2
o	2
a	3
b	3
m	3
p	3
c	4
f	4

Tabel II-4 Tabel Header [5]

Itemid	Support count
...	...

Langkah 2 : Kemudian cek nilai *minimum support* untuk setiap *item*, jika memenuhi kondisi nilai *item support >= minimum support*, maka masukan dalam urutan L-*order* ke dalam tabel lain yang diberi nama *linear list.* Tabel l*inear list* terdiri dari.

Tabel II-5 Tabel Linear list [5]

Itemid	Count
...	...

item yang memenuhi nilai *minimum support* dapat dilihat pada Tabel II-6.

Tabel II-6 Item yang Memenuhi Minimum Support

Item	Support Count
a	3
b	3
m	3
p	3
c	4
f	4

Langkah 3 : Sesuai dengan *flowchart* seluruh transaksi berjalan dan menghasilkan *frequent itemset* berdasarkan pembacaan terhadap tabel *linear list* secara satu per satu.

Tabel II-7 Tabel Transaksi (T100 sampai T500) Linear List [5]

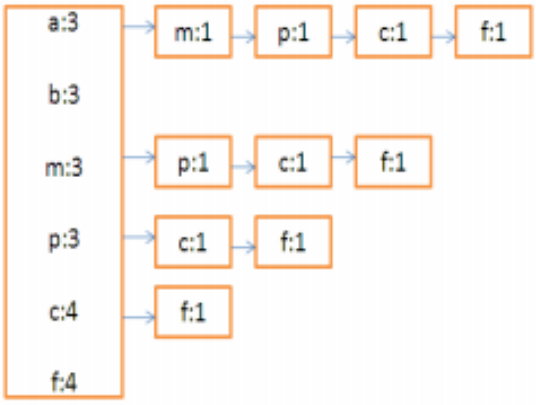
TID	Items
T100	a, m, p, c, f
T200	a, b, m, c, f
T300	b, f
T400	b, p, c
T500	a, m, p, c, f

Tahapan dari pembangunan *flowchart* adalah sebagai berikut :

1. Pembacaan T100 : a, m, p, c, f

Langkah pertama yaitu buat *list* yang berisi *item*, dan nilai *support count* yang merupakan *item* yang memenuhi nilai *minimum support* dan telah diurutkan sebelumnya, kemudian cari *item* pertama dalam transaksi yaitu *item* a dalam *list* yang telah dibuat, kemudian buat lintasan untuk menyimpan transaksi selanjutnya yaitu *item* m, dan beri nilai *support count* 1 untuk setiap transaksi yang dilalui sehingga terbentuk pada *list* a = (a:3 → m:1), karena *item* m bukan merupakan *item* terakhir dalam transaksi, maka buat lintasan baru dari *item* m, untuk menyimpan transaksi berikutnya yaitu p dan beri nilai *support count* 1 sehingga terbentuk pada *list* a = (a:3 → m:1 → p:1), pada saat yang bersamaan cari *item* m dalam *list*, kemudian buat juga lintasan untuk menyimpan transaksi yang sama yaitu p dan beri nilai *support count* 1 sehingga terbentuk pada *list* m = (m:3 → p:1), karena *item* p juga bukan merupakan *item* terakhir di dalam transaksi, maka buat lintasan baru

dari *item* p untuk menyimpan transaksi selanjutnya yaitu c dan beri nilai *support count* 1, sehingga terbentuk pada *list* a = (a:3 → m:1 → p:1 → c:1), begitu juga pada *list* m sehingga terbentuk pada *list* m = (m:3 → p:1 → c:1), dan pada saat yang bersamaan cari *item* p yang merupakan *item* ke 3 dalam transaksi, dalam *list* kemudian buat lintasan untuk menyimpan transaksi yang sama yaitu c, dan beri nilai *support count* 1, sehingga terbentuk dalam *list* p = (p:3 → c:1), proses ini dilakukan sampai dengan *item* terakhir di dalam transaksi yaitu *item* f. Hasil dari pembentukan *flowchart* untuk transaksi T100 : a, m, p, c, f, dapat dilihat pada Gambar II-6.

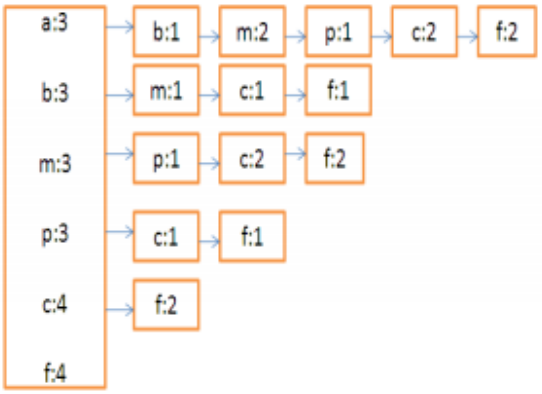


Gambar II-6 Setelah Pembacaan T100 [5]

2. Pembacaan T200 : a, b, m, c, f

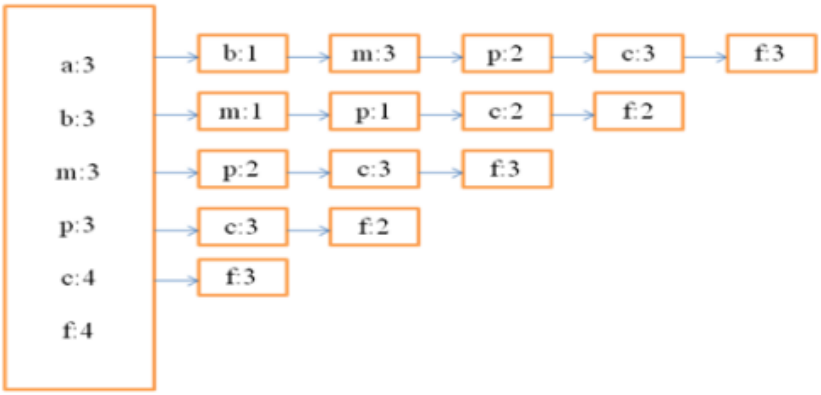
Pada pembacaan T200, tahapan dalam pembuatan lintasan hampir sama seperti pada saat pembacaan T100, caranya yaitu cari *item* pertama di dalam transaksi yaitu *item* a dalam *list*, kemudian buat lintasan untuk menyimpan transaksi selanjutnya yaitu b dan beri nilai *support count* 1, karena pada *list* a telah terbentuk lintasan sebelumnya yaitu (a:3 → m:1 → p:1 → c:1 → f:1), maka dilakukan perbandingan urutan antara *item* ke 2 yang akan ditambahkan yaitu *item* b, dengan *item* yang sudah ada sebelumnya yaitu *item* m, karena *item* b memiliki urutan prioritas terlebih dahulu dibandingkan dengan *item* m, maka untuk *item* b akan dibuat lintasan terlebih dahulu kemudian *item* b di geser setelah pembuatan lintasan *item* b, sehingga *list* a berubah menjadi (a:3 → b:1 → m:1 →p:1 → c:1 →f:1), untuk *item* selanjutnya sama seperti pada pembacaan T100, hanya saja jika *item* di dalam *list*

sudah ada sebelumnya, maka setiap *item* yang memiliki lintasan yang sama, ditambahkan 1 nilai *support count* untuk lintasan yang dilaluinya. Hasil dari pembacaan T200 dapat dilihat pada Gambar II-7.



Gambar II-7 Setelah Penelusuran T200 [5]

Langkah yang sama yang diterapkan pada pembacaan T100, T200, sampai dengan transaksi terakhir yaitu T500, kemudian *frequent item* akan dibangun berdasarkan hasil pembentukan transaksi yang terakhir. Hasil dari pembentukan *flowchart* dari seluruh transaksi dapat dilihat pada Gambar II-8.



Gambar II-8 Setelah Pembacaan T500 [5]

Aturan untuk membangun *frequent item* :

Tahapan selanjutnya adalah pembentukan *frequent pattern* berdasarkan *flowchart* yang sudah terbentuk sebelumnya, dengan cara mengambil *item* dari dalam *list* yang memiliki nilai *support count* >= *minimum support* yang telah ditentukan sebelumnya yaitu = 3, misalkan *item* m:3 dan p:2 tidak memenuhi nilai *minimum*

support, sehingga tidak menghasilkan *frequent pattern*, kemudian *item* m:3 dan c:3, memenuhi nilai *minimum support* sehingga menghasilkan *frequent pattern* mc:3, kemudian *item* m:3 dan f:3, memenuhi nilai *minimum support*, dan menghasilkan *frequent pattern* mf:3, dan yang terakhir adalah *item* m:3,c:3, dan f:3 memenuhi nilai *minimum support* sehingga menghasilkan *frequent pattern* mcf:3. Hasil dari pembentukan *frequent item* dapat dilihat pada Tabel II-8.

Tabel II-8 Generated Frequent item sets [5]

Items	Frequent Pattern
f	(f:4)
c	(c:4) (cf:3)
p	(p:3) (pc:3)
m	(m:3) (mc:3) (mf:3) (mcf:3)
b	(b:3)
a	(a:3) (am:3) (ac:3) (af:3) (amc:3) (acf:3) (amf:3) (amcf:3)

II.2.6 Basis Data

Database adalah kumpulan data yang saling berkaitan yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak untuk menghasilkan informasi yang bermanfaat. Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan kembali [11]. Proses dasar yang terdapat di dalam database, yaitu :

1 Pembuatan data baru (*create database*)
2 Penambahan data (*insert*)
3 Mengubah data (*update*)
4 Menghapus data (*delete*)

II.2.7 Database Management System (DBMS)

DBMS adalah *software* yang menangani semua akses ke basis data. Secara konsep apa yang terjadi adalah sebagai berikut [11].

1 *User* melakukan pengaksesan basis data untuk informasi yang diperlukannya menggunakan suatu bahasa manipulasi data, biasanya disebut SQL.

2 DBMS menerima *request* dari *user*, pemetaan eksternal/ konseptual, skema konseptual, pemetaan konseptual/ internal, dan struktur penyimpanan.
3 DBMS memeiksa skema eksternal *user*, pemetaan eksternal/ konseptual, skema konseptual, pemetaan konseptual/ internal, dan struktur penyimpanan.
4 DBMS mengeksekusi operasi-operasi yang diperlukan untuk memenuhi permintaan *user*.

II.2.8 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) menggambarkan aliran data atau informasi di mana di dalamnya terlihat keterkaitan diantara data-data yang ada, dari mana asal data, dan ke mana tujuan data yang keluar dari sebuah sistem, interaksi antara data yang tersimpan dan proses yang dilakukan pada data tersebut.

Teknik pembuatan DFD dimulai dengan menggambarkan sistem secara global, dan dilanjutkan dengan analisis pada masing-masing bagian. Diagram konteks merupakan diagram yang menggambarkan sistem secara menyeluruh, atau dapat dikatakan sebagai DFD level 0.

Untuk penggambaran lebih detail dalam sebuah sistem, dapat dilakukan dengan menggambarkan DFD level 1, 2 dan seterusnya. Sebagian besar penggambaran akan dihentikan pada DFD level 2. *Data Flow Diagram* secara umum harus dapat dimengerti oleh *programmer* karena akan menjadi panduan untuk pembuatan program [15].

II.2.9 Entity Relationship Diagram (ERD)

Pada dasarnya ERD (*Entity Relationship Diagram*) adalah sebuah diagram yang secara konseptual memetakan hubungan antara penyimpanan pada diagram DFD. ERD digunakan untuk melakukan pemodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah *database* yang baik.

Sebuah *entiti* adalah sebuah “benda” (*thing*) atau “objek” (*object*) di dunia nyata yang dapat dibedakan dari semua objek lainnya. *Entity sets* adalah

sekumpulan *entiti* yang mempunyai tipe yang sama. Kesamaan tipe ini dilihat dari atribut/*property* yang dimiliki oleh setiap *entiti* [11].