

**ANALISIS DATA POLA PEMINJAMAN BUKU  
MENGUNAKAN ALGORITMA FP-GROWTH  
(STUDI KASUS PERPUSTAKAAN OTORITA BATAM)**

**TUGAS AKHIR**

Oleh :

**Suryati Ali 3311201082**

Disusun untuk memenuhi syarat kelulusan Program Diploma III



**PROGRAM STUDI TEKNIK INFORMATIKA  
POLITEKNIK NEGERI BATAM  
BATAM  
2015**

# **HALAMAN PENGESAHAN**

**ANALISIS DATA POLA PEMINJAMAN BUKU  
MENGUNAKAN ALGORITMA FP-GROWTH  
(STUDI KASUS PERPUSTAKAAN OTORITA BATAM)**

**Oleh :**

**Suryati Ali (3311201082)**

Tugas Akhir ini telah diterima dan disahkan  
sebagai persyaratan untuk memperoleh gelar

Ahli Madya

di

**PROGRAM STUDI DIPLOMA 3 TEKNIK INFORMATIKA  
POLITEKNIK NEGERI BATAM**

Batam, 20 Januari 2015

Disetujui oleh:

Pembimbing,

**Yeni Rokhayati, M.Sc**

**NIK. 112093**

## HALAMAN PERNYATAAN

Dengan ini, saya:

NIM : 3311201082

Nama : Suryati Ali

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

ANALISIS DATA POLA PEMINJAMAN BUKU  
MENGUNAKAN ALGORITMA FP-GROWTH  
(STUDI KASUS PERPUSTAKAAN OTORITA BATAM)

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 20 Januari 2015

**Suryati Ali**  
33110201082

## KATA PENGANTAR

Dengan rahmat Tuhan Yang Maha Esa, penulis dapat menyelesaikan Tugas Akhir yang berjudul “Analisis Data Pola Peminjaman Buku Menggunakan Algoritma FP-Growth (Studi Kasus Perpustakaan Otorita Batam)” ini sesuai dengan waktu yang telah ditetapkan. Laporan Tugas Akhir ini dibuat untuk memenuhi kriteria penilaian dari mata kuliah wajib yang harus diselesaikan dan dipertanggungjawabkan sebagaimana mestinya.

Laporan yang ditujukan guna melengkapi syarat kelulusan Program Studi Teknik Informatika Politeknik Negeri Batam ini dalam pengerjaannya mendapat banyak arahan dan bimbingan dari berbagai pihak, oleh karena itu dalam kesempatan ini penulis hendak mengucapkan ungkapan terima kasih yang tak terhingga kepada:

1. Allah S.W.T., yang senantiasa memberikan limpahan anugerah dan nikmat-Nya berupa nikmat Iman, nikmat Islam, dan nikmat kesehatan yang selalu mengiringi penulis.
2. Keluarga, terutama kedua orang tua penulis yang selalu menjadi inspirasi dan memberikan semangat, kasih sayang serta dukungan kepada penulis.
3. Ibu Yeni Rokhayati, M.Sc, selaku pembimbing Tugas Akhir yang telah meluangkan banyak waktu dan tak henti-hentinya memberikan motivasi, bimbingan serta arahan demi kelancaran Tugas Akhir ini.
4. Pak Dwi Ely Kurniawan, M.Kom, selaku pengampuh mata kuliah Tugas Akhir ini yang telah memberikan arahan dalam pengerjaan Tugas Akhir ini.
5. Para dosen, teman-teman seperjuangan TA, dan seluruh pihak yang ikut serta memberikan motivasi dan membantu dalam menyelesaikan Laporan Tugas Akhir ini.

Penulis menyadari Tugas Akhir ini tidak luput dari sedikit banyaknya ketidaksesuaian baik dari segi kata-kata maupun strukturnya. Penulis mengharapkan adanya masukan ide-ide baru berupa kritik dan saran untuk penyempurnaan Tugas Akhir ini.

Akhir kata penulis mengucapkan terima kasih, semoga dengan adanya Tugas Akhir ini dapat bermanfaat bagi pembaca.

Batam, 20 Januari 2015

Penulis

# ABSTRAK

## ANALISIS DATA POLA PEMINJAMAN BUKU MENGUNAKAN ALGORITMA FP-GROWTH (STUDI KASUS PERPUSTAKAAN OTORITA BATAM)

Perpustakaan Otorita Batam masih menggunakan prosedur peminjaman yang dilakukan secara manual yaitu menggunakan buku untuk pencatatan peminjaman buku. Data peminjaman buku yang tercatat secara manual tersebut dapat mengalami penumpukkan karena jika buku tersebut penuh buku tersebut tidak digunakan lagi.

*Data mining* merupakan salah satu alat analisis yang dapat mengatasi penumpukkan data tersebut. Tugas akhir ini melakukan analisis dan pembuatan sistem tentang manfaat *data mining* dalam peminjaman buku di perpustakaan.

Algoritma *data mining* yang digunakan untuk menganalisis data peminjaman buku yaitu *FP-Growth* dari metode asosiasi. Algoritma ini berupa pembangunan *FP-tree* yang akhirnya akan menghasilkan beberapa aturan asosiasi. Aturan asosiasi inilah yang akan digunakan untuk meletakkan buku-buku yang sering dipinjam secara berdekatan. Aturan asosiasi yang diperoleh, diterapkan ke dalam sistem yang dibangun menggunakan bahasa pemrograman *Java Netbeans*. Sehingga asosiasi buku dari satu kategori buku dapat ditampilkan melalui sistem ini agar mempermudah pengurus dan pengunjung perpustakaan.

Kata kunci: *data mining*, *FP-growth*, aturan asosiasi, peminjaman buku, perpustakaan.

# ABSTRACT

## ANALISIS DATA POLA PEMINJAMAN BUKU MENGUNAKAN ALGORITMA FP-GROWTH (STUDI KASUS PERPUSTAKAAN OTORITA BATAM)

Otorita Batam's Library still using the procedure of lending books manually that uses the book for recording the lending books. The data of lending books recorded manually can experience buildup because if that book is full then the book can't used again.

*Data mining* is one of analyze's tool that can fix the buildup to data. This final project do the analyze and make the system about benefits of data mining in lending books in the library.

Algorithm of data mining used to analyze the data of lending books is *FP-Growth* from association method. This algorithm in the form of development *FP-tree* that finally will result in some association rules. This association rules will be used to puts the books which are often borrowed by the adjacent. Association rules which is obtained, applied to the system built using programming language *Java Netbeans*. So the association of books from category books can showed by this system in order to facilitate administrators and visitors of the library.

Keywords: *data mining*, *FP-growth*, association rules, lending books, library.

# DAFTAR ISI

|   |      |
|---|------|
| HALAMAN PENGESAHAN .....  | ii   |
| HALAMAN PERNYATAAN.....   | iii  |
| KATA PENGANTAR.....   | iv   |
| ABSTRAK .....   | vi   |
| ABSTRACT .....  | vii  |
| DAFTAR ISI .....  | viii |
| DAFTAR GAMBAR.....  | xi   |
| DAFTAR TABEL .....  | xiii |
| BAB I PENDAHULUAN .....   | 1    |
| 1.1 Latar Belakang.....   | 1    |
| 1.2 Rumusan Masalah.....  | 2    |
| 1.3 Batasan Masalah .....   | 2    |
| 1.4 Tujuan Penelitian .....                                       | 2    |
| 1.5 Sistematika Penulisan .....                                   | 3    |
| BAB II LANDASAN TEORI.....  | 4    |
| 2.1 Penelitian Sebelumnya.....                                    | 4    |
| 2.2 Analisis Data.....  | 5    |
| 2.3 Data Mining .....   | 6    |
| 2.3.1 Metode Asosiasi .....                                       | 8    |
| 2.3.2 Algoritma FP-Growth .....                                   | 10   |
| 2.4 Perangkat-Perangkat Lunak yang Mendukung dalam Analisis ..... | 21   |
| 2.4.1 Microsoft Excel.....  | 21   |
| 2.4.2 Microsoft Visio .....                                       | 22   |
| 2.4.3 Netbeans .....  | 22   |
| 2.4.4 XAMPP.....  | 22   |
| 2.4.5 StarUML.....  | 23   |
| BAB III ANALISIS DAN PERANCANGAN .....                            | 25   |
| 3.1 Proses Pengumpulan Data .....                                 | 25   |
| 3.2 Tahapan Preprocessing Data.....                               | 25   |
| 3.2.1 Transformasi Data .....                                     | 25   |
| 3.2.2 Penyeleksian Data .....                                     | 26   |
| 3.3 Penerapan Algoritma FP-Growth .....                           | 27   |

|   |  |           |
|---|--|-----------|
| 3.3.1                                   | Menghitung Support Count 1-Item .....              | 28        |
| 3.3.2                                   | Pembangunan FP-Tree .....                          | 30        |
| 3.3.3                                   | Pencarian Frequent Itemset .....                   | 32        |
| 3.3.4                                   | Pencarian Confidence dan Aturan Asosiasi .....     | 37        |
| 3.4                                     | Deskripsi Sistem .....                             | 40        |
| 3.5                                     | Lingkungan Pengembangan.....                       | 41        |
| 3.6                                     | Perancangan Sistem .....                           | 42        |
| 3.7                                     | Use Case Diagram .....                             | 42        |
| 3.8                                     | Skenario Use Case .....                            | 42        |
| 3.8.1                                   | Skenario Mencari Asosiasi Buku .....               | 43        |
| 3.8.2                                   | Skenario Mencari Daftar Buku .....                 | 43        |
| 3.8.3                                   | Skenario Mencari Informasi Buku .....              | 43        |
| 3.9                                     | Sequence Diagram .....                             | 43        |
| 3.9.1                                   | Sequence Diagram Mencari Asosiasi Buku .....       | 44        |
| 3.9.2                                   | Sequence Diagram Mencari Informasi Buku .....      | 44        |
| 3.9.3                                   | Sequence Diagram Mencari Daftar Buku .....         | 46        |
| 3.10                                    | Class Diagram.....                                 | 46        |
| 3.11                                    | Desain Antar Muka Sistem .....                     | 47        |
| 3.11.1                                  | Desain Antar Muka Halaman Utama .....              | 48        |
| 3.11.2                                  | Desain Antar Muka Asosiasi Buku .....              | 49        |
| 3.11.3                                  | Desain Antar Muka Informasi Buku .....             | 51        |
| 3.11.4                                  | Desain Antar Muka Daftar Buku .....                | 53        |
| 3.12                                    | ER Diagram .....                                   | 55        |
| 3.12.1                                  | Spesifikasi Database Databuku .....                | 55        |
| <b>BAB IV HASIL DAN PEMBAHASAN.....</b> |  | <b>56</b> |
| 4.1                                     | Algoritma .....                                    | 56        |
| 4.1.1                                   | Algoritma Mencari Asosiasi Buku .....              | 56        |
| 4.1.2                                   | Algoritma Mencari Informasi Buku .....             | 57        |
| 4.1.3                                   | Algoritma Mencari Daftar Buku .....                | 57        |
| 4.2                                     | Implementasi Basis Data .....                      | 58        |
| 4.3                                     | Implementasi Kelas.....                            | 58        |
| 4.4                                     | Implementasi Perancangan AntarMuka.....            | 59        |
| 4.4.1                                   | Hasil Implementasi Perancangan Halaman Utama ..... | 59        |
| 4.4.2                                   | Hasil Implementasi Perancangan Asosiasi Buku ..... | 60        |

|   |     |
|---|-----|
| 4.4.3 Hasil Implementasi Perancangan Informasi Buku .....             | 64  |
| 4.4.4 Hasil Implementasi Perancangan Daftar Buku .....                | 66  |
| 4.5 Pengujian .....   | 68  |
| 4.5.1 Strategi Pengujian .....  | 68  |
| 4.5.2 Skenario Pengujian.....   | 68  |
| 4.5.3 Hasil Pengujian .....   | 68  |
| BAB V KESIMPULAN DAN SARAN .....                                      | 71  |
| 5.1 Kesimpulan .....  | 71  |
| 5.2 Saran .....   | 72  |
| DAFTAR PUSTAKA.....   | 73  |
| LAMPIRAN .....  | 73  |
| Lampiran A (Data Mentah) .....  | A-1 |
| Lampiran B (Transformasi Data) .....                                  | B-1 |
| Lampiran C (Penyeleksian Data) .....                                  | C-1 |
| Lampiran D (Transaksi Setekah Diurutkan) .....                        | D-1 |
| Lampiran E (FP-Tree) .....  | E-1 |
| Lampiran F (Lintasan Berakhir di Item) .....                          | F-1 |
| Lampiran G (Simpul yang Dibuang) .....                                | G-1 |
| Lampiran H (Conditional FP-Tree) .....                                | H-1 |
| Lampiran I (Subsets dan Confidence) .....                             | I-1 |
| Lampiran J (Layout Penempatan Buku Perpustakaan Otorita Batam) .....  | J-1 |
| Lampiran K (Layout Penempatan Buku Berdasarkan Aturan Asosiasi) ..... | K-1 |

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 1. Tahap-Tahap Data Mining .....                            | 7  |
| Gambar 2. Tiga Informasi Penting FP-Tree dan Tabel Header.....     | 10 |
| Gambar 3. Hasil Pembentukan FP-Tree Setelah Pembacaan TID 1 .....  | 12 |
| Gambar 4. Hasil Pembentukan FP-Tree Setelah Pembacaan TID 2 .....  | 12 |
| Gambar 5. Hasil Pembentukan FP-Tree Setelah Pembacaan TID 3 .....  | 12 |
| Gambar 6. Hasil Pembentukan FP-Tree Setelah Pembacaan TID 10 ..... | 13 |
| Gambar 7. Algoritma FP-Growth.....                                 | 15 |
| Gambar 8. Lintasan yang Mengandung Simpul E.....                   | 15 |
| Gambar 9. Lintasan yang Mengandung Simpul D .....                  | 16 |
| Gambar 10. Lintasan yang Mengandung Simpul C .....                 | 16 |
| Gambar 11. Lintasan yang Mengandung Simpul B .....                 | 16 |
| Gambar 12. Lintasan yang Mengandung Simpul A .....                 | 17 |
| Gambar 13. Lintasan yang Tidak Berakhir di E.....                  | 17 |
| Gambar 14. Support Count Simpul Diperbaharui .....                 | 18 |
| Gambar 15. Conditional FP-Tree Untuk E.....                        | 19 |
| Gambar 16. Pohon Prefix yang Berakhir di DE .....                  | 20 |
| Gambar 17. Pembentukan FP-Tree Berdasarkan Transaksi 1 .....       | 30 |
| Gambar 18. Pembentukan FP-Tree Berdasarkan Transaksi 2.....        | 30 |
| Gambar 19. Pembentukan FP-Tree Berdasarkan Transaksi 3.....        | 31 |
| Gambar 20. Pembentukan FP-Tree Berdasarkan Transaksi 185.....      | 31 |
| Gambar 21. Pembentukan FP-Tree Berdasarkan Transaksi 301.....      | 32 |
| Gambar 22. Lintasan yang Berakhir di H.....                        | 33 |
| Gambar 23. Lintasan yang Berakhir di B .....                       | 33 |
| Gambar 24. Lintasan yang Berakhir di F .....                       | 34 |
| Gambar 25. Lintasan yang Berakhir di D.....                        | 34 |
| Gambar 26. Lintasan Baru dengan Simpul H Dibuang .....             | 35 |
| Gambar 27. Lintasan Baru dengan Simpul D Dibuang.....              | 36 |
| Gambar 28. Conditional FP-Tree untuk H .....                       | 36 |
| Gambar 29. Conditional FP-Tree untuk D .....                       | 37 |
| Gambar 30. Deskripsi Sistem .....                                  | 41 |
| Gambar 31. Use Case Diagram .....                                  | 42 |

|  |    |
|--|----|
| Gambar 32. Sequence Diagram Mencari Asosiasi Buku .....                | 44 |
| Gambar 33. Sequence Diagram Mencari Informasi Buku .....               | 45 |
| Gambar 34. Sequence Diagram Mencari Daftar Buku.....                   | 46 |
| Gambar 35. Class Diagram.....  | 47 |
| Gambar 36. Desain Antar Muka Halaman Utama.....                        | 48 |
| Gambar 37. Desain Antar Muka Asosiasi Buku.....                        | 49 |
| Gambar 38. Desain Antar Muka Informasi Buku.....                       | 51 |
| Gambar 39. Desain Antar Muka Daftar Buku.....                          | 53 |
| Gambar 40. Entity Relational Diagram Tabel Buku .....                  | 55 |
| Gambar 41. Hasil Implementasi Halaman Utama.....                       | 59 |
| Gambar 42. Hasil Implementasi Halaman Utama Submenu Menu.....          | 59 |
| Gambar 43. Hasil Implementasi Halaman Utama Submenu Tentang.....       | 60 |
| Gambar 44. Hasil Implementasi Input Asosiasi Buku .....                | 61 |
| Gambar 45. Hasil Implementasi Input Asosiasi Buku Combo Box Kedua..... | 62 |
| Gambar 46. Hasil Implementasi Output Asosiasi Buku.....                | 63 |
| Gambar 47. Option Pane Tombol Keluar Asosiasi Buku.....                | 63 |
| Gambar 48. Hasil Implementasi Input Informasi Buku .....               | 64 |
| Gambar 49. Hasil Implementasi Output Informasi Buku.....               | 65 |
| Gambar 50. Option Pane Tombol Keluar Informasi Buku.....               | 65 |
| Gambar 51. Hasil Implementasi Input Daftar Buku.....                   | 66 |
| Gambar 52. Hasil Implementasi Output Daftar Buku .....                 | 67 |
| Gambar 53. Option Pane Tombol Keluar Daftar Buku .....                 | 67 |
| Gambar 54. Skenario Pengujian Sistem .....                             | 68 |

## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 1. Perbandingan dengan Penelitian Sebelumnya .....    | 4  |
| Tabel 2. Data Transaksi Mentah .....                        | 11 |
| Tabel 3. Frekuensi Kemunculan Tiap Item .....               | 11 |
| Tabel 4. Transaksi Data Setelah Diurutkan .....             | 11 |
| Tabel 5. Hasil Frequent Itemset .....                       | 20 |
| Tabel 6. Transformasi Data .....                            | 26 |
| Tabel 7. Penyeleksian Data .....                            | 27 |
| Tabel 8. Support Count 1-Item.....                          | 28 |
| Tabel 9. Kategori Buku dan Simbolnya .....                  | 28 |
| Tabel 10. 1-Item dan Support Count Setelah Diurutkan .....  | 29 |
| Tabel 11. Transaksi Setelah Diurutkan.....                  | 29 |
| Tabel 12. Hasil Frequent Itemset .....                      | 37 |
| Tabel 13. Itemset dan Confidence .....                      | 38 |
| Tabel 14. Itemset yang Memenuhi Minimum Confidence.....     | 39 |
| Tabel 15. Aturan Asosiasi .....                             | 39 |
| Tabel 16. Skenario Use Case Mencari Asosiasi Buku .....     | 43 |
| Tabel 17. Skenario Use Case Mencari Daftar Buku.....        | 43 |
| Tabel 18. Skenario Use Case Mencari Informasi Buku .....    | 43 |
| Tabel 19. Penjelasan Desain Antar Muka Halaman Utama .....  | 48 |
| Tabel 20. Penjelasan Desain Antar Muka Asosiasi Buku .....  | 50 |
| Tabel 21. Penjelasan Desain Antar Muka Informasi Buku ..... | 52 |
| Tabel 22. Penjelasan Desain Antar Muka Daftar Buku.....     | 54 |
| Tabel 23. Spesifikasi Tabel Buku .....                      | 55 |
| Tabel 24. Implementasi Basis Data Tabel Buku .....          | 58 |
| Tabel 25. Implementasi Kelas .....                          | 58 |
| Tabel 26. Hasil Pengujian Sistem.....                       | 69 |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Menurut Milburga (1986), perpustakaan adalah suatu unit kerja yang berupa tempat menyimpan koleksi bahan pustaka yang diatur secara sistematis dengan cara tertentu untuk dipergunakan secara berkesinambungan oleh pemakainya sebagai sumber informasi. Berdasarkan pengertian tersebut, koleksi bahan pustaka yang berada dalam perpustakaan harusnya diatur secara sistematis berdasarkan kategorinya masing-masing agar mempermudah pengurus maupun pengunjung perpustakaan mencari buku yang dikehendaki. Tidak semua perpustakaan memberikan label nama pada rak-rak bukunya sehingga pengunjung kesulitan mencari buku yang diinginkan. Contohnya perpustakaan Otorita Batam.

Perpustakaan Otorita Batam merupakan perpustakaan yang dapat dikunjungi oleh umum, baik masyarakat Batam maupun luar Batam. Buku-buku yang tersedia di perpustakaan tersebut dapat dibaca dan dipinjam oleh umum. Prosedur peminjaman buku di perpustakaan Otorita Batam masih dilakukan secara manual sehingga apabila daftar data peminjaman buku penuh, maka daftar peminjaman yang penuh tersebut tidak digunakan lagi. Hal ini menyebabkan terjadinya penumpukan data yang besar. Padahal daftar data peminjaman buku tersebut masih dapat diolah sehingga menghasilkan sesuatu yang bermanfaat. Salah satu manfaatnya adalah untuk penataan letak buku-buku di perpustakaan karena penataan buku di perpustakaan tersebut tidak berdasarkan kategori buku. Misalnya buku-buku sejarah bercampur dengan buku agama atau buku ilmu-ilmu terapan dalam rak pada baris pertama. Hal ini mengakibatkan pengunjung perpustakaan kesulitan menemukan buku yang diinginkan.

Menurut Han (2006), data mining adalah proses mengekstasi pola-pola yang menarik (implisit, tidak diketahui sebelumnya, dan berpotensi dapat dimanfaatkan) dari data yang berukuran besar. Berdasarkan definisi tersebut, data mining dapat dimanfaatkan untuk menganalisis tumpukan data-data yang tidak digunakan lagi dalam jumlah besar seperti data peminjaman buku di perpustakaan. Oleh karena itu, hal ini menjadi dasar untuk menemukan

pola peminjaman buku di perpustakaan menggunakan algoritma *FP-growth* dari metode asosiasi karena algoritma tersebut dapat menganalisis data lebih cepat dibandingkan algoritma apriori. Pola peminjaman buku yang dihasilkan melalui analisis data tersebut berupa aturan asosiasi yang dapat dijadikan acuan tata letak buku di perpustakaan agar pengunjung lebih mudah menemukan buku yang diinginkan seperti yang terlihat pada lampiran K.

Apabila dibandingkan dengan aturan asosiasi yang disajikan melalui tulisan pada selebar kertas maupun buku, aturan asosiasi yang disajikan melalui sistem jauh lebih menarik perhatian pengunjung maupun pengurus perpustakaan. Tampilan pada sistem lebih mempersingkat aturan asosiasi dibandingkan aturan asosiasi yang tertulis panjang pada kertas maupun buku. Oleh karena itu, hal ini menjadi dasar untuk menerapkan aturan asosiasi tersebut ke dalam sistem guna mempermudah pengguna mengetahui hubungan asosiasi buku-buku di perpustakaan Otorita Batam.

## **1.2 Rumusan Masalah**

Beberapa rumusan masalah dalam penulisan laporan tugas akhir ini adalah sebagai berikut:

1. Bagaimana mengetahui buku yang sering di pinjam secara bersamaan?
2. Bagaimana memanfaatkan data peminjaman buku perpustakaan Otorita Batam?
3. Bagaimana membangun sistem yang menerapkan aturan asosiasi yang diperoleh?

## **1.3 Batasan Masalah**

Ruang lingkup data yang di analisa adalah sebagai berikut:

1. Hanya menganalisis data peminjaman buku di perpustakaan Otorita Batam pada tahun 2013.
2. Menggunakan metode asosiasi dan algoritma *FP-Growth*.
3. Bahasa pemrograman yang digunakan untuk membangun sistem adalah *Java Netbeans*.

## **1.4 Tujuan Penelitian**

Tujuan penelitian yang terdapat dalam penulisan laporan tugas akhir ini antara lain:

1. Mencari aturan asosiasi dari data peminjaman buku perpustakaan Otorita Batam menggunakan algoritma *FP-Growth*.

2. Mengimplementasikan data peminjaman buku di perpustakaan menggunakan algoritma *FP-growth* sebagai acuan tata letak buku di perpustakaan.
3. Aturan asosiasi yang diperoleh diterapkan ke dalam sistem berbasis desktop guna mempermudah pengunjung menemukan buku yang diinginkan.

## **1.5 Sistematika Penulisan**

### **BAB I PENDAHULUAN**

Pada bagian ini, laporan tugas akhir membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan penulisan, serta sistematika penulisan laporan tugas akhir ini.

### **BAB II LANDASAN TEORI**

Tinjauan pustaka dalam laporan tugas akhir ini membahas tentang perbandingan antara penelitian sekarang dan penelitian sebelumnya, teori-teori yang berhubungan dengan data mining serta perangkat-perangkat lunak yang digunakan.

### **BAB III ANALISIS DAN PERANCANGAN**

Pada bagian ini membahas tentang proses pengolahan data dari data mentah hingga pengolahan data menggunakan algoritma yang akan digunakan, dan perancangan sistem yang akan dibuat mulai dari *usecase*, *sequence*, diagram kelas, hingga rancangan *interface* sementara.

### **BAB IV HASIL DAN PEMBAHASAN**

Pada bagian ini, laporan tugas akhir berisi hasil implementasi dan pembahasannya dari analisis dan perancangan konsep yang telah dilakukan pada sebelumnya.

### **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dari pembahasan pada perancangan awal serta analisa yang diperoleh. Untuk lebih meningkatkan mutu dari sistem yang telah dibuat maka diberikan saran-saran untuk perbaikan dan penyempurnaan sistem.

### **DAFTAR PUSTAKA**

Pada bab ini berisi tentang referensi-referensi yang telah dipakai oleh penulis sebagai acuan dan penunjang yang mendukung penyelesaian tugas akhir ini.

## BAB II

### LANDASAN TEORI

#### 2.1 Penelitian Sebelumnya

Adapun perbandingan antara penelitian analisis data yang dibuat dengan penelitian analisis data sebelumnya terkait Tugas Akhir ini dapat dilihat seperti pada tabel 1.

**Tabel 1. Perbandingan dengan Penelitian Sebelumnya**

| Perbandingan        | Penelitian 1                      | Penelitian 2                                 | Penelitian 3           | Penelitian Sekarang               |
|---------------------|-----------------------------------|--|------------------------|-----------------------------------|
| Objek yang diteliti | Data peminjaman buku perpustakaan | Data transaksi penjualan <i>dealer</i> mobil | Keranjang pasar        | Data peminjaman buku perpustakaan |
| Studi kasus         | Politeknik Negeri Batam           | <i>Dealer</i> mobil di Malang                | Salah satu Supermarket | Perpus. Otorita Batam             |
| Algoritma           | Apriori                           | FP-Growth                                    | FP-Growth              | FP-Growth                         |
| Software pengembang | Visual Basic                      | Netbeans (Java)                              | -                      | Netbeans (Java)                   |

Penjelasan lebih lanjut mengenai penelitian analisis data sebelumnya yang terdapat pada tabel 1 yaitu:

1. Analisis Data Pola Peminjaman Buku Perpustakaan Menggunakan Metode Asosiasi (Yolanda dan Annisa, 2012). Pada penelitian tersebut dilakukan analisis data untuk mengetahui buku-buku yang saling berhubungan melalui pola peminjaman buku perpustakaan Politeknik Negeri Batam. Pada penelitian tersebut menggunakan salah satu algoritma dari metode asosiasi yaitu algoritma Apriori. Kekurangan dalam penelitian tersebut adalah membutuhkan waktu yang lama dalam menemukan *frequent itemset* karena algoritma apriori menscan *database* lebih dari 2 kali. Sedangkan untuk Analisis Data Pola Peminjaman Buku Menggunakan Algoritma FP-Growth memiliki kelebihan dibandingkan penelitian sebelumnya dari segi algoritmanya. Algoritma FP-Growth hanya menscan *database* paling banyak 2 kali sehingga untuk menemukan *frequent itemset* membutuhkan waktu yang lebih cepat.
2. Penerapan Metode *Association Rule* dengan Algoritma FP-Growth Untuk Mengetahui Karakteristik Pembeli Mobil (Putra, dkk, 2012). Pada penelitian tersebut dilakukan analisa data untuk mengetahui karakteristik pembeli mobil berdasarkan data transaksi penjualan *dealer* mobil. Sehingga sistem ini nantinya mampu menampilkan bagaimana

karakteristik pembeli dengan variabel jenis kelamin pembeli dan rentang usia pembeli melalui masukan data dari pengguna berupa jenis mobil. Selain itu, sistem ini juga menampilkan keterangan dari *rule*, *support*, *confidence*, dan *lift ratio* dari masing-masing aturan yang diperoleh dari perhitungan algoritma *FP-Growth*. Pada penelitian tersebut menggunakan salah satu algoritma dari metode asosiasi yaitu algoritma *FP-Growth*. Kelemahan yang terdapat dalam penelitian tersebut terletak pada pengolahan datanya yang tidak dijelaskan bagaimana gambaran *fp-tree* dan bagaimana langkah-langkah menyelesaikan kasus menggunakan algoritma *fp-growth* hingga menghasilkan aturan asosiasi yang diterapkan ke sistem terkait. Kelebihan Analisis Data Pola Peminjaman Buku Menggunakan Algoritma *FP-Growth* dibandingkan penelitian sebelumnya yaitu menjelaskan proses pengolahan datanya mulai pembentukan *fp-tree* hingga hasil aturan asosiasi yang diterapkan pada sistem nantinya.

3. Asosiasi Data Mining Menggunakan Algoritma *FP-Growth* Untuk *Market Basket Analysis* (Fatihatul, dkk, 2012). Pada penelitian tersebut dilakukan analisa data untuk mengetahui perilaku pembelian dari konsumen yang dapat dimanfaatkan untuk strategi pemasaran. Penelitian tersebut tidak menerapkannya dalam bentuk sistem namun penelitian tersebut menggunakan *software* WEKA 3.6.4 hanya untuk mencari aturan yang dibentuk dari beberapa transaksi pembelian yang terjadi di *supermarket* dalam suatu periode tertentu. Berdasarkan aturan yang dihasilkan tersebut, maka strategi pemasaran yang dapat digunakan oleh *supermarket* tersebut adalah menempatkan *item-item* yang saling *frequent* secara berdekatan. Pada penelitian tersebut menggunakan salah satu algoritma dari metode asosiasi yaitu algoritma *FP-Growth*. Pengolahan data pada penelitian sebelumnya menggunakan WEKA sehingga format *file* di *Excel* harus *.csv* terlebih dahulu. Meskipun melalui WEKA dapat diketahui *rule association* dengan cepat dan mudah, namun jika dilakukan secara manual seperti pada Analisis Data Pola Peminjaman Buku Menggunakan Algoritma *FP-Growth* akan lebih menyenangkan dan mudah dimengerti perbedaan antara algoritma *FP-growth* dengan algoritma lainnya.

## 2.2 Analisis Data

Menurut Moleong (2007), analisis data adalah proses mengatur urutan data, mengorganisasikannya ke dalam suatu pola, kategori, dan satuan uraian dasar. Berdasarkan definisi tersebut, analisis data diperlukan untuk mendapatkan informasi penting yang dapat dimanfaatkan untuk suatu tujuan tertentu. Data-data mentah yang akan dianalisis dapat

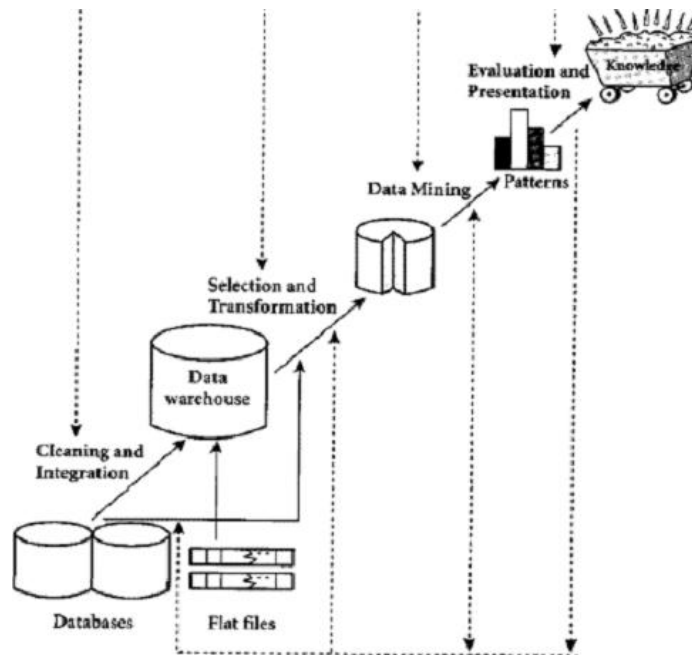
berupa data apa saja, baik data kuantitatif maupun data kualitatif. Data kuantitatif biasanya berupa data yang berupa angka-angka, misalnya data sensus penduduk, data anggaran biaya kuliah, dan lain sebagainya. Sementara data kualitatif biasanya berupa data-data yang diperoleh melalui penelitian (studi lapangan), misalnya data peminjaman buku.

Data peminjaman buku di perpustakaan Otorita Batam berupa data mentah yang terdiri dari beberapa variabel yaitu nama peminjam, buku yang dipinjam, tanggal buku di pinjam, dan tanggal buku di kembalikan. Prosedur peminjaman buku di perpustakaan Otorita Batam masih dilakukan secara manual yaitu dengan mengisi daftar peminjaman buku yang telah disediakan di perpustakaan tersebut. Kelemahan dari prosedur peminjaman buku yang telah ditetapkan tersebut adalah terjadinya penumpukan data yang cukup besar karena apabila arsip berisi daftar data peminjaman buku penuh, arsip tersebut hanya disimpan dan tidak dihiraukan lagi. Sementara di sisi lain, data peminjaman buku tersebut masih dapat di analisis untuk menemukan pola yang bermanfaat bagi pengurus maupun pengunjung perpustakaan menggunakan data mining.

### **2.3 Data Mining**

Beberapa definisi data mining menurut para ahli seperti menurut Kusri (2009), data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database*. Sama halnya menurut Han (2006), data mining adalah proses mengekstasi pola-pola yang menarik (implisit, tidak diketahui sebelumnya, dan berpotensi dapat dimanfaatkan) dari data yang berukuran besar. Selain itu menurut Turban (2005), data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar.

Data mining seharusnya dipahami sebagai suatu proses, yang memiliki tahapan-tahapan tertentu dan juga ada umpan balik dari setiap tahapan ke tahapan sebelumnya. Pada umumnya proses data mining berjalan interaktif karena tidak jarang hasil data mining pada awalnya tidak sesuai dengan harapan analisisnya sehingga perlu dilakukan desain ulang prosesnya. Menurut Han (2006), data mining memiliki beberapa tahapan seperti yang terlihat pada gambar 1.



**Gambar 1. Tahap-Tahap Data Mining**

Tahap-tahap data mining berdasarkan gambar 1 dijelaskan sebagai berikut:

1. Pembersihan data (*data cleaning*), digunakan untuk membuang data yang tidak konsisten dan *noise*.
2. Integrasi data (*data integration*), menggabungkan data yang berasal dari sumber berbeda ke suatu *database* yang sama.
3. Transformasi data (*data transformation*), untuk menentukan kualitas dari hasil data *mining* sehingga data diubah menjadi bentuk sesuai untuk di-*mining*.
4. Aplikasi teknik data mining, salah satu bagian dari proses data mining.
5. Evaluasi pola yang ditemukan, hasil dari teknik data mining berupa pola-pola yang khas maupun model prediksi dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai.
6. Presentasi pengetahuan, untuk menghasilkan aksi tahap terakhir dari proses data mining adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisa yang didapat.

Menurut Larose (2005), data mining terdiri dari beberapa metode berdasarkan pekerjaan atau tugas yang dapat dilakukan, yaitu:

1. Deskripsi

Terkadang peneliti dan analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Deskripsi dari

pola kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

## 2. Estimasi

Estimasi hampir sama dengan klasifikasi, namun variabel target estimasi lebih ke arah numerik dari pada ke arah kategori. Model dibangun menggunakan baris data (*record*) lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi.

## 3. Prediksi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

## 4. *Clustering*

*Clustering* merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas obyek-obyek yang memiliki kemiripan. Berbeda dengan klasifikasi, pada pengklasteran tidak ada variabel target. Pengklasteran tidak melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target, akan tetapi, algoritma pengklasteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.

## 5. Asosiasi

Tugas asosiasi dalam data mining adalah untuk menemukan atribut yang muncul dalam satu waktu. Metode asosiasi digunakan untuk mengenali kelakuan dari kejadian-kejadian khusus atau proses dimana link asosiasi muncul pada setiap kejadian.

### **2.3.1 Metode Asosiasi**

Menurut Kusriani (2009), analisis asosiasi atau *association rule* mining adalah teknik data mining untuk menemukan aturan asosiasi antara suatu kombinasi *item*. *Interestingness measure* yang dapat digunakan dalam data mining adalah:

1. *Support*, suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *item* atau *itemset* dari keseluruhan transaksi.

2. *Confidence*, suatu ukuran yang menunjukkan hubungan antar dua *item* secara *conditional* (berdasarkan suatu kondisi tertentu).

Metodologi dasar analisis asosiasi terbagi menjadi 2 tahap, yakni melakukan analisa pola frekuensi tinggi (*frequent pattern*) dan berikutnya adalah proses pembentukan aturan asosiasi.

1. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support* sebuah *item* diperoleh dengan memakai rumus berikut:

$$Support(A) = \frac{J_{u \quad h \quad t_i \quad m \quad A}}{T \quad t}$$

Sedangkan nilai dari *support 2 item* diperoleh melalui rumus:

$$Support(A,B) = (A \cap B) \frac{J_{u \quad h \quad t_i \quad m \quad A \quad d \quad B}}{T \quad t_i}$$

2. Pembentukan aturan asosiasi

Setelah semua pola frekuensi tinggi ditemukan, selanjutnya dicari aturan asosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiasi “jika A maka B”. Nilai *confidence* dari aturan “jika A maka B” diperoleh dari rumus:

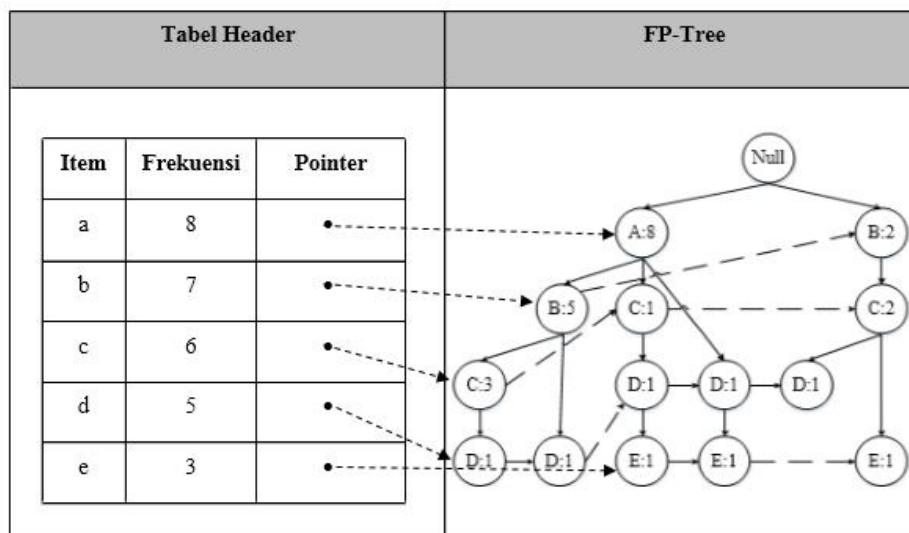
$$Confidence = P(B|A) = \frac{J_{u \quad h \quad t_i \quad s \quad i \quad m \quad A \quad d \quad B}}{J_{u \quad h \quad t_i \quad m \quad A}}$$

Analisis data pola peminjaman buku perpustakaan menggunakan metode asosiasi karena metode asosiasi dapat digunakan untuk mencari pola hubungan dalam data. Pola hubungan dalam data yang dimaksud adalah jika buku sejarah dan buku ekonomi sering dipinjam secara bersamaan, maka buku sejarah dan buku ekonomi akan diletakkan secara berdampingan, begitu pula dengan pola lainnya. Metode asosiasi memiliki lebih dari satu algoritma untuk menganalisis data yaitu algoritma Apriori, algoritma *FP-Growth*, algoritma *Hash Based*, dan lain sebagainya.

### 2.3.2 Algoritma FP-Growth

Menurut Samuel (2008), algoritma *FP-growth* merupakan pengembangan dari algoritma Apriori sehingga kekurangan dari algoritma Apriori diperbaiki oleh algoritma *FP-growth*. *Frequent Pattern Growth* (*FP-growth*) adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data. Karakteristik algoritma *FP-growth* adalah struktur data yang digunakan adalah *tree* yang disebut dengan *FP-tree*. Jika menggunakan *FP-tree*, algoritma *FP-growth* dapat langsung mengekstrak *frequent itemset* dari *FP-tree*. *FP-tree* adalah sebuah pohon dengan definisi sebagai berikut:

- FP-tree* dibentuk oleh sebuah akar yang diberi label *null*, sekumpulan sub-*tree* yang beranggotakan *item-item* tertentu, dan sebuah tabel *frequent header*.
- Setiap simpul dalam *FP-tree* mengandung tiga informasi penting, yaitu label *item*, menginformasikan jenis *item* yang direpresentasikan simpul tersebut, *support count*, merepresentasikan jumlah lintasan transaksi yang melalui simpul tersebut, dan *pointer* penghubung yang menghubungkan simpul-simpul dengan label *item* sama antar-lintasan, ditandai dengan garis panah putus-putus. Ketiga informasi penting tersebut dapat dilihat seperti pada gambar 2.



Gambar 2. Tiga Informasi Penting FP-Tree dan Tabel Header

Berdasarkan gambar 2, dapat dijelaskan bahwa dalam pembangunan *FP-tree* terdiri dari item dan frekuensi untuk setiap simpulnya. Misalnya dapat dilihat pada gambar 2 bahwa item a dengan frekuensi 8 simpul *FP-treenya* menjadi a:8.

Menurut Samuel (2008), misalkan untuk contoh soal 1 diberikan tabel data transaksi seperti pada tabel 2, dengan minimum *support count*=2.

**Tabel 2. Data Transaksi Mentah**

| TID | Transaksi     |
|-----|---------------|
| 1   | a, b          |
| 2   | b, c, d, g, h |
| 3   | a, c, d,e     |
| 4   | a, d, e       |
| 5   | a, b, z, c    |
| 6   | a, b, c, d    |
| 7   | a, r          |
| 8   | a, b, c       |
| 9   | a, b, d       |
| 10  | b, c, e       |

Frekuensi kemunculan tiap *item* dapat dilihat seperti pada tabel 3.

**Tabel 3. Frekuensi Kemunculan Tiap Item**

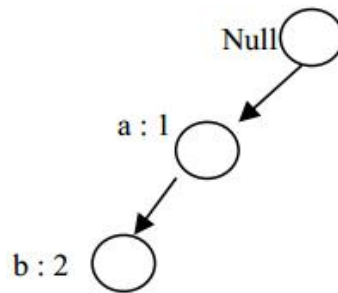
| Item | Frekuensi |
|------|-----------|
| a    | 8         |
| b    | 7         |
| c    | 6         |
| d    | 5         |
| e    | 3         |
| r    | 1         |
| z    | 1         |
| g    | 1         |
| h    | 1         |

Setelah dilakukan pemindaian pertama, didapatkan *item* yang memiliki *support count* lebih besar atau sama dengan 2 yaitu *item* a, b, c, d, dan e. Kelima *item* inilah yang akan berpengaruh pada *FP-tree*, selebihnya (r, z, g, h) dapat dibuang karena tidak berpengaruh signifikan. Selanjutnya kemunculan *item* yang *frequent* dalam setiap transaksi, diurutkan berdasarkan yang frekuensinya paling tinggi seperti terlihat pada tabel 4.

**Tabel 4. Transaksi Data Setelah Diurutkan**

| TID | Transaksi    |
|-----|--------------|
| 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}    |

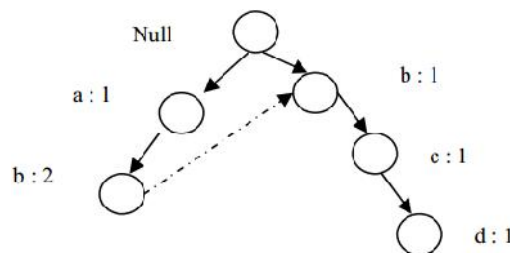
Gambar 3 memberikan ilustrasi mengenai pembentukan FP-tree setelah pembacaan TID 1.



**Gambar 3. Hasil Pembentukan FP-Tree Setelah Pembacaan TID 1**

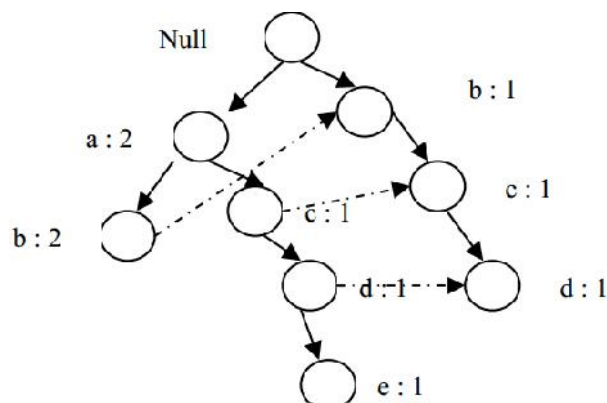
Berdasarkan gambar 3, dapat dilihat bahwa hasil pembentukan FP-tree dibangun berdasarkan transaksi 1 yaitu {a,b}. Sehingga a:1 dan b:2 karena satu jalur untuk 1 transaksi. Pembangunan FP-tree selalu dibangun dengan simpul null.

Selanjutnya hasil pembentukan FP-tree setelah pembacaan TID 2 dan TID 3 terlihat seperti pada gambar 4 dan gambar 5.



**Gambar 4. Hasil Pembentukan FP-Tree Setelah Pembacaan TID 2**

Berdasarkan gambar 4, dapat dijelaskan bahwa FP-tree untuk transaksi kedua dibuat cabang baru karena berbeda simpul awalnya dengan transaksi 1 sehingga pada cabang yang simpul atasnya a tidak dapat dilalui.

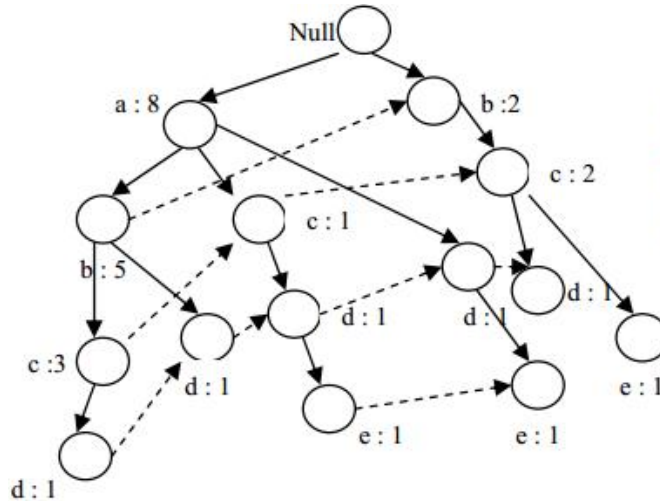


**Gambar 5. Hasil Pembentukan FP-Tree Setelah Pembacaan TID 3**

Berdasarkan pembentukan FP-tree pada gambar 5, simpul atas a berubah menjadi 2 karena dilalui 2 kali untuk transaksi ketiga. Selain itu, simpul a memiliki cabang baru untuk

transaksi 3 yaitu {a,c,d,e} karena simpul setelah a pada transaksi sebelumnya adalah b bukan c sehingga tidak dapat dilanjutkan.

Setelah pembacaan TID 10 hasil pembentukan FP-tree-nya terlihat seperti pada gambar 6.



Gambar 6. Hasil Pembentukan FP-Tree Setelah Pembacaan TID 10

Berdasarkan gambar 6, dapat dijelaskan bahwa hasil pembentukan FP-tree pada transaksi 10 terdiri dari beberapa cabang dimana cabang-cabang tersebut memiliki banyak jalur yang tidak hanya dilalui 1 atau 2 kali. Misalnya *item* a yang memiliki frekuensi 8, artinya *item* ini dilalui sebanyak 8 kali. Selain itu, *item* b pada jalur {a,b,c,d} memiliki 5 frekuensi yang menandakan bahwa *item* tersebut dilalui sebanyak 5 kali.

Setiap simpul pada FP-tree mengandung nama sebuah *item* dan *counter support* yang berfungsi untuk menghitung frekuensi kemunculan *item* tersebut dalam tiap lintasan transaksi. FP-tree yang merepresentasikan data transaksi pada tabel 2 didapatkan dengan cara sebagai berikut:

1. Kumpulan data dipindai pertama kali untuk menentukan *support count* dari setiap *item*. *Item* yang tidak *frequent* dibuang, sedangkan *frequent item* dimasukkan dan disusun dengan urutan menurun, seperti yang terlihat pada tabel 2.
2. Pemindaian kedua, yaitu pembacaan TID pertama {a,b} akan membuat simpul a dan b, sehingga terbentuk lintasan transaksi *Null* a b. *Support count* dari setiap simpul bernilai awal 1.
3. Setelah pembacaan transaksi kedua {b,c,d}, terbentuk lintasan kedua yaitu *Null* b c d. *Support count* masing-masing *count* juga bernilai awal 1. Walaupun b

ada pada transaksi pertama, namun karena *prefix* transaksinya tidak sama, maka transaksi kedua ini tidak bisa dimampatkan dalam satu lintasan.

4. Transaksi keempat memiliki *prefix* transaksi yang sama dengan transaksi pertama, yaitu *a*, maka lintasan transaksi ketiga dapat ditimpakan di *a*, sambil menambah *support count* dari *a*, dan selanjutnya membuat lintasan baru sesuai dengan transaksi ketiga seperti yang terlihat pada gambar 5.
5. Proses ini dilanjutkan sampai *FP-tree* berhasil dibangun berdasarkan tabel data transaksi yang diberikan.

Setelah tahap pembangunan *FP-tree* dari sekumpulan data transaksi, akan diterapkan algoritma *FP-growth* untuk mencari *frequent itemset* yang signifikan. Menurut Han (2006), metode *FP-growth* dapat dibagi menjadi 3 tahapan, yaitu:

1. Tahap pembangkitan *conditional patterns base*

*Conditional patterns base* adalah *sub database* yang berisi *prefix path* (lintasan *prefix*) dan *suffix pattern* (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui *FP-tree* yang telah dibangun sebelumnya.

2. Tahap 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*.

3. Tahap pencarian *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 rekursif.

Ketiga tahap tersebut merupakan langkah yang akan dilakukan untuk mendapat *frequent itemset*, yang dapat dilihat pada algoritma seperti terlihat pada gambar 7.

```

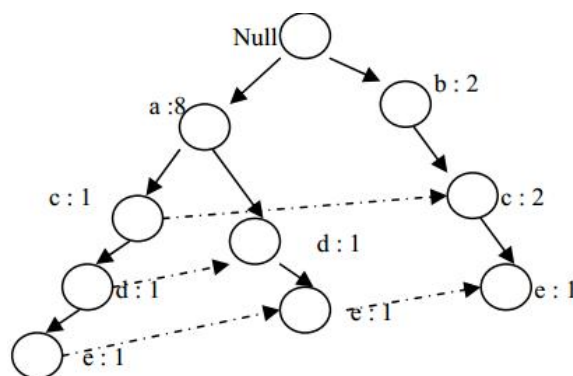
Input: FP-tree Tree
Output: Rt sekumpulan lengkap pola frequent
Method: FP-growth(Tree, null)
Procedure: FP-growth(Tree, α)
(
01:if Tree mengandung single path P;
02:then untuk tiap kombinasi (dinotasikan β) dari node-node dalam path P do
03:bangkitkan pola β α dengan support = minimum support dari node-node dalam β;
04:else untuk tiap a1 dalam header dari Tree do {
05:bangkitkan pola
06:bangun β = a1 α dengan support = a1.support
07:if Tree β = ∅
08:then panggil FP-growth(Tree, β) }
)

```

Gambar 7. Algoritma FP-Growth

Berdasarkan gambar 7, dapat dijelaskan bahwa algoritma *FP-growth* memiliki *input* *FP-tree* dan *output* berupa sekumpulan lengkap pola *frequent*. Prosedur dari algoritma *FP-growth* ini adalah langkah-langkah dari pengerjaan algoritma *FP-growth*.

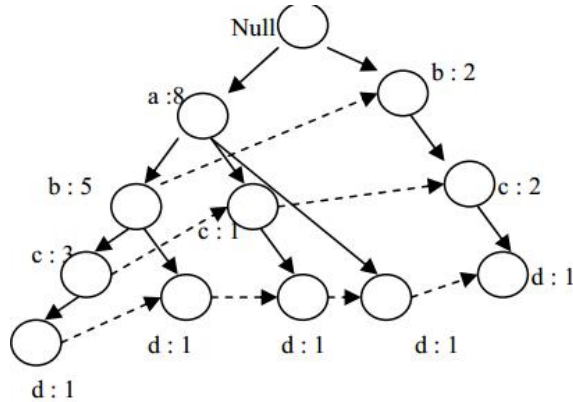
Algoritma *FP-growth* akan coba diterapkan pada contoh soal 1. Untuk menemukan *frequent itemset* dari contoh soal 1, maka perlu ditentukan *FP-tree* dengan lintasan yang berakhir dengan *support count* terkecil, yaitu e. Berturut-turut ditentukan juga yang berakhir di d, c, b, dan a. Proses pembentukan dapat dilihat seperti pada gambar 8 hingga gambar 12.



Gambar 8. Lintasan yang Mengandung Simpul E

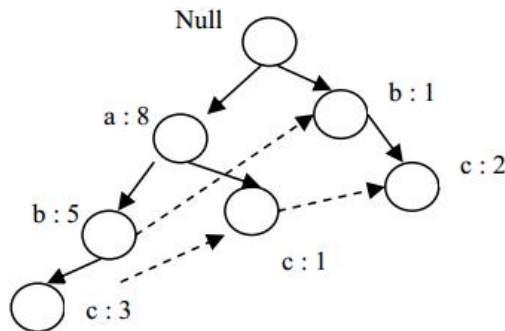
Berdasarkan gambar 8, dapat dijelaskan bahwa dari pembangunan *FP-tree* terakhir yaitu pada transaksi 10, dibangun kembali *FP-tree* dengan lintasan terakhir (lintasan dengan *support count* terkecil). Adapun lintasan *FP-tree* yang memiliki *support count* terkecil yaitu

yang mengandung simpul e. Berdasarkan gambar 8, simpul e terdiri dari 3 cabang yang mengandung simpul e yaitu {a,c,d,e},{a,d,e}, dan {b,c,e}.



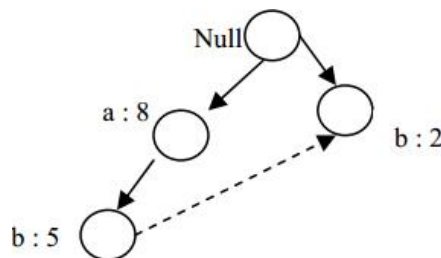
**Gambar 9. Lintasan yang Mengandung Simpul D**

Berdasarkan gambar 9, dapat dijelaskan simpul d terdiri dari 5 cabang yang mengandung simpul d yaitu {a,b,c,d}, {a,b,d}, {a,c,d}, {a,d}, {b,c,d}. Cabang-cabang yang mengandung d diperoleh tetap dari pembangunan FP-tree pada transaksi kesepuluh tadi bukan merupakan lanjutan dari lintasan yang mengandung simpul e sebelumnya.



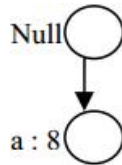
**Gambar 10. Lintasan yang Mengandung Simpul C**

Berdasarkan gambar 10, dapat dijelaskan simpul c terdiri dari 3 cabang yang mengandung simpul d yaitu {a,b,c}, {a,c}, dan {b,c}. Cabang-cabang yang mengandung c diperoleh tetap dari pembangunan FP-tree pada transaksi kesepuluh tadi bukan merupakan lanjutan dari lintasan yang mengandung simpul d sebelumnya.



**Gambar 11. Lintasan yang Mengandung Simpul B**

Berdasarkan gambar 11, dapat dijelaskan simpul b terdiri dari 2 cabang yang mengandung simpul b yaitu {a,b} dan {b}. Cabang-cabang yang mengandung b diperoleh tetap dari pembangunan FP-tree pada transaksi kesepuluh tadi bukan merupakan lanjutan dari lintasan yang mengandung simpul c sebelumnya.

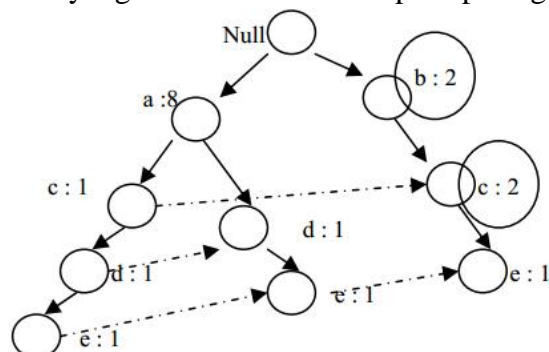


Gambar 12. Lintasan yang Mengandung Simpul A

Berdasarkan gambar 12, dapat dijelaskan simpul a terdiri dari 1 cabang saja yang mengandung simpul a yaitu {a}. Cabang yang mengandung a diperoleh tetap dari pembangunan FP-tree pada transaksi kesepuluh tadi bukan merupakan lanjutan dari lintasan yang mengandung simpul b sebelumnya.

Algoritma FP-growth menemukan *frequent itemset* yang berakhiran *suffix* tertentu dengan menggunakan metode *divide and conquer* untuk memecah *problem* menjadi *subproblem* yang lebih kecil. Contohnya, jika ingin menemukan semua *frequent itemset* yang berakhiran e. Hal pertama yang harus dilakukan ialah mengecek apakah *support count* dari e memenuhi minimum *support count*=2. Karena *support count* dari e adalah 3, dan  $3 \geq 2$ , maka e adalah *item* yang *frequent*.

Setelah mengetahui bahwa *item* e adalah *item* yang *frequent*, maka *subproblem* selanjutnya adalah menemukan *frequent itemset* dengan akhiran de, ce, be, dan ae. Dengan menggabungkan seluruh solusi dari *subproblem* yang ada, maka himpunan semua *frequent itemset* yang berakhiran *item* e akan didapatkan. Untuk lebih jelas dapat dilihat contoh menemukan *frequent itemset* yang berakhiran *item* e seperti pada gambar 13.

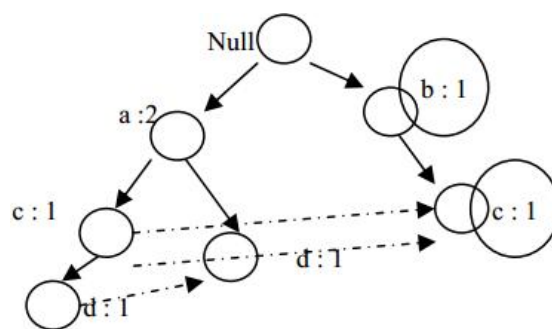


Gambar 13. Lintasan yang Tidak Berakhir di E

Berdasarkan gambar 13, dapat dijelaskan bahwa terdapat lintasan yang tidak berakhir di e karena lintasan pada jalur *FP-tree* tersebut terdiri dari 2 lintasan yaitu {b,c,e} dan {b,c}. Lintasan yang tidak berakhir di e yaitu {b,c} karena berakhir di c bukan e. Lintasan yang tidak berakhir di e ditandai dengan garis melingkar yang terlihat pada gambar 13 yaitu lintasan {b,c}.

Adapun langkah-langkah untuk menemukan *frequent itemset* yang berakhir *item* e yaitu:

1. Langkah pertama yang dilakukan adalah membangun sebuah upapohon *FP-tree* dengan hanya menyertakan lintasan yang berakhir di e.
2. *Support count* dari *item* e dihitung dan dibandingkan dengan minimum *support count*=3. Karena memenuhi, maka {e} termasuk *frequent itemset*, karena *support count*=2.
3. Karena *item* e *frequent*, maka perlu dipecahkan *subproblem* untuk menemukan *frequent itemset* yang berakhir dengan de, ce, be, dan ae. Sebelum memecahkan *subproblem* ini, maka upapohon *FP-tree* tersebut harus diubah terlebih dahulu menjadi *conditional FP-tree*. *Conditional FP-tree* mirip dengan *FP-tree* biasa, namun *conditional FP-tree* dimaksudkan untuk mencari *frequent itemset* yang berakhir *item* tertentu.
4. *Conditional FP-tree* dapat dibentuk dengan cara:
  - a. Setiap lintasan yang tidak mengandung e di buang. Pada contoh, lintasan terkanan, terdapat lintasan yang tidak mengandung e, yaitu *null* b c. Lintasan ini dapat dibuang dengan cara mengurangi *support count* menjadi 1, sehingga lintasan tersebut hanya mengandung transaksi {b,c,e}, seperti pada gambar 14.

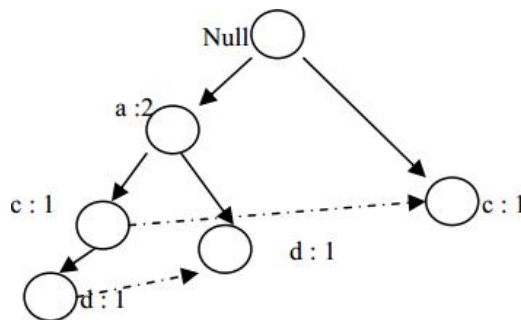


**Gambar 14. Support Count Simpul Diperbaharui**

Berdasarkan gambar 14, dapat dijelaskan bahwa lintasan {b,c} yang tidak berakhir di e sudah dihapus sehingga terlihat frekuensinya juga berubah menjadi 1. Sedangkan lintasan lainnya juga diperbaharui berdasarkan banyaknya lintasan dilalui. Seperti pada lintasan {a,c,d,e}, terlihat pada gambar 14 bahwa simpul e

telah dihapus namun frekuensi a yang tadinya berjumlah 8 diganti menjadi 2 karena hanya 2 kali dilalui yaitu lintasan (a,c,d) dan lintasan {a,d}.

- b. Setelah semua lintasan berakhir di e, maka simpul e dapat dibuang, karena setiap nilai *support count* pada simpul orang tuanya telah mencerminkan transaksi yang berakhir di e. *Subproblem* selanjutnya yang harus dipecahkan adalah mencari lintasan *frequent itemset* yang berakhir di de, ce, be, dan ae.
- c. Karena nilai *support count* dari b adalah 1, yang berarti transaksi yang mengandung b dan e hanya 1 transaksi, maka berdasarkan prinsip *anti-monotone heuristic*, simpul b dan lintasan yang mengandung be dapat dibuang, karena jika *item* b tidak *frequent*, maka setiap transaksi yang berakhiran be juga tidak *frequent*. Terbentuk *Conditional FP-tree* untuk e, seperti pada gambar 15.

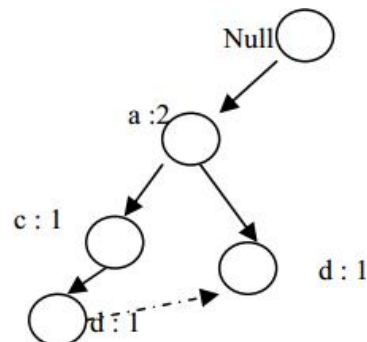


Gambar 15. Conditional FP-Tree Untuk E

Berdasarkan gambar 15, dapat dijelaskan bahwa *conditional FP-tree* terbentuk karena *item-item* yang tidak *frequent* dari gambar 14 dihapuskan sehingga *conditional FP-tree* untuk simpul e seperti yang terlihat pada gambar 15. Adapun *item-item* yang tidak *frequent* dari gambar 14 yaitu *item* b. Maka pada gambar 15, *conditional FP-tree* terdiri dari lintasan {a,c,d,e}, {a,d,e}, dan {c,e}.

- 5. *FP-tree* menggunakan *Conditional FP-tree* untuk membangun pohon lintasan *prefix* untuk menemukan *frequent itemset* yang berakhir dengan pasangan *item* de, ce, dan ae.

6. Untuk lintasan *prefix* *de*, yang terbentuk dari *Conditional FP-tree* untuk *item e* dapat dilihat seperti pada gambar 16.



Gambar 16. Pohon Prefix yang Berakhir di DE

Berdasarkan gambar 16, dapat dijelaskan bahwa lintasan yang berakhir di  $\{d,e\}$  sama dengan *conditional FP-tree* dengan lintasan-lintasan yang *frequent* yaitu  $\{a,c,d,e\}$  dan  $\{a,d,e\}$  namun tidak ada  $\{c,e\}$  karena  $\{c,e\}$  tidak berakhir di  $\{d,e\}$ .

7. Dengan menjumlahkan *support count* dari *d*, yang tidak lain adalah jumlah *frequent itemset* yang berakhir di *de*, didapat bahwa  $\{d,e\}$  juga termasuk dalam *frequent itemset*.
8. Selanjutnya, algoritma *FP-tree* akan mengulangi langkah yang sama dengan langkah ketiga, sehingga didapatkan *conditional FP-tree* untuk *de* hanya berisi satu daun, yaitu *a*, dengan *support count*=2. Sehingga  $\{a,d,e\}$  termasuk dalam *frequent itemset*.
9. *Subproblem* berikutnya yaitu dengan menemukan *frequent itemset* yang berakhir dengan *ce*. Didapat  $\{c,e\}$  juga merupakan *frequent itemset*. Begitupula dengan  $\{a,e\}$ .

Setelah memeriksa *frequent itemset* untuk beberapa akhiran (*suffix*), maka didapat hasil yang dirangkum dalam tabel 5.

Tabel 5. Hasil Frequent Itemset

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

Dengan metode *divide and conquer*, maka pada setiap langkah rekursif, algoritma *FP-growth* akan membangun sebuah *conditional FP-tree* baru yang telah diperbaharui nilai *support count*, dan membuang lintasan yang mengandung *item-item* yang tidak *frequent* lagi.

Analisis data pola peminjaman buku ini akan menggunakan algoritma *FP-growth* karena algoritma ini menscan *database* paling banyak 2 kali sedangkan algoritma yang umumnya digunakan (algoritma Apriori) menscan *database* setiap kali dilakukannya iterasi. Selain itu, algoritma *FP-growth* lebih cepat dibandingkan algoritma Apriori karena pada algoritma *FP-growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemset* sehingga tidak diperlukan *generate candidate* untuk mendapatkan *frequent itemsets*.

## **2.4 Perangkat-Perangkat Lunak yang Mendukung dalam Analisis**

### **2.4.1 Microsoft Excel**

Menurut Ananda (2011), Microsoft Excel merupakan perangkat lunak (*software*) yang mengolah data, meliputi perhitungan dasar, penggunaan fungsi-fungsi, pembuatan grafik, dan manajemen data. Selain itu, Excel juga dapat digunakan untuk menyelesaikan berbagai urusan administratif dari yang sederhana sampai kompleks. Proses menganalisis data peminjaman buku di perpustakaan Otorita Batam menggunakan Microsoft Excel 2013. Proses menganalisis data yang dimaksud salah satunya adalah proses *generalisasi* data.

Alasan menggunakan Microsoft Excel dalam pengolahan data peminjaman buku karena beberapa kelebihanannya, antara lain:

1. Mempunyai kemampuan menampung data yang cukup besar dengan 1 juta baris dan 16.000 kolom dalam 1 sheet. Jadi dalam 1 sheet bisa menampung jawaban 1 juta responden dan 16 ribu jawaban atau pertanyaan.
2. Microsoft Excel mempunyai format yang paling populer dan fleksibel jadi sebagian besar *software* data *entry* ada fasilitas konversi ke format Excel atau format lain yang bisa dibaca Excel. Jika dibutuhkan pengguna Excel bisa konversi balik dari Excel ke *software* statistik lainnya.
3. Microsoft Excel mempunyai program penggunaan rumus yang sangat lengkap sehingga mempermudah pengolahan angka untuk menghasilkan dokumen yang lebih canggih.
4. Menggunakan *Pivot Tables*, pengguna bisa kerja lebih efektif karena semua tabel *summary* yang pengguna rencanakan dapat dibuat dahulu walaupun data belum masuk semua. Setiap ada data masuk otomatis *pivot table* akan me-*refresh* sehingga tabel akan *terupdate* sendiri.

### 2.4.2 Microsoft Visio

Menurut Pascal (2007), Microsoft Visio adalah *software* pembuat diagram dan *flowchart* berbasis vektor yang awalnya dibuat oleh Visio Corporation. Visio mulai terintegrasi di dalam Microsoft Office sejak Office 2007. Sejumlah fitur baru ditambahkan, terutama di integrasi antara data dan diagram, serta otomatisasi dalam pembuatan bagan dan diagram.

Microsoft Visio yang digunakan adalah Microsoft Visio 2013 untuk membuat *FP-tree* dari algoritma *FP-growth*. Kelebihan dari Microsoft Visio 2013 yaitu:

1. *Tool-toolnya* sangat mudah dipahami.
2. Cara penggunaannya sangat mudah.
3. Lebih banyak pilihan gambarnya.

### 2.4.3 Netbeans

Menurut Whitten (2002), Netbeans adalah *Integrated Development Environment (IDE)* berbasis *java* dari Sun Microsystem yang berjalan diatas *Swing*. Netbeans merupakan *software development* yang *open source* dengan kata lain *software* ini dibawah pengembangan bersama dan bebas biaya. Alasan digunakannya Netbeans dalam pembuatan sistem penerapan data mining ini karena sesuai dengan kemampuan laptop yang digunakan untuk pembuatan sistem hingga *running* sistem.

Netbeans yang akan digunakan untuk sistem penerapan data mining ini adalah Netbeans 7. Netbeans 7 memiliki beberapa fitur baru yang lebih sempurna dibandingkan Netbeans versi sebelumnya. Beberapa fitur baru Netbeans 7 yaitu *JDK 7*, *weblogic server*, *oracle database*, *glassfish*, *java*, *Java EE*, bahasa pemrograman web, *PHP*, dan *C/C++*. Aplikasi *java* memiliki komponen GUI yang disimpan pada kontainer yang biasa disebut *form*. Bahasa pemrograman *java* mengumpulkan komponen antarmuka dari *form* GUI yang telah terbentuk. Pada Netbeans komponen GUI ini telah tersedia. Hal ini mempermudah pengguna dalam membangun dan merancang *form java*.

### 2.4.4 XAMPP

Menurut Madcoms (2009), XAMPP adalah salah satu paket *software web server* yang terdiri dari *Apache*, *MySQL*, *PHP*, dan *phpMyAdmin*. Perangkat lunak komputer ini memiliki kelebihan untuk bisa berperan sebagai *server webApache* untuk simulasi pengembangan

*website*. Tool pengembangan web ini mendukung teknologi *web* populer seperti *PHP*, *MySQL*, dan *Perl*. XAMPP juga dilengkapi fitur manajemen *database PHPMyAdmin* seperti pada *server hosting* sungguhan, sehingga pengembang *web* dapat mengembangkan aplikasi *web* berbasis *database* secara mudah.

XAMPP akan digunakan sebagai tempat pembuatan dan penyimpanan *database* dari aplikasi penerapan data mining ini. XAMPP yang digunakan untuk pembuatan *database* adalah XAMPP versi 1.8.3.5. Alasan lebih memilih XAMPP dibandingkan tempat pembuatan *database* lainnya karena XAMPP memiliki beberapa kelebihan, yaitu:

1. Dapat berperan sebagai *server web Apache* untuk simulasi pengembangan *website*.
2. Melalui XAMPP, *programmer* dapat menguji aplikasi *web* yang dikembangkan dan merepresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet.
3. XAMPP dilengkapi fitur manajemen *database PHPMyAdmin* seperti pada *server hosting* sungguhan, sehingga pengembang *web* dapat mengembangkan aplikasi *web* berbasis *database* secara mudah.
4. XAMPP dapat dijalankan di sistem operasi *Windows2000/XP/Vista/7* dan sistem operasi lainnya.

#### **2.4.5 StarUML**

Menurut Triandini (2012), *StarUML* adalah *platform* pemodelan perangkat lunak yang mendukung UML (*Unified Modelling Language*). *StarUML* secara aktif mendukung pendekatan MDA (*Model Driven Architecture*) dengan mendukung konsep UML *profile*. *StarUML* memiliki 9 fitur diagram yang mendukung, antara lain:

1. *Use Case Diagram*
2. *Class Diagram*
3. *Sequence Diagram*
4. *Collaboration Diagram*
5. *Statechart Diagram*
6. *Activity Diagram*
7. *Component Diagram*
8. *Deployment Diagram*
9. *Composite Structure Diagram* (UML versi 2.0)

*StarUML* digunakan untuk membuat beberapa pemodelan sistem yang menerapkan data mining ini. Diagram yang digunakan untuk pemodelan sistem hanya 3 dari 9 diagram, yaitu *use case diagram*, *class diagram*, dan *sequence diagram*. *StarUML* memiliki beberapa

keunggulan, yaitu: ringan, *open source*, dan simbol-simbolnya mudah di mengerti. *StarUML* unggul dalam hal kustomisasi lingkungan kerja pengguna, dan memiliki ekstensibilitas tinggi pada fungsionalitasnya. *StarUML* yang digunakan untuk pemodelan sistem ini yaitu versi 5.0.2.

## BAB III

### ANALISIS DAN PERANCANGAN

#### 3.1 Proses Pengumpulan Data

Pengumpulan data transaksi dilakukan selama 2 minggu di Perpustakaan Otorita Batam. Berdasarkan hasil pengumpulan data, didapatkan data mentah peminjaman buku perpustakaan sebanyak 531 data berdasarkan transaksi yang dilakukan oleh pengunjung maupun pengurus perpustakaan tersebut dari bulan Mei tahun 2013 hingga bulan Maret tahun 2014.

Dalam analisis data peminjaman buku terdapat beberapa proses yang akan dilakukan untuk mendapatkan aturan asosiasi. Setelah pengumpulan data dilakukan, dalam analisis peminjaman buku ini terdapat beberapa tahapan yang akan dilakukan terhadap data transaksi yang terkumpul. Salah satunya yaitu tahapan *preprocessing data*.

#### 3.2 Tahapan Preprocessing Data

*Preprocessing data* adalah suatu proses atau langkah yang dilakukan untuk membuat data mentah menjadi data yang berkualitas. *Preprocessing data* diperlukan karena data di dunia ini kotor artinya terdiri dari beberapa data yaitu data *noise*, data tidak lengkap, dan data tidak konsisten. *Preprocessing data* terdiri dari 4 jenis yaitu data *cleaning*, data *integration*, data *transformation*, dan data *reduction*. Pada penelitian ini, *preprocessing data* yang akan dilakukan adalah transformasi data dan penyeleksian data.

##### 3.2.1 Transformasi Data

*Data transformation* atau transformasi data adalah mengubah bentuk data menjadi bentuk yang sesuai untuk proses *mining*. Transformasi data terdiri dari beberapa jenis yaitu normalisasi, generalisasi, penghalusan, agregasi, dan konstruksi atribut. Data yang telah terkumpul akan ditransformasi secara generalisasi.

Generalisasi data adalah mengubah bentuk data mentah ke konsep yang lebih tinggi melalui penggunaan konsep hierarki. Generalisasi data dalam penelitian ini adalah mengubah data

mentah yang berupa judul buku kemudian digolongkan ke dalam suatu jenis golongan buku yang sama berupa kategori buku. Misal buku yang berjudul Adam Tak Diusir Dari Surga di generalisasikan menjadi kategori buku agama. Contoh pengelompokan data buku berdasarkan kategori buku dapat dilihat seperti pada tabel 6.

**Tabel 6. Transformasi Data**

| No. | Judul Buku   | Itemset yang di Transformasi                           |
|-----|--|--|
| 1   | Merajut Batam Masa Depan; Bercermin Sejarah Batam Masa Depan; Mengungkap Fakta Pembangunan Batam   | Filsafat; Filsafat; Filsafat                           |
| 2   | Referensi Birokrat Publik; Bulan sebagai Ujung Tombak; Komitmen Setengah Hati                      | Sejarah, Geografi, dan Biografi; Kesusasteraan; Bahasa |
| 3   | Catatan Hitam Lima Presiden Indonesia; Pak Beye dan Kerabatnya; The Victory is Yours               | Bahasa; Filsafat; Kesusasteraan                        |
| 4   | Penulisan Skripsi; Jam Piket Organ Tubuh Manusia   | Karya Umum; Kesenian dan Olahraga                      |
| 5   | Food and Cancer; Manajemen Peti Kemas  | Kesenian dan Olahraga; Ilmu Terapan/Teknologi          |
| 6   | Teknik Penulisan Laporan; 3 dari 30 Anekdote dan Abu Nawas   | Bahasa; Kesusasteraan                                  |
| 7   | Batam Sebagai Ujung Tombak; Kertas Kerja Perseorangan (KKP); Laporan Perkembangan Batam Tahun 2006 | Kesusasteraan; Karya Umum; Karya Umum                  |
| 8   | Wirausaha Muda Mandiri; Anak Umum Anak Langit  | Karya Umum; Kesusasteraan                              |
| 9   | Access 2010; Visual Basic 2010   | Ilmu Terapan/Teknologi; Ilmu Terapan/Teknologi         |
| 10  | Adam Tak Diusir dari Surga; Tionghoa Batam   | Agama; Agama   |

Untuk melihat transformasi data selengkapnya dapat dilihat pada lampiran B.

### 3.2.2 Penyeleksian Data

Penyeleksian data dilakukan dengan tujuan mempermudah menemukan asosiasi dari buku. Apabila dalam suatu transaksi terdapat 1-*item* dan 2 atau lebih namun data tersebut tergolong dalam kategori yang sama (sejenis) maka data tersebut akan diseleksi karena tidak akan ditemukan asosiasi pada transaksi tersebut. Contoh penyeleksian data pada *itemset* dapat dilihat seperti pada tabel 7.

**Tabel 7. Penyeleksian Data**

| No. | Judul Buku  | Itemset yang di Transformasi                           |
|-----|---|--|
| 1   | <i>Merajut Batam Masa Depan; Bercermin Sejarah Batam Masa Depan; Mengungkap Fakta Pembangunan Batam</i> | <i>Filsafat; Filsafat; Filsafat</i>                    |
| 2   | Referensi Birokrat Publik; Bulan sebagai Ujung Tombak; Komitmen Setengah Hati                           | Sejarah, Geografi, dan Biografi; Kesusasteraan; Bahasa |
| 3   | Catatan Hitam Lima Presiden Indonesia; Pak Beye dan Kerabatnya; The Victory is Yours                    | Bahasa; Filsafat; Kesusasteraan                        |
| 4   | Penulisan Skripsi; Jam Piket Organ Tubuh Manusia  | Karya Umum; Kesenian dan Olahraga                      |
| 5   | Food and Cancer; Manajemen Peti Kemas   | Kesenian dan Olahraga; Ilmu Terapan/Teknologi          |
| 6   | Teknik Penulisan Laporan; 3 dari 30 Anekdote dan Abu Nawas  | Bahasa; Kesusasteraan                                  |
| 7   | Batam Sebagai Ujung Tombak; Kertas Kerja Perseorangan (KKP); Laporan Perkembangan Batam Tahun 2006      | Kesusasteraan; Karya Umum; Karya Umum                  |
| 8   | Wirausaha Muda Mandiri; Anak Umum Anak Langit   | Karya Umum; Kesusasteraan                              |
| 9   | Access 2010; Visual Basic 2010  | Ilmu Terapan/Teknologi; Ilmu Terapan/Teknologi         |
| 10  | <i>Adam Tak Diusir dari Surga; Tionghoa Batam</i>   | <i>Agama; Agama</i>                                    |

Berdasarkan tabel 7, dapat dilihat bahwa terdapat beberapa transaksi yang dimiringkan (*italic*). Hal tersebut dilakukan untuk menyeleksi transaksi yang terdiri dari 2 *item* yang merupakan transaksi sejenis. Transaksi yang diseleksi tersebut tidak dihilangkan karena tidak akan didapatkan asosiasi pada transaksi tersebut. Setelah melalui tahap *preprocessing*, data yang akan diolah menjadi berkurang hingga diperoleh 301 data. Untuk melihat transaksi yang diseleksi lebih lengkap dapat dilihat pada lampiran C.

### 3.3 Penerapan Algoritma FP-Growth

Data yang telah di *preprocessing* selanjutnya akan diolah kembali menggunakan algoritma *fp-growth* dari metode asosiasi data *mining*. Hal ini dilakukan untuk menemukan pola peminjaman yang dilakukan oleh peminjam buku. Adapun penerapan algoritma *fp-growth* dapat dilakukan beberapa tahap mulai dari menghitung *support count* dari 1-*item*, mengurutkan 1-*item* berdasarkan *support count* terbesar hingga terkecil, hingga pembentukan aturan asosiasi.

### 3.3.1 Menghitung Support Count 1-Item

Langkah awal menganalisis data menggunakan algoritma *fp-growth* adalah menentukan minimum *support count*-nya. Cara menentukan minimum *support count* adalah melalui *support minimum*. Rumus menghitung *support count* minimum adalah sebagai berikut:

$$\text{Support count minimum} = \text{support minimum} * \text{jumlah data}$$

Dalam penelitian ini, ditentukan *support minimum*nya 2% sehingga diperoleh *support count* minimumnya seperti dibawah ini:

$$\text{Support count minimum} = \frac{2}{100} * 301 = 6,02 = 6.$$

Setelah ditemukan *support count* minimumnya, dilanjutkan dengan menghitung *support count* dari 1-*item* dan diperoleh hasil seperti pada tabel 8.

**Tabel 8. Support Count 1-Item**

| Item | Support Count |
|------|---------------|
| a    | 95            |
| b    | 25            |
| c    | 68            |
| d    | 148           |
| e    | 93            |
| f    | 40            |
| g    | 80            |
| h    | 17            |
| i    | 79            |
| j    | 68            |

Nama *item* pada tabel 8 diperoleh kategori buku yang disimbolkan menjadi *item* A, B, dan seterusnya. Tujuannya adalah mempermudah mengolah data transaksi peminjaman buku. Kategori buku dan simbol-simbol dapat dilihat seperti pada tabel 9.

**Tabel 9. Kategori Buku dan Simbolnya**

| Kategori Buku    | Simbol |
|------------------|--------|
| Karya umum       | a      |
| Filsafat         | b      |
| Agama            | c      |
| Ilmu-ilmu sosial | d      |
| Bahasa           | e      |
| Ilmu-ilmu murni  | f      |

|                                 |   |
|---------------------------------|---|
| Ilmu terapan dan teknologi      | g |
| Kesenian dan olahraga           | h |
| Kesusasteraan                   | i |
| Sejarah, geografi, dan biografi | j |

Setelah diperoleh *support count* dari masing-masing *item*, selanjutnya 1-*item* tersebut diurutkan dari *support count* terbesar hingga terkecil sehingga terlihat seperti pada tabel 10.

**Tabel 10. 1-Item dan Support Count Setelah Diurutkan**

| Item | Support Count |
|------|---------------|
| d    | 148           |
| a    | 95            |
| e    | 93            |
| g    | 80            |
| i    | 79            |
| c    | 68            |
| j    | 68            |
| f    | 40            |
| b    | 25            |
| h    | 17            |

Berdasarkan tabel 10, diperoleh *frequent item-list* yaitu d:148, a:95, e:93, g:80, i:79, c:68, j:68, f:40, b:25, h:17. Langkah selanjutnya yaitu mengurutkan transaksi berdasarkan *frequent item-list* sehingga terlihat seperti pada tabel 11. Untuk data selengkapnya dapat dilihat di lampiran D.

**Tabel 11. Transaksi Setelah Diurutkan**

| Transaksi | Item       | Item       |
|-----------|------------|------------|
| 1         | j, i, e    | e, i, j    |
| 2         | e, b, i    | e, i, b    |
| 3         | a, h       | a, h       |
| 4         | h, g       | g, h       |
| 5         | e, i       | e, i       |
| 298       | g, c       | g, c       |
| 299       | e, c, j, g | e, g, c, j |
| 300       | d, j       | d, j       |
| 301       | d, a       | d, a       |

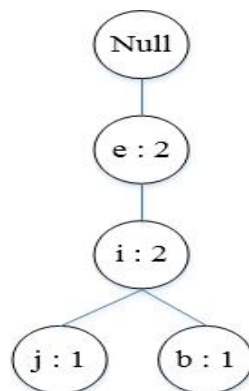
### 3.3.2 Pembangunan FP-Tree

Setelah 301 transaksi diurutkan sesuai dengan *frequent item-list*, langkah selanjutnya adalah pembangunan *fp-tree*. Pembentukan *fp-tree* pada transaksi 1, 2, dan 3 dapat dilihat seperti pada gambar 17 hingga gambar 19.



Gambar 17. Pembentukan FP-Tree Berdasarkan Transaksi 1

Berdasarkan gambar 17, dapat dijelaskan bahwa *FP-tree* tersebut dibangun berdasarkan transaksi 1 pada tabel 11. Untuk simpul pertama pembangunan *FP-tree* selalu diawali dengan simpul *null*, kemudian baru diikuti dengan pembangunan simpul dari *item-item* pada transaksi 1.



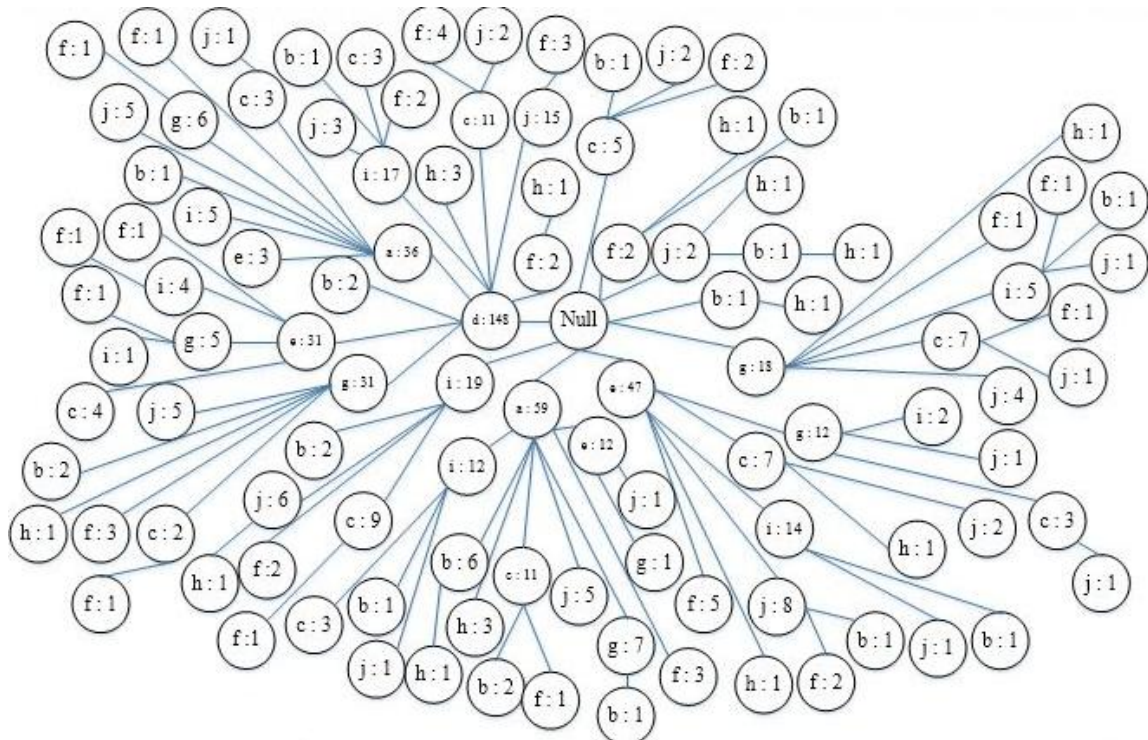
Gambar 18. Pembentukan FP-Tree Berdasarkan Transaksi 2

Berdasarkan gambar 18, dapat dijelaskan bahwa *FP-tree* tersebut dibangun berdasarkan transaksi 2 pada tabel 11. Pembangunan *FP-tree* pada gambar 18 ini merupakan lanjutan dari *FP-tree* pada gambar 17. *Item e* dan *item i* pada transaksi {e,i,b} frekuensinya berubah menjadi 2 karena e dan i dilalui sebanyak 2 kali dan karena setelah e dan i adalah b maka dibangun simpul baru yaitu b:1 karena simpul tersebut tidak dapat dibangun melalui jalur simpul j:1.



pada tabel 11. Dapat dilihat seperti pada gambar 20, semakin banyak transaksi peminjaman yang dilakukan, *FP-tree* yang dibentuk semakin rimbun.

Sementara pembentukan *fp-tree* pada transaksi peminjaman buku akhir atau transaksi 301 dapat dilihat seperti pada gambar 21.



**Gambar 21. Pembentukan FP-Tree Berdasarkan Transaksi 301**

Berdasarkan gambar 21, dapat dijelaskan bahwa *FP-tree* yang terbentuk merupakan lanjutan dari pembangunan *FP-tree* sebelumnya. *FP-tree* ke-301 ini dibangun berdasarkan transaksi pada tabel 11. Dapat dilihat seperti pada gambar 21, *item* g pada lintasan {g,c,f} memiliki frekuensi sebesar 17. *Item* d untuk lintasan {d,e,g,i} memiliki frekuensi sebesar 149. Untuk melihat pembentukan *fp-tree* lebih lengkap dapat dilihat pada lampiran E.

### 3.3.3 Pencarian Frequent Itemset

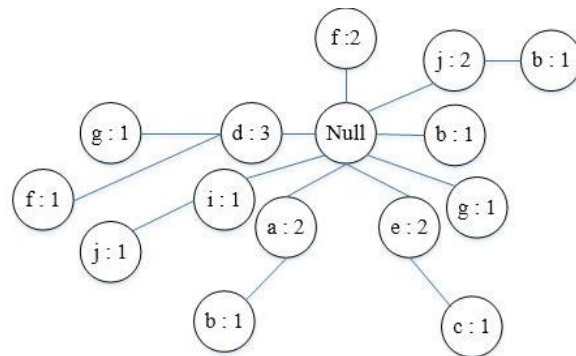
Setelah dibangun 301 *fp-tree*, dilakukan penghitungan kembali pada tiap *item* di *fp-tree* tersebut untuk menyesuaikan dengan tabel 10. Jika jumlah frekuensi atau *support count* masing-masing *item* sama dengan yang terdapat pada tabel 10, artinya *fp-tree* yang dibangun sudah benar. Pada penelitian ini, jumlah *support count* masing-masing *item* sama dengan tabel 10. Artinya *fp-tree* yang dibangun diatas sudah benar dan dapat dilanjutkan pada tahap selanjutnya yaitu pencarian *frequent itemset*.





Algoritma FP-growth menemukan *frequent itemset* yang berakhiran *suffix* tertentu dengan menggunakan metode *divide and conquer* untuk memecah *problem* menjadi *subproblem* yang lebih kecil. Contohnya jika ingin mengetahui semua *frequent itemset* yang berakhiran h, maka harus dipastikan terlebih dahulu apakah h memiliki *support count* yang melebihi atau sama dengan minimum *support count* yang telah ditentukan diawal atau tidak yaitu 6. Karena h memiliki *support count* sebesar 17 maka h dapat dikatakan *item* yang *frequent*.

Setelah mengetahui bahwa *item* h adalah *item* yang *frequent*, maka *subproblem* selanjutnya adalah menemukan *frequent itemset* dengan akhiran bh, fh, jh, ch, ih, gh, eh, ah, dan dh. Dengan menggabungkan seluruh solusi dari *subproblem* yang ada, maka himpunan semua *frequent itemset* yang berakhiran *item* h akan didapatkan. Berdasarkan gambar 22, dapat dilihat bahwa terdapat lintasan yang tidak berakhir di h yaitu: gj, ij, ab, dan df. Langkah selanjutnya untuk menemukan *frequent itemset* yang berakhiran h yaitu dengan membuang semua simpul h dan memperbaharui *support count* simpul di atasnya seperti terlihat pada gambar 26.



**Gambar 26. Lintasan Baru dengan Simpul H Dibuang**

Berdasarkan gambar 26, dapat dijelaskan bahwa simpul h pada lintasan-lintasan yang berakhiran dengan h sudah dibuang dan *support count* simpul atasnya diperbaharui. Seperti pada lintasan {d,g} dan {d,f} pada gambar 26, lintasan sebelumnya terdiri dari {d,g,h}, {d,h} dan {d,f,h}. Kemudian lintasan tersebut dihapus simpul h-nya sehingga hanya tersisa {d,g} dan {d,f}. Sebelumnya frekuensi *item* d adalah 148, setelah diperbaharui maka *support count* atau frekuensi dari *item* d berubah menjadi 3 karena 3 kali dilalui yaitu melalui lintasan {d,f}, {d,g}, dan {d,h}. Begitu pula dengan *item* lainnya.

Sementara lintasan baru dengan simpul d yang dibuang dapat dilihat seperti pada gambar 27.



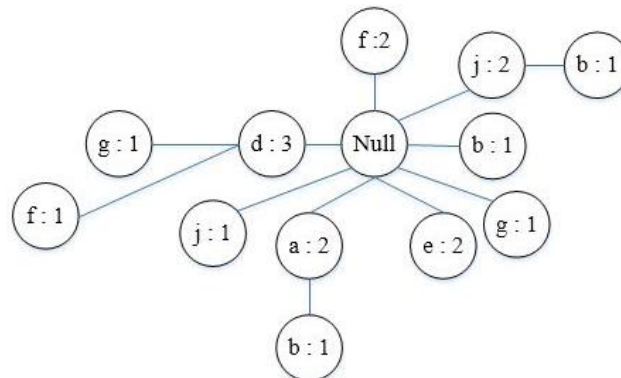
**Gambar 27. Lintasan Baru dengan Simpul D Dibuang**

Berdasarkan gambar 27, dapat dijelaskan bahwa simpul d pada lintasan-lintasan yang berakhir dengan d sudah dibuang dan simpul yang tersisa hanya *null* karena seperti pada gambar 25, lintasan yang mengandung simpul d hanya ada 1 lintasan.

Untuk melihat lintasan baru yang telah dibuang simpulnya dapat dilihat secara lengkap pada lampiran G.

Langkah berikutnya untuk menemukan *frequent itemset* dari masing-masing *item* yaitu dengan memeriksa kembali apakah daun di lintasan baru dengan simpul yang telah dibuang *frequent* atau tidak untuk membentuk *conditional FP-tree*.

Seperti yang terlihat pada gambar 26, terdapat 5 *item* yang tidak *frequent* yaitu g, i, e, a, dan d. Untuk membangun *conditional FP-tree* dari gambar 26, kelima *item* yang tidak *frequent* itu dihapus sehingga *conditional FP-tree* untuk simpul h terlihat seperti pada gambar 28.



**Gambar 28. Conditional FP-Tree untuk H**

Berdasarkan gambar 28, dapat dijelaskan bahwa *conditional FP-tree* yang dibangun untuk h diperoleh setelah melihat pada gambar 26, apakah *item-item* yang mengandung h *frequent* atau tidak. Jika *item* tidak *frequent* maka *item* tersebut tidak dimasukkan ke dalam *conditional FP-tree*. Adapun beberapa *item* yang tidak *frequent* dari gambar 26 yaitu *item* i dan *item* c.

Begitu juga untuk pembentukan *conditional FP-tree* untuk *item* lainnya sehingga *conditional FP-tree* untuk *d* dapat dilihat seperti pada gambar 29.



**Gambar 29. Conditional FP-Tree untuk D**

Berdasarkan gambar 29, dapat dijelaskan bahwa *conditional FP-tree* yang dibangun untuk *d* diperoleh setelah melihat pada gambar 27, apakah *item-item* yang mengandung *d* *frequent* atau tidak. Jika *item* tidak *frequent* maka *item* tersebut tidak dimasukkan ke dalam *conditional FP-tree*. Akan tetapi karena pada gambar 27 tidak terdapat simpul-simpul yang membentuk lintasan kecuali simpul *null*, maka *conditional FP-tree* untuk *item d* seperti yang terlihat pada gambar 29.

Untuk melihat *conditional FP-tree* secara lengkap dapat dilihat pada lampiran H.

Setelah memeriksa *frequent itemset* untuk beberapa akhiran (*suffix*), maka diperoleh hasil seperti yang terangkum pada tabel 12.

**Tabel 12. Hasil Frequent Itemset**

| Suffix | Frequent Itemset   |
|--------|--|
| h      | {h}, {f,h}, {j,h}, {j,b,h}, {b,h}, {g,h}, {e,h}, {a,h}, {a,b,h}, {d,f,h}, {d,g,h}, {d,h}   |
| b      | {b}, {c,b}, {j,b}, {g,i,b}, {e,i,b}, {e,j,b}, {a,g,b}, {a,c,b}, {a,i,b}, {i,b}, {d,g,b}, {d,a,b}, {d,i,b}, {g,b}, {e,b}, {a,b}, {d,b}, {j,b}                             |
| f      | {f}, {c,f}, {g,i,f}, {g,f}, {g,c,f}, {e,j,f}, {e,f}, {a,c,f}, {a,f}, {i,c,f}, {i,f}, {j,f}, {d,g,c,f}, {d,f}, {d,e,g,f}, {d,e,i,f}, {d,a,g,f}, {d,i,f}, {d,c,f}, {d,j,f} |
| j      | {j}, {c,j}, {g,j}, {g,i,j}, {g,c,j}, {e,g,c,j}, {e,j}, {e,c,j}, {e,i,j}, {i,j}, {a,j}, {a,e,j}, {a,i,j}, {d,j}, {d,g,j}, {d,a,j}, {d,a,c,j}, {d,i,j}, {d,c,j}            |
| c      | {c}, {g,c}, {e,c}, {e,g,c}, {a,c}, {a,i,c}, {i,c}, {d,c}, {d,i,c}, {d,a,c}, {d,e,c}, {d,g,c}   |
| i      | {i}, {g,i}, {e,i}, {e,g,i}, {a,i}, {d,i}, {d,a,i}, {d,e,i}, {d,e,g,i}  |
| g      | {g}, {e,g}, {a,g}, {a,e,g}, {d,g}, {d,a,g}, {d,e,g}  |
| e      | {e}, {d,e}, {a,e}, {a,d,e}   |
| a      | {a}, {a,d}   |
| d      | {d}  |

### 3.3.4 Pencarian Confidence dan Aturan Asosiasi

Berdasarkan semua *frequent itemset* yang diperoleh seperti pada tabel 12, tidak semua dihitung *confidence*-nya karena dalam menghasilkan aturan asosiasi, minimal 2 *frequent itemset* yang dihitung terdapat 2 *item* dimana jika meminjam buku A maka meminjam buku B. Sehingga yang layak dihitung *confidence*-nya terdiri dari 95 *subsets*. Setelah mendapatkan *frequent itemset* yang akan dihitung, selanjutnya membuat aturan dengan

mencari *confidence* dari masing-masing *subsets*. *Item* yang memenuhi minimum *confidence* saja yang akan dibuat aturan asosiasinya. Minimum *confidence* untuk analisis data peminjaman buku ini yaitu 55%.

Untuk menemukan *confidence* masing-masing *subsets*, digunakan rumus sebagai berikut:

$$Confidence = P(B|A) = \frac{J_u}{J_v} \frac{h t_1}{h t_2} \frac{m}{m} \frac{A d}{A} \frac{B}{A}$$

Untuk mencari *confidence*, masing-masing *subsets* perlu dibolak balik karena frekuensi dari masing-masing *itemset* tidak sama jumlahnya. Misalnya frekuensi *itemset* f dari *subset* fh dan *itemset* h dari *subset* hf yang berbeda mempengaruhi pembagi dalam perhitungan *confidence* meskipun frekuensi dari hf dan fh sama. Hal ini mengakibatkan perhitungan *confidence* yang seharusnya 95 *subsets* menjadi 406 *subsets*. Adapun potongan hasil perhitungan *confidence* dari masing-masing *subsets* atau *itemset* terlihat seperti pada tabel 13. Untuk hasil perhitungan *confidence* selengkapnya dapat dilihat pada lampiran I.

**Tabel 13. Itemset dan Confidence**

| Itemset | Frekuensi | Confidence |
|---------|-----------|------------|
| f→h     | 2         | 5%         |
| j→h     | 3         | 4%         |
| b→h     | 2         | 8%         |
| g→h     | 2         | 3%         |
| e→h     | 2         | 2%         |
| a→h     | 4         | 4%         |
| d→h     | 5         | 3%         |
| h→b     | 2         | 12%        |
| h→g     | 2         | 12%        |
| h→e     | 2         | 12%        |
| h→a     | 4         | 24%        |
| h→d     | 5         | 29%        |
| j,b→h   | 1         | 13%        |
| j,h→b   | 1         | 33%        |
| b,h→j   | 1         | 50%        |

Berdasarkan hasil perhitungan *confidence* pada tabel 13, diperoleh 10 *itemset* yang memenuhi minimum *confidence* seperti terlihat pada tabel 14.

**Tabel 14. Itemset yang Memenuhi Minimum Confidence**

| Itemset | Frekuensi | Confidence |
|---------|-----------|------------|
| f→j     | 5         | 100%       |
| b,c→a   | 2         | 67%        |
| b,g→d   | 2         | 67%        |
| c,f→d   | 5         | 56%        |
| j,a→d   | 6         | 55%        |
| e,g,f→d | 1         | 100%       |
| e,i,f→d | 1         | 100%       |
| a,g,f→d | 1         | 100%       |
| c,a,j→d | 1         | 100%       |
| d,g,i→e | 1         | 100%       |

Seperti terlihat pada tabel 14, diperoleh 10 *itemset* yang memenuhi minimum *confidence* yaitu 55%. Hal ini berarti terdapat 10 aturan asosiasi yang dapat dibentuk. Aturan asosiasi tersebut dapat dilihat pada tabel 15.

**Tabel 15. Aturan Asosiasi**

| Itemset | Aturan Asosiasi   |
|---------|---|
| f→j     | Jika meminjam buku f, berarti meminjam buku j           |
| b,c→a   | Jika meminjam buku b dan c, berarti meminjam buku a     |
| b,g→d   | Jika meminjam buku b dan g, berarti meminjam buku d     |
| c,f→d   | Jika meminjam buku c dan f, berarti meminjam buku d     |
| j,a→d   | Jika meminjam buku j dan a, berarti meminjam buku d     |
| e,g,f→d | Jika meminjam buku e, g, dan f, berarti meminjam buku d |
| e,i,f→d | Jika meminjam buku e, i, dan f, berarti meminjam buku d |
| a,g,f→d | Jika meminjam buku a, g, dan f, berarti meminjam buku d |
| c,a,j→d | Jika meminjam buku c, a, dan j, berarti meminjam buku d |
| d,g,i→e | Jika meminjam buku d, g, dan i, berarti meminjam buku e |

Berdasarkan proses data mining yang telah dilakukan, maka diperoleh hasil sebagai berikut:

1. {Ilmu-ilmu murni, Sejarah, geografi, dan biografi}.
2. {Filsafat, Agama, Karya umum}.
3. {Filsafat, Ilmu-ilmu terapan dan teknologi, Ilmu-ilmu sosial}.
4. {Agama, Ilmu-ilmu murni, Ilmu-ilmu sosial}.
5. {Sejarah, geografi, dan biografi, Karya umum, Ilmu-ilmu sosial}.
6. {Bahasa, Ilmu-ilmu terapan dan teknologi, Ilmu-ilmu murni, Ilmu-ilmu sosial}.
7. {Bahasa, Kesusasteraan, Ilmu-ilmu murni, Ilmu-ilmu sosial}.
8. {Karya umum, Ilmu-ilmu terapan dan teknologi, Ilmu-ilmu murni, Ilmu-ilmu sosial}.
9. {Agama, Karya umum, Sejarah, geografi, dan biografi, Ilmu-ilmu sosial}.
10. {Ilmu-ilmu sosial, Ilmu-ilmu terapan dan teknologi, Kesusasteraan, Bahasa}.

Terdapat aturan asosiasi yang diperoleh yaitu:

1. Jika meminjam buku ilmu-ilmu murni, maka meminjam buku sejarah, geografi, dan biografi (*confidence*=100%).
2. Jika meminjam buku filsafat dan agama, maka meminjam buku karya umum (*confidence*=67%).
3. Jika meminjam buku filsafat dan ilmu-ilmu terapan dan teknologi, maka meminjam buku ilmu-ilmu sosial (*confidence*=67%).
4. Jika meminjam buku agama dan ilmu-ilmu murni, maka meminjam buku ilmu-ilmu sosial (*confidence*=56%).
5. Jika meminjam buku sejarah, geografi, dan biografi dan karya umum, maka meminjam buku ilmu-ilmu sosial (*confidence*=55%).
6. Jika meminjam buku bahasa, ilmu-ilmu terapan dan teknologi, dan ilmu-ilmu murni, maka meminjam buku ilmu-ilmu sosial (*confidence*=100%).
7. Jika meminjam buku bahasa, kesusasteraan, dan ilmu-ilmu murni, maka meminjam buku ilmu-ilmu sosial (*confidence*=100%).
8. Jika meminjam buku karya umum, ilmu-ilmu terapan dan teknologi, dan ilmu-ilmu murni, maka meminjam buku ilmu-ilmu sosial (*confidence*=100%).
9. Jika meminjam buku agama, karya umum, dan sejarah, geografi, dan biografi, maka meminjam buku ilmu-ilmu sosial (*confidence*=100%).
10. Jika meminjam buku ilmu-ilmu sosial, ilmu-ilmu terapan dan teknologi dan kesusasteraan, maka meminjam buku bahasa (*confidence*=100%).

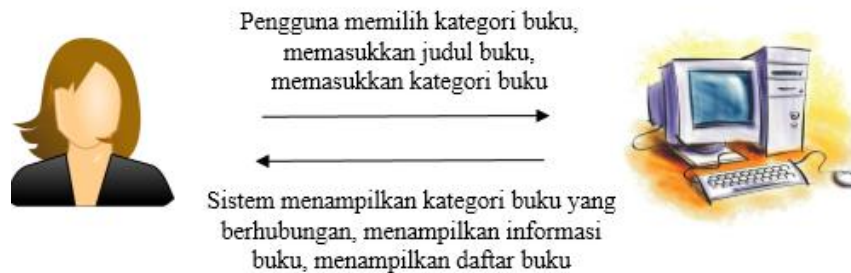
Contoh penguasaan aturan asosiasi dijelaskan sebagai berikut:

1. Sebagai dasar acuan pengelola perpustakaan untuk menyusun koleksi buku seperti yang terlihat pada lampiran J.
2. Mempermudah para pengunjung untuk mencari buku-buku yang berasosiasi atau berhubungan.

### **3.4 Deskripsi Sistem**

Pembuatan sistem yang menerapkan data mining ini, bertujuan untuk menerapkan analisa peminjaman buku di perpustakaan Otorita Batam sehingga dapat membantu pengurus maupun pengunjung perpustakaan untuk mengetahui buku yang saling berasosiasi berdasarkan pola data yang ditemukan melalui analisis data menggunakan algoritma FP-

*growth* sehingga aturan ini, diharapkan mampu meningkatkan minat pengunjung terhadap peminjaman buku di perpustakaan Otorita Batam. Sistem ini juga dapat digunakan oleh pengunjung untuk mencari informasi buku dan daftar buku. Cara kerja sistem ini dapat dilihat seperti pada gambar 30.



**Gambar 30. Deskripsi Sistem**

Berdasarkan gambar 30, dapat dijelaskan proses kerja antara pengguna dan sistem. Langkah pertama pengurus atau pengunjung masuk ke sistem kemudian memilih kategori buku yang ingin dicari asosiasinya. Selanjutnya sistem akan menampilkan kategori buku yang berasosiasi atau berkaitan dengan kategori buku yang dipilih oleh pengguna. Selain itu, pengguna juga dapat memasukkan judul buku ke sistem untuk mengetahui informasi terkait judul buku tersebut. Selanjutnya sistem akan menampilkan informasi terkait judul buku yang dimasukkan pengguna. Informasi tersebut berupa id buku, kategori, pengarang, penerbit, kota terbit, dan tahun terbit buku tersebut. Selain mencari asosiasi dan informasi buku, pengguna juga dapat memasukkan kategori buku ke sistem untuk mengetahui buku apa saja yang termasuk dalam kategori buku tersebut. Selanjutnya sistem akan menampilkan daftar buku yang termasuk kategori buku yang dimasukkan oleh pengguna.

### **3.5 Lingkungan Pengembangan**

Spesifikasi dari sistem adalah:

1. Bahasa pemrograman: Netbeans (*Java*).
2. Perangkat keras (*Hardware*):
  - *Processor*: minimal Pentium 4.
  - Kebutuhan memori utama: minimal 1 GB.
3. Sistem operasi: Windows 7.
4. *Database*: XAMPP.

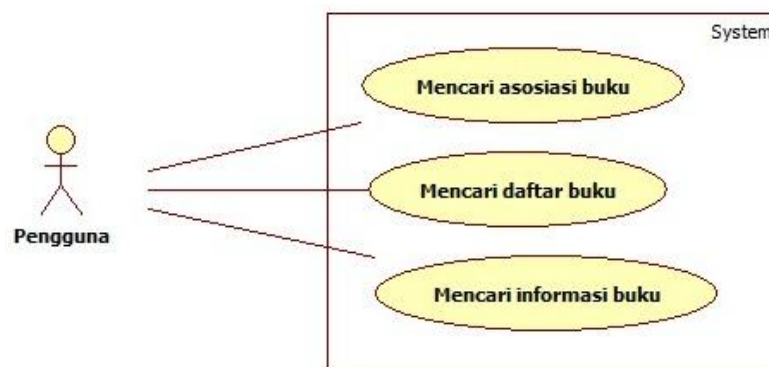
### 3.6 Perancangan Sistem

Perancangan proses pada sistem menggunakan *sequence diagram*. Adapun proses-proses yang ditangani oleh sistem adalah sebagai berikut:

1. Sistem mampu menampilkan asosiasi buku berdasarkan kategori buku.
2. Sistem mampu menampilkan informasi buku berdasarkan judul buku.
3. Sistem mampu menampilkan daftar buku berdasarkan kategori buku.

### 3.7 Use Case Diagram

Diagram *use case* yang digunakan dalam pembuatan sistem dapat dilihat seperti pada gambar 31.



Gambar 31. Use Case Diagram

Berdasarkan gambar 31, dapat dijelaskan bahwa aktor yang menggunakan sistem yaitu pengguna. Pengguna bisa berupa pengunjung perpustakaan maupun pengurus perpustakaan tersebut. Hal yang dapat dilakukan pengguna terhadap sistem ada 3 yaitu mencari asosiasi buku, mencari daftar buku, dan mencari informasi buku. Mencari asosiasi buku artinya pengguna dapat mencari kategori buku yang saling berhubungan dengan memilih kategori buku apa yang ingin dicari asosiasinya. Mencari daftar buku artinya pengguna dapat mengetahui buku-buku apa saja yang termasuk dalam kategori buku yang diinput pengguna ke sistem. Mencari informasi buku artinya pengguna dapat mengetahui informasi terkait judul buku yang diinput pengguna ke sistem. Informasi buku tersebut berupa ID buku, kategori buku, pengarang, penerbit, kota terbit, dan tahun terbit buku tersebut.

### 3.8 Skenario Use Case

Berdasarkan diagram *use case* pada gambar 31, maka pengguna dapat melakukan 3 hal pada sistem yaitu mencari asosiasi buku, mencari daftar buku, dan mencari informasi buku.

### 3.8.1 Skenario Mencari Asosiasi Buku

Skenario mencari asosiasi buku dapat dilihat seperti pada tabel 16.

**Tabel 16. Skenario Use Case Mencari Asosiasi Buku**

|               |   |
|---------------|---|
| Aktor         | Pengguna  |
| Kondisi awal  | Pengguna masuk ke halaman utama sistem  |
| Kondisi akhir | Sistem menampilkan buku yang saling berasosiasi   |
| Skenario      | <ol style="list-style-type: none"><li>1. Pengguna memilih kategori buku yang tersedia di sistem.</li><li>2. Sistem mencari buku yang berasosiasi ke dalam <i>database</i>.</li><li>3. Sistem menampilkan buku yang berasosiasi.</li></ol> |

### 3.8.2 Skenario Mencari Daftar Buku

Skenario mencari daftar buku dapat dilihat seperti pada tabel 17.

**Tabel 17. Skenario Use Case Mencari Daftar Buku**

|               |  |
|---------------|--|
| Aktor         | Pengguna   |
| Kondisi awal  | Pengguna masuk ke halaman utama sistem   |
| Kondisi akhir | Sistem menampilkan daftar buku   |
| Skenario      | <ol style="list-style-type: none"><li>1. Pengguna memasukkan kategori buku ke sistem.</li><li>2. Sistem mencari daftar buku terkait kategori buku ke dalam <i>database</i>.</li><li>3. Sistem menampilkan daftar buku yang termasuk dalam kategori buku yang <i>diinput</i>.</li></ol> |

### 3.8.3 Skenario Mencari Informasi Buku

Skenario mencari informasi buku dapat dilihat seperti pada tabel 18.

**Tabel 18. Skenario Use Case Mencari Informasi Buku**

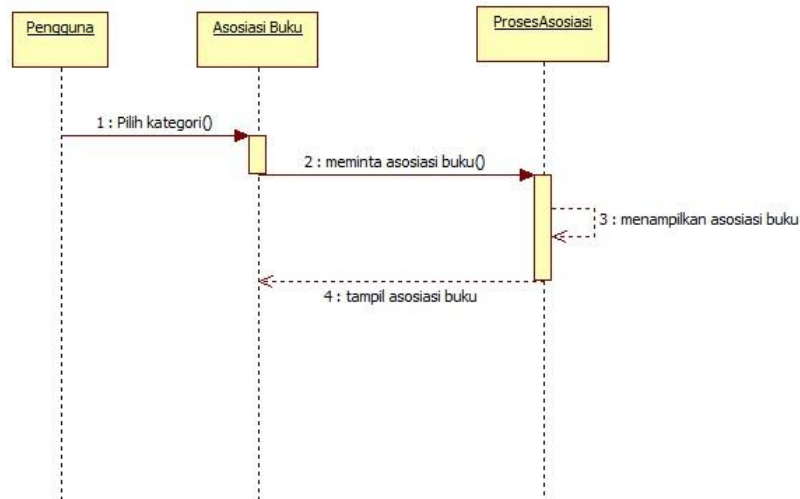
|               |   |
|---------------|---|
| Aktor         | Pengguna  |
| Kondisi awal  | Pengguna masuk ke halaman utama sistem  |
| Kondisi akhir | Sistem menampilkan informasi buku   |
| Skenario      | <ol style="list-style-type: none"><li>1. Pengguna memasukkan judul buku ke sistem.</li><li>2. Sistem mencari informasi buku terkait judul buku ke dalam <i>database</i>.</li><li>3. Sistem menampilkan informasi buku sesuai dengan judul buku yang <i>diinput</i>.</li></ol> |

## 3.9 Sequence Diagram

*Sequence diagram* dalam penelitian ini ada 3 sesuai dengan diagram *use case* yaitu *sequence diagram* mencari asosiasi buku, *sequence diagram* mencari daftar buku, dan *sequence diagram* mencari informasi buku.

### 3.9.1 Sequence Diagram Mencari Asosiasi Buku

*Sequence diagram* mencari asosiasi buku terlihat seperti pada gambar 32.

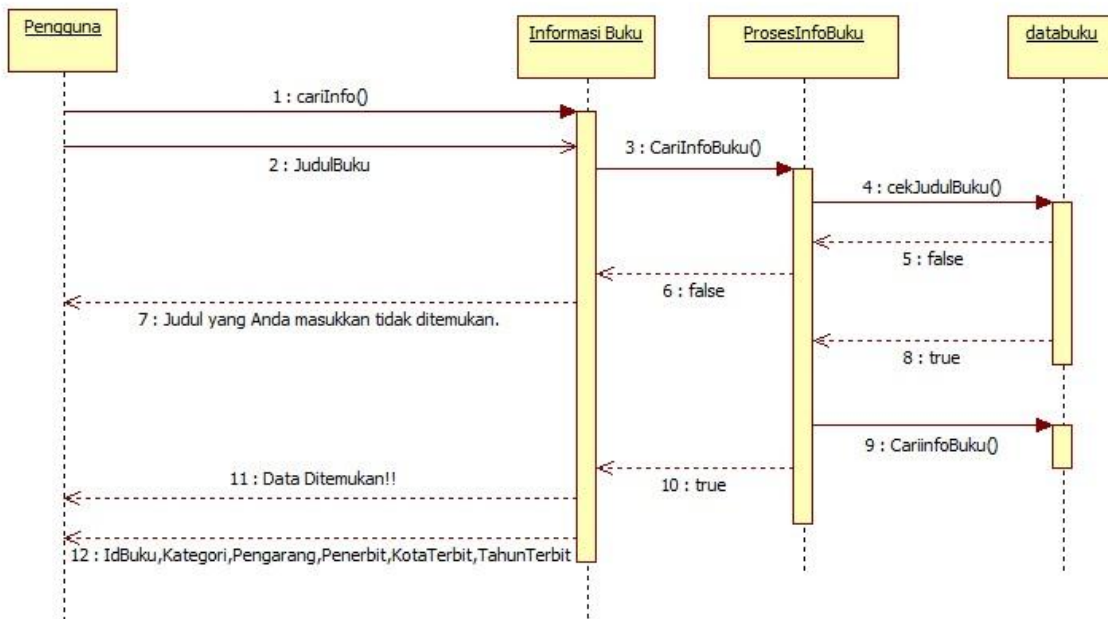


**Gambar 32. Sequence Diagram Mencari Asosiasi Buku**

Berdasarkan gambar 32, dapat dijelaskan bahwa *sequence diagram* mencari asosiasi buku dimulai dengan pengguna yang masuk ke halaman asosiasi buku kemudian memilih kategori yang sudah tersedia pada halaman tersebut. Selanjutnya pengguna menekan tombol cari dan asosiasi dari kategori buku yang dipilih pengguna sebelumnya akan ditampilkan ke halaman asosiasi buku tersebut.

### 3.9.2 Sequence Diagram Mencari Informasi Buku

*Sequence diagram* mencari informasi buku terlihat seperti pada gambar 33.

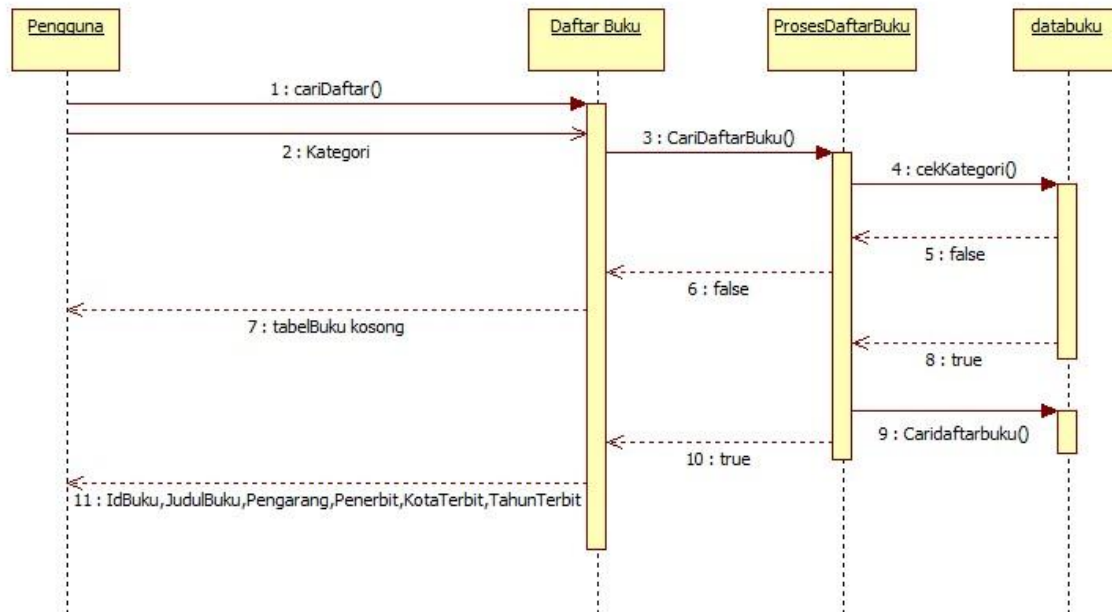


**Gambar 33. Sequence Diagram Mencari Informasi Buku**

Berdasarkan gambar 33, *sequence diagram* mencari informasi buku menjelaskan proses aktor yaitu pengguna yang melakukan pencarian informasi buku. Pertama sistem akan menampilkan *form* informasi buku. Pada *form* ini pengguna akan memasukkan judul buku yang diinginkan ke sistem kemudian mengklik tombol cari. Selanjutnya sistem akan mengecek apakah judul buku yang dimasukkan pengguna benar atau tidak. Jika judul buku yang dimasukkan oleh pengguna salah atau tidak tersedia dalam *database*, maka sistem akan menampilkan pesan kesalahan. Namun apabila judul yang dimasukkan oleh pengguna benar atau ada dalam *database*, maka sistem akan menampilkan pesan bahwa data ditemukan dan menampilkan informasi buku yang diinginkan pengguna berupa IdBuku, Kategori, Pengarang, Penerbit, KotaTerbit, dan TahunTerbit.

### 3.9.3 Sequence Diagram Mencari Daftar Buku

Sequence diagram mencari daftar buku terlihat seperti pada gambar 34.

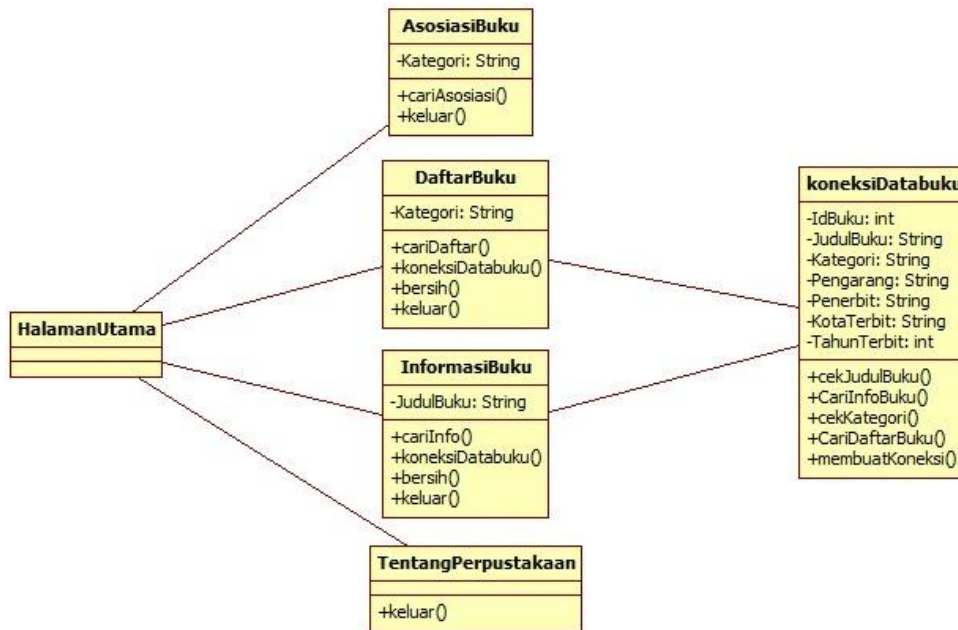


Gambar 34. Sequence Diagram Mencari Daftar Buku

Berdasarkan gambar 34, *sequence diagram* mencari daftar buku menjelaskan proses aktor yaitu pengguna yang melakukan pencarian daftar buku. Pertama sistem akan menampilkan *form* daftar buku. Pada *form* ini pengguna akan memasukkan kategori buku yang diinginkannya ke sistem kemudian mengklik tombol cari. Selanjutnya sistem akan mengecek apakah kategori yang dimasukkan pengguna benar atau tidak. Jika judul buku yang dimasukkan oleh pengguna salah atau tidak tersedia dalam *database*, maka tabel buku pada halaman daftar buku tidak akan menampilkan apa-apa (kosong). Namun apabila kategori yang dimasukkan oleh pengguna benar atau ada dalam *database*, maka sistem akan menampilkan daftar buku yang diinginkan pengguna pada tabel buku yang terdiri dari beberapa kolom yaitu IdBuku, JudulBuku, Pengarang, Penerbit, KotaTerbit, dan TahunTerbit.

### 3.10 Class Diagram

*Class diagram* atau kelas diagram untuk penelitian ini dapat dilihat seperti pada gambar 35.



Gambar 35. Class Diagram

Berdasarkan gambar 35, dapat dijelaskan bahwa kelas diagram untuk penelitian ini terdiri dari beberapa kelas yang terdiri dari 5 kelas GUI dan 1 kelas koneksi yang menghubungkan kelas GUI dengan *database*. Kelas GUI yang diakses menggunakan *database* hanya 2 yaitu kelas DaftarBuku dan kelas InformasiBuku.

### 3.11 Desain Antar Muka Sistem

Desain antarmuka sistem ini ada 4 mulai dari halaman utama, halaman asosiasi buku, halaman informasi buku, hingga halaman daftar buku.

### 3.11.1 Desain Antar Muka Halaman Utama

Desain antar muka halaman utama dapat dilihat seperti pada gambar 36.



Gambar 36. Desain Antar Muka Halaman Utama

Berdasarkan gambar 36, dapat dilihat bahwa desain antar muka halaman utama sistem terdiri dari *menu bar* yang terdiri dari *menu* dan *tentang*, *label* selamat datang, *label* otorita, *label* gedung, *label* PTI, *label* BP Batam, dan gambar perpustakaan Otorita Batam. *Menu bar* menu jika diklik memiliki 3 *submenu* yaitu asosiasi buku, informasi buku, dan daftar buku. Masing-masing *submenu* tersebut jika diklik akan terhubung ke halaman lainnya. Jika diklik *submenu* asosiasi buku, maka sistem akan menampilkan halaman asosiasi buku, begitu juga untuk *submenu* lainnya. Sedangkan untuk *menu bar* tentang berisi *submenu* perpustakaan. Jika *submenu* tersebut diklik, maka sistem akan menampilkan halaman tentang perpustakaan yang berisi tentang informasi-informasi terkait perpustakaan Otorita Batam.

Penjelasan dari desain antar muka halaman utama dapat dilihat pada tabel 19.

Tabel 19. Penjelasan Desain Antar Muka Halaman Utama

| Id_Objek   | Jenis | Nama    | Keterangan  |
|------------|-------|---------|---|
| mn_menu    | Menu  | Menu    | Jika pengguna mengklik menu ini akan muncul 3 submenu pilihan yaitu submenu asosiasi buku, informasi buku, dan daftar buku.   |
| mn_tentang | Menu  | Tentang | Jika pengguna mengklik menu ini maka akan muncul submenu perpustakaan yang menghubungkan dengan halaman tentang perpustakaan. |

|               |       |                               |   |
|---------------|-------|-------------------------------|---|
| lb_SELAMATDTG | Label | SELAMAT DATANG                | Hanya sebuah tulisan selamat datang.                              |
| lb_POB        | Label | DI PERPUSTAKAAN OTORITA BATAM | Hanya sebuah tulisan nama perpustakaan.                           |
| lb_Gedung     | Label | Gedung B, Lt 1-2              | Hanya sebuah tulisan letak perpustakaan.                          |
| lb_PTI        | Label | Pusat Teknologi Informasi     | Hanya sebuah tulisan letak perpustakaan lanjutan atasnya.         |
| lb_BPBATAM    | Label | BP Batam                      | Hanya sebuah tulisan letak perpustakaan lanjutan atasnya.         |
| gambarPOB     | Label | -                             | Hanya sebuah gambar penampakan gedung perpustakaan Otorita Batam. |

### 3.11.2 Desain Antar Muka Asosiasi Buku

Desain antar muka asosiasi buku dapat dilihat seperti pada gambar 37.

Gambar 37. Desain Antar Muka Asosiasi Buku

Berdasarkan gambar 37, dapat dilihat bahwa desain antar muka asosiasi buku terdiri dari *label* Asosiasi Buku, *label* pilih kategori, *combo box* kategori1, *combo box* kategori2, *combo box* kategori2, *label* asosiasi buku, *label* asosiasi, *label* hasil asosiasi, dan 2 tombol fungsi cari dan keluar. Tombol cari digunakan untuk mencari asosiasi buku yang berkaitan dengan kategori buku yang dipilih. Apabila pengguna mengklik tombol cari, sistem akan menampilkan hasil asosiasi dari kategori buku yang dipilih ke *label* hasil. Apabila pengguna mengklik tombol keluar, maka akan muncul *option pane* yakin atau tidak ingin keluar dari halaman asosiasi buku. Jika pengguna memilih ya, maka sistem akan keluar dari halaman asosiasi buku. Jika pengguna memilih tidak, maka sistem akan tetap menampilkan halaman asosiasi buku. Penjelasan dari desain antar muka asosiasi buku dapat dilihat pada tabel 20.

**Tabel 20. Penjelasan Desain Antar Muka Asosiasi Buku**

| Id_Objek         | Jenis      | Nama                 | Keterangan   |
|------------------|------------|----------------------|--|
| lb_asosiasibuku  | Label      | Asosiasi Buku        | Hanya tulisan judul halaman tersebut.  |
| lb_kategoribuku1 | Label      | Pilih kategori buku: | Hanya tulisan sebagai petunjuk.  |
| cbKategori1      | Combo box  | --Pilih--            | Jika pengguna mengklik combo box ini maka pengguna dapat memilih kategori buku yang telah disediakan sistem untuk mencari asosiasi bukunya. Jika pengguna memilih salah satu kategori pada combo box ini maka combo box lainnya secara otomatis akan berubah juga. |
| lb_kategoribuku2 | Label      | Pilih kategori buku: | Hanya tulisan sebagai petunjuk.  |
| cbKategori2      | Combo box  | --Pilih--            | Jika pengguna mengklik combo box ini maka pengguna dapat memilih kategori buku yang telah disediakan sistem untuk mencari asosiasi bukunya. Jika pengguna memilih salah satu kategori pada combo box ini maka combo box lainnya secara otomatis akan berubah juga. |
| lb_kategoribuku3 | Label      | Pilih kategori buku: | Hanya tulisan sebagai petunjuk.  |
| cbKategori3      | Combo box  | --Pilih--            | Jika pengguna mengklik combo box ini maka pengguna dapat memilih kategori buku yang telah disediakan sistem untuk mencari asosiasi bukunya. Combo box ini terhubung dengan combo box lainnya.  |
| btnCari          | Button     | Cari                 | Jika pengguna mengklik tombol ini, maka pengguna akan menemukan kategori buku yang berasosiasi dengan kategori yang pengguna pilih.  |
| Separator        | Separator  | -                    | Hanya berupa garis pemisah antara tempat pengguna memilih kategori dan hasil yang ditampilkan kepada pengguna.   |
| lb_asosiasibukuu | Label      | Asosiasi Buku        | Hanya tulisan judul halaman tersebut.  |
| lb_berasosiasi   | Label      | Berasosiasi dengan:  | Hanya petunjuk bahwa hasil akan tampil dsekitar label tersebut.  |
| txtAsosiasibuku  | Text Field | -                    | Tempat menampilkan hasil pencarian asosiasi buku.  |
| btnKeluar        | Button     | Keluar               | Jika pengguna mengklik tombol ini, maka akan muncul pertanyaan yakin atau tidak untuk keluar dari sistem. Jika pengguna memilih ya, maka pengguna akan keluar dari halaman asosiasi buku, jika tidak   |

|  |  |  |   |
|--|--|--|---|
|  |  |  | maka pengguna akan tetap berada dalam halaman tersebut. |
|--|--|--|---|

### 3.11.3 Desain Antar Muka Informasi Buku

Desain antar muka informasi buku dapat dilihat seperti pada gambar 38.

Gambar 38. Desain Antar Muka Informasi Buku

Berdasarkan gambar 38, dapat dilihat bahwa desain antar muka informasi buku terdiri dari *label* Informasi Buku, *label* judul buku, *text field* judul buku, *label* informasi buku, *label* id buku, *label* kategori, *label* pengarang, *label* penerbit, *label* kota terbit, *label* tahun terbit, *text field* id buku, *text field* kategori, *text field* pengarang, *text field* penerbit, *text field* kota terbit, *text field* tahun terbit, dan 3 tombol fungsi cari, ulang dan keluar. Tombol cari digunakan untuk mencari informasi buku yang berkaitan dengan judul buku yang *diinput* pengguna. Apabila pengguna mengklik tombol cari, sistem akan menampilkan informasi buku dari judul buku yang *diinput* ke *text field* id buku sampai *text field* tahun terbit. Apabila pengguna mengklik tombol ulang, maka secara otomatis seluruh *text field* akan berubah menjadi seperti semula atau kosong kembali. Apabila pengguna mengklik tombol keluar, maka akan muncul *option pane* yakin atau tidak ingin keluar dari halaman informasi buku. Jika pengguna memilih ya, maka sistem akan keluar dari halaman informasi buku. Jika pengguna memilih tidak, maka sistem akan tetap menampilkan halaman informasi buku. Penjelasan dari desain antar muka informasi buku dapat dilihat pada tabel 21.

**Tabel 21. Penjelasan Desain Antar Muka Informasi Buku**

| Id_Objek          | Jenis      | Nama                 | Keterangan  |
|-------------------|------------|----------------------|---|
| lb_informasibuku  | Label      | Informasi Buku       | Hanya judul halaman tersebut.   |
| lb_judulbuku      | Label      | Masukkan judul buku: | Hanya petunjuk untuk text field dibawahnya.   |
| txtJudulBuku      | Text Field | -                    | Tempat memasukkan judul buku yang diinginkan pengguna.  |
| btnCari           | Button     | Cari                 | Jika pengguna mengklik tombol ini, sistem akan menampilkan informasi buku berdasarkan judul buku yang dimasukkan pengguna sebelumnya. |
| btnUlang          | Button     | Ulang                | Jika pengguna mengklik tombol ini, maka seluruh text field akan berubah seperti semula atau menjadi kosong kembali.                   |
| Separator         | Separator  | -                    | Hanya garis pemisah antara tempat pengguna memasukkan judul buku dan tempat hasil pencarian informasi buku.                           |
| lb_informasibuku1 | Label      | Informasi Buku       | Hanya judul halaman tersebut.   |
| lb_IDBuku         | Label      | ID Buku:             | Hanya keterangan id buku.   |
| txtIDBuku         | Text Field | -                    | Tempat hasil pencarian informasi buku berupa id buku dari judul buku yang dimasukkan pengguna sebelumnya.                             |
| lb_Kategori       | Label      | Kategori:            | Hanya keterangan kategori buku.   |
| txtKategori       | Text Field | -                    | Tempat hasil pencarian informasi buku berupa kategori buku dari judul buku yang dimasukkan pengguna sebelumnya.                       |
| lb_Pengarang      | Label      | Pengarang:           | Hanya keterangan pengarang.   |
| txtPengarang      | Text Field | -                    | Tempat hasil pencarian informasi buku berupa pengarang dari judul buku yang dimasukkan pengguna sebelumnya.                           |
| lb_Penerbit       | Label      | Penerbit:            | Hanya keterangan penerbit.  |
| txtPenerbit       | Text Field | -                    | Tempat hasil pencarian informasi buku berupa penerbit dari judul buku yang dimasukkan pengguna sebelumnya.                            |
| lb_KotaTerbit     | Label      | Kota Terbit:         | Hanya keterangan kota terbit.   |
| txtKotaTerbit     | Text Field | -                    | Tempat hasil pencarian informasi buku berupa kota terbit dari judul buku yang dimasukkan pengguna sebelumnya.                         |

|                |            |               |   |
|----------------|------------|---------------|---|
| lb_TahunTerbit | Label      | Tahun Terbit: | Hanya keterangan tahun terbit.  |
| txtTahunTerbit | Text Field | -             | Tempat hasil pencarian informasi buku berupa tahun terbit dari judul buku yang dimasukkan pengguna sebelumnya.  |
| btnKeluar      | Button     | Keluar        | Jika pengguna mengklik tombol ini, maka akan muncul pertanyaan yakin atau tidak untuk keluar dari sistem. Jika pengguna memilih ya, maka pengguna akan keluar dari halaman informasi buku, jika tidak maka pengguna akan tetap berada dalam halaman tersebut. |

### 3.11.4 Desain Antar Muka Daftar Buku

Desain antar muka daftar buku dapat dilihat seperti pada gambar 39.

Gambar 39. Desain Antar Muka Daftar Buku

Berdasarkan gambar 39, dapat dilihat bahwa desain antar muka daftar buku terdiri dari *label* Daftar Buku, *label* kategori, *text field* kategori buku, *label* daftar buku, tabel daftar buku, dan 3 tombol fungsi cari, ulang dan keluar. Tombol cari digunakan untuk mencari daftar buku yang berkaitan dengan kategori buku yang *diinput* pengguna. Apabila pengguna

mengklik tombol cari, sistem akan menampilkan daftar buku dari kategori buku yang diinput ke tabel daftar buku. Apabila pengguna mengklik tombol ulang, maka secara otomatis *text field* kategori buku akan berubah menjadi seperti semula atau kosong kembali. Apabila pengguna mengklik tombol keluar, maka akan muncul *option pane* yakin atau tidak ingin keluar dari halaman daftar buku. Jika pengguna memilih ya, maka sistem akan keluar dari halaman daftar buku. Jika pengguna memilih tidak, maka sistem akan tetap menampilkan halaman daftar buku. Penjelasan dari desain antar muka daftar buku dapat dilihat pada tabel 22.

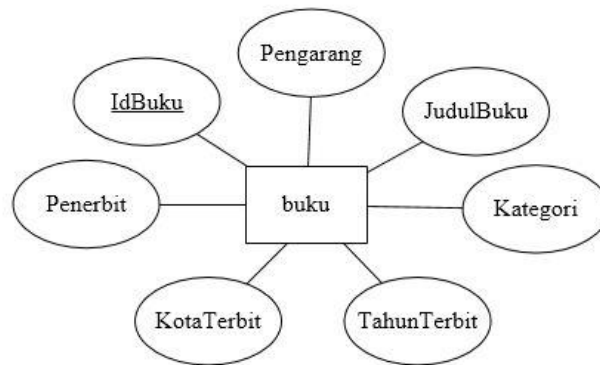
**Tabel 22. Penjelasan Desain Antar Muka Daftar Buku**

| Id_Objek        | Jenis      | Nama                    | Keterangan   |
|-----------------|------------|-------------------------|--|
| lb_daftarbuku   | Label      | Daftar Buku             | Hanya judul halaman tersebut.  |
| lb_kategoribuku | Label      | Masukkan kategori buku: | Hanya petunjuk untuk text field dibawahnya.  |
| txtKategoriBuku | Text Field | -                       | Tempat memasukkan kategori buku yang diinginkan pengguna.  |
| btnCari         | Button     | Cari                    | Jika pengguna mengklik tombol ini, sistem akan menampilkan daftar buku berdasarkan kategori buku yang dimasukkan pengguna sebelumnya.                                      |
| btnUlang        | Button     | Ulang                   | Jika pengguna mengklik tombol ini, maka text field kategori buku akan berubah seperti semula atau menjadi kosong kembali.  |
| Separator       | Separator  | -                       | Hanya garis pemisah antara tempat pengguna memasukkan kategori buku dan tempat hasil pencarian daftar buku.  |
| lb_daftarbuku1  | Label      | Daftar Buku             | Hanya judul halaman tersebut.  |
| tabelBuku       | Table      | -                       | Tempat menampilkan hasil pencarian daftar buku berdasarkan kategori buku yang dimasukkan pengguna sebelumnya.  |
| btnKeluar       | Button     | Keluar                  | Jika pengguna mengklik tombol ini, maka akan muncul pertanyaan yakin atau tidak untuk keluar dari sistem. Jika pengguna memilih ya, maka pengguna akan keluar dari halaman |

|  |  |  |   |
|--|--|--|---|
|  |  |  | daftar buku, jika tidak maka pengguna akan tetap berada dalam halaman tersebut. |
|--|--|--|---|

### 3.12 ER Diagram

*Entity relational diagram* berkaitan dengan *database* atau basis data dari sistem. *ER diagram* digunakan untuk mengetahui hubungan antar tabel *database* yang digunakan dalam sistem. *ER diagram* untuk penelitian ini dapat dilihat pada gambar 40.



**Gambar 40. Entity Relational Diagram Tabel Buku**

Berdasarkan gambar 40, dapat dilihat tabel buku dari *database* databuku memiliki beberapa atribut yang akan digunakan untuk penyimpanan data-data buku. Atribut-atribut yang dimaksud yaitu IdBuku, JudulBuku, Penerbit, Kategori, KotaTerbit, TahunTerbit, dan Pengarang. Adapun atribut yang digunakan sebagai *primary key* yaitu atribut IdBuku karena sifatnya yang berbeda-beda untuk tiap buku artinya setiap buku memiliki id yang berbeda-beda.

#### 3.12.1 Spesifikasi Database Databuku

Penjelasan dari tabel buku pada *database* databuku dapat dilihat pada tabel 23.

**Tabel 23. Spesifikasi Tabel Buku**

| Nama Field  | Deskripsi Isi            | Type & Panjang | Boleh NULL  | Keterangan  |
|-------------|--------------------------|----------------|-------------|-------------|
| IdBuku      | Berisi id buku           | Varchar (5)    | Tidak boleh | Primary key |
| Kategori    | Berisi kategori buku     | Varchar (100)  | Tidak boleh | -           |
| JudulBuku   | Berisi judul buku        | Varchar (100)  | Tidak boleh | -           |
| Penerbit    | Berisi penerbit buku     | Varchar (50)   | Tidak boleh | -           |
| KotaTerbit  | Berisi kota terbit buku  | Varchar (30)   | Tidak boleh | -           |
| TahunTerbit | Berisi tahun terbit buku | Int (4)        | Tidak boleh | -           |
| Pengarang   | Berisi pengarang buku    | Varchar (30)   | Tidak boleh | -           |

# BAB IV

## HASIL DAN PEMBAHASAN

### 4.1 Algoritma

#### 4.1.1 Algoritma Mencari Asosiasi Buku

```
Pemrograman Mencari Asosiasi Buku
{untuk mencari asosiasi buku dari kategori buku yang dipilih pengguna}

Deklarasi:
    cbKategori1, cbKategori2, cbKategori3: String

Algoritma:
    if
        cbKategori1 = Ilmu-ilmu murni
            then
                output (“Sejarah, geografi, dan biografi”)

    else if
        cbKategori1 = Filsafat && cbKategori2 = Agama
            then
                output (“Karya umum”)

    else if
        cbKategori1 = Filsafat && cbKategori2 = Ilmu terapan dan teknologi
            then
                output (“Ilmu-ilmu sosial”)

    else if
        cbKategori1 = Agama && cbKategori2 = Ilmu-ilmu murni
            then
                output (“Ilmu-ilmu sosial”)

    else if
        cbKategori1 = Sejarah, geografi, dan biografi && cbKategori2 = Karya umum
            then
                output (“Ilmu-ilmu sosial”)

    else if
        cbKategori1 = Bahasa && cbKategori2 = Ilmu terapan dan teknologi && cbKategori3 =
        Ilmu-ilmu murni
            then
                output (“Ilmu-ilmu sosial”)

    else if
        cbKategori1 = Bahasa && cbKategori2 = Kesusteraan && cbKategori3 = Ilmu-ilmu murni
            then
```

```

        output (“Ilmu-ilmu sosial”)

    else if
        cbKategori1 = Karya umum && cbKategori2 = Ilmu terapan dan teknologi && cbKategori3
        = Ilmu-ilmu murni
            then
                output (“Ilmu-ilmu sosial”)

    else if
        cbKategori1 = Agama && cbKategori2 = Karya umum && cbKategori3 = Sejarah, geografi,
        dan biografi
            then
                output (“Ilmu-ilmu sosial”)

    else if
        cbKategori1 = Ilmu-ilmu sosial && cbKategori2 = Ilmu terapan dan teknologi &&
        cbKategori3 = Kesusasteraan
            then
                output (“Bahasa”)

```

#### 4.1.2 Algoritma Mencari Informasi Buku

```

Pemrograman Mencari Informasi Buku
{untuk mencari informasi buku berdasarkan judul buku yang dimasukkan pengguna}

Deklarasi:
    txtJudulBuku: String
    txtIDBuku, txtTahunTerbit: int
    txtKategori, txtPengarang, txtPenerbit, txtKotaTerbit: String

Algoritma:
    if
        txtJudulBuku == empty
            then
                output (“Silahkan masukkan judul buku”)
    else if
        txtJudulBuku == true
            then
                output (txtIDBuku, txtKategori, txtPengarang, txtPenerbit, txtKotaTerbit,
                txtTahunTerbit)

                output (“Data Ditemukan!”)
    else
        then
            output (“Judul buku tidak ditemukan”)

```

#### 4.1.3 Algoritma Mencari Daftar Buku

```

Pemrograman Mencari Daftar Buku
{untuk mencari daftar buku berdasarkan kategori buku yang dimasukkan pengguna}

Deklarasi:

```

```

txtKategori: String
txtJudulBuku, txtPengarang, txtPenerbit, txtKotaTerbit: String
txtIDBuku, txtTahunTerbit: int

```

Algoritma:

```

if
  txtKategori == empty
  then
    output ("Silahkan masukkan kategori buku")
else if
  txtKategori == true
  then
    output (txtIDBuku, txtJudulBuku, txtPengarang, txtPenerbit, txtKotaTerbit,
            txtTahunTerbit)
else
  then
    output (null)

```

## 4.2 Implementasi Basis Data

Implementasi basis data dari sistem yang menerapkan data *mining* ini dijelaskan pada tabel 24. Basis data atau *database* yang digunakan dalam sistem ini hanya 1 yaitu *database* databuku dan terdiri dari 1 tabel juga yang bernama tabel buku.

**Tabel 24. Implementasi Basis Data Tabel Buku**

| Nama Field  | Tipe    | Panjang | Keterangan  |
|-------------|---------|---------|-------------|
| IdBuku      | Int     | 5       | Primary key |
| Kategori    | Varchar | 100     | -           |
| Pengarang   | Varchar | 30      | -           |
| JudulBuku   | Varchar | 100     | -           |
| Penerbit    | Varchar | 50      | -           |
| KotaTerbit  | Varchar | 30      | -           |
| TahunTerbit | Int     | 4       | -           |

## 4.3 Implementasi Kelas

Implementasi kelas dalam sistem yang menerapkan data *mining* ini dijelaskan dalam tabel 25.

**Tabel 25. Implementasi Kelas**

| No. | Nama Kelas          | Nama File Implementasi   |
|-----|---------------------|--------------------------|
| 1.  | HalamanUtama        | HalamanUtama.java        |
| 2.  | AsosiasiBuku        | AsosiasiBuku.java        |
| 3.  | InformasiBuku       | InfoBuku.java            |
| 4.  | DaftarBuku          | DaftarBuku.java          |
| 5.  | TentangPerpustakaan | TentangPerpustakaan.java |

## 4.4 Implementasi Perancangan AntarMuka

Berikut adalah hasil *screen shot* dari implementasi perancangan *input* dan *output* data.

### 4.4.1 Hasil Implementasi Perancangan Halaman Utama

Hasil implementasi perancangan halaman utama dapat dilihat pada gambar 41.



Gambar 41. Hasil Implementasi Halaman Utama

Berdasarkan gambar 41, dapat dilihat bahwa halaman utama dari sistem memiliki *menu bar* yang terdiri dari menu menu dan tentang. Untuk menu menu dan tentang memiliki *submenu* masing-masing. Untuk menu menu memiliki *submenu* yang terlihat seperti pada gambar 42.



Gambar 42. Hasil Implementasi Halaman Utama Submenu Menu

Berdasarkan gambar 42, dapat dilihat bahwa menu menu terdiri dari 3 *submenu* yang jika dipilih salah satunya maka akan menghubungkan ke halaman yang diklik. Misalnya jika pengguna mengklik *submenu* asosiasi buku, maka sistem akan menampilkan halaman antar muka asosiasi buku kepada pengguna, begitu pula untuk *submenu* lainnya.

Menu tentang juga memiliki *submenu* sama halnya dengan menu menu seperti yang terlihat pada gambar 43.

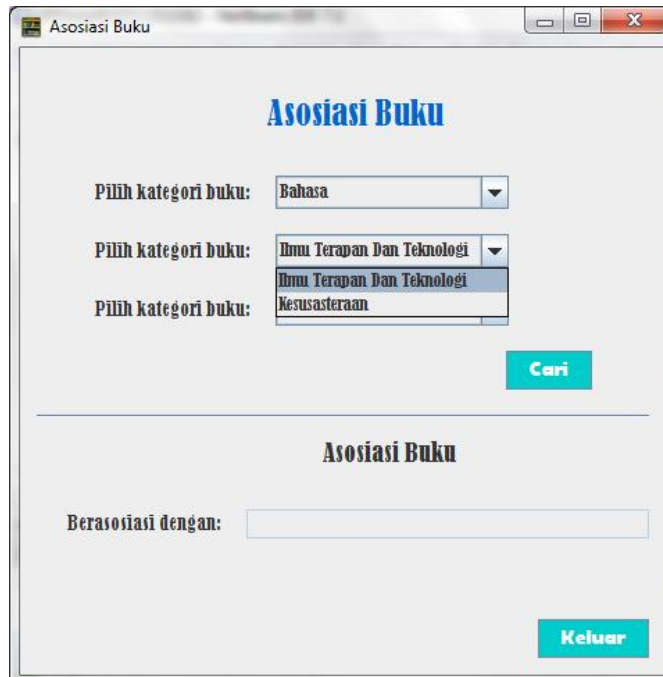


Gambar 43. Hasil Implementasi Halaman Utama Submenu Tentang

Berdasarkan gambar 43, dapat dilihat bahwa menu tentang terdiri dari 1 *submenu* saja yaitu *submenu* perpustakaan. Jika pengguna mengklik *submenu* perpustakaan, maka sistem akan menampilkan halaman antar muka tentang perpustakaan yang berisi informasi-informasi terkait perpustakaan seperti alamat perpustakaan, jam kerja perpustakaan, kategori buku yang tersedia di perpustakaan, dan lain sebagainya.

#### 4.4.2 Hasil Implementasi Perancangan Asosiasi Buku

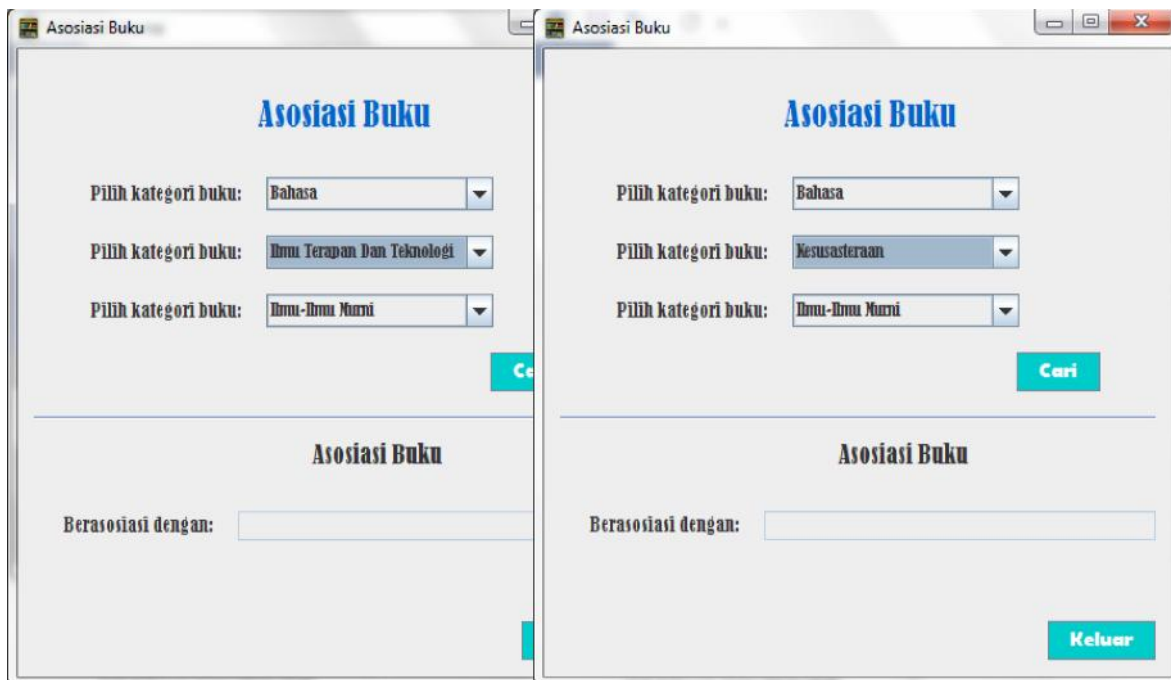
Hasil implementasi perancangan *input* asosiasi buku dapat dilihat seperti pada gambar 44.



Gambar 44. Hasil Implementasi Input Asosiasi Buku

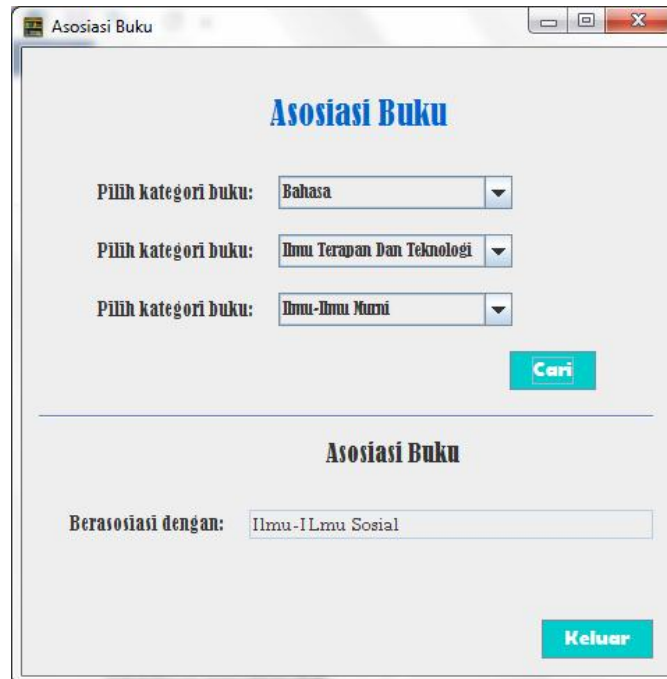
Berdasarkan gambar 44, dapat dilihat bahwa apabila pengguna memilih salah satu kategori buku di *combo box* teratas, maka secara otomatis 2 *combo box* dibawahnya akan mengikuti kategori apa yang dipilih pengguna di *combo box* teratas. Contohnya seperti gambar 44, pengguna memilih kategori buku bahasa untuk *combo box* teratas maka 2 *combo box* dibawahnya mengikuti dengan *combo box* kedua secara otomatis terdiri dari 2 pilihan yaitu kategori buku ilmu terapan dan teknologi dan kesusasteraan. Apabila pengguna memilih kategori buku ilmu murni dan teknologi maupun kesusasteraan pada *combo box* kedua, maka

secara otomatis *combo box* terakhir akan terisi kategori buku ilmu-ilmu murni seperti pada gambar 45.



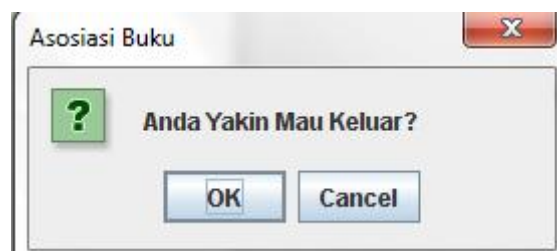
Gambar 45. Hasil Implementasi Input Asosiasi Buku Combo Box Kedua

Berdasarkan gambar 45, dapat dilihat bahwa apabila pengguna memilih kategori buku ilmu terapan dan teknologi maupun kesusasteraan pada *combo box* kedua maka pada *combo box* ketiga secara otomatis tetap akan terisi kategori buku ilmu-ilmu murni. Hasil *output* dari *input* asosiasi buku diatas dapat dilihat pada gambar 46.



Gambar 46. Hasil Implementasi Output Asosiasi Buku

Berdasarkan gambar 46, dapat dilihat bahwa jika kategori buku bahasa, ilmu terapan dan teknologi, dan ilmu-ilmu murni, maka berasosiasi dengan ilmu-ilmu sosial. Hasil *output* dari *input* yang dilakukan melalui *combo box* terletak pada *text field* dibawahnya. Hasil *output* akan tampil pada antar muka asosiasi buku apabila pengguna mengklik tombol cari setelah memilih kategori buku pada *combo box*. Apabila pengguna mengklik tombol keluar maka akan muncul pilihan seperti pada gambar 47.




Gambar 47. Option Pane Tombol Keluar Asosiasi Buku

Berdasarkan gambar 47, dapat dilihat bahwa terdapat 2 pilihan ketika pengguna mengklik tombol keluar pada halaman asosiasi buku yaitu tombol ok dan *cancel*. Apabila pengguna mengklik tombol ok, maka secara otomatis halaman asosiasi buku akan tertutup. Namun apabila pengguna mengklik tombol *cancel*, maka pengguna akan tetap berada di halaman asosiasi buku.

### 4.4.3 Hasil Implementasi Perancangan Informasi Buku

Hasil implementasi perancangan *input* informasi buku dapat dilihat seperti pada gambar 48.



The screenshot shows a web browser window with the title "Informasi Buku". The main content area has a heading "Informasi Buku" in blue. Below it, there is a label "Masukkan judul buku:" and a text input field containing "La Tahzan". Underneath the input field are two buttons: "Ulang" and "Cari". A horizontal line separates this section from the next. The second section is also titled "Informasi Buku" and contains several input fields: "ID Buku", "Kategori", "Pengarang", "Penerbit", "Kota Terbit", and "Tahun Terbit". A "Keluar" button is located at the bottom right of this section.

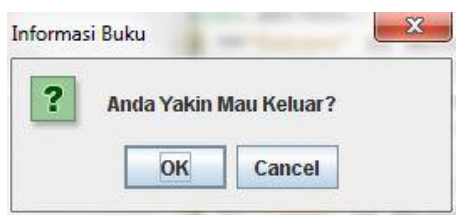
Gambar 48. Hasil Implementasi Input Informasi Buku

Berdasarkan gambar 48, dapat dilihat pengguna memasukkan salah satu judul buku yaitu “La Tahzan” pada *text field* judul buku. Selanjutnya pengguna mengklik tombol cari pada halaman informasi buku tersebut sehingga informasi buku terkait judul buku “La Tahzan” terlihat seperti pada gambar 49.



**Gambar 49. Hasil Implementasi Output Informasi Buku**

Berdasarkan gambar 49, dapat dilihat bahwa *output* dari *input* judul buku yang dimasukkan pengguna sebelumnya ditampilkan pada masing-masing *text field* dibawahnya. Selain itu, apabila judul buku tersebut tersedia pada *database*, sistem akan menampilkan pesan “Data Ditemukan!”. Tombol ulang pada halaman informasi buku digunakan untuk mengosongkan *text field* seperti semula. Jika pengguna mengklik tombol keluar pada halaman informasi buku, maka akan muncul pesan seperti terlihat pada gambar 50.

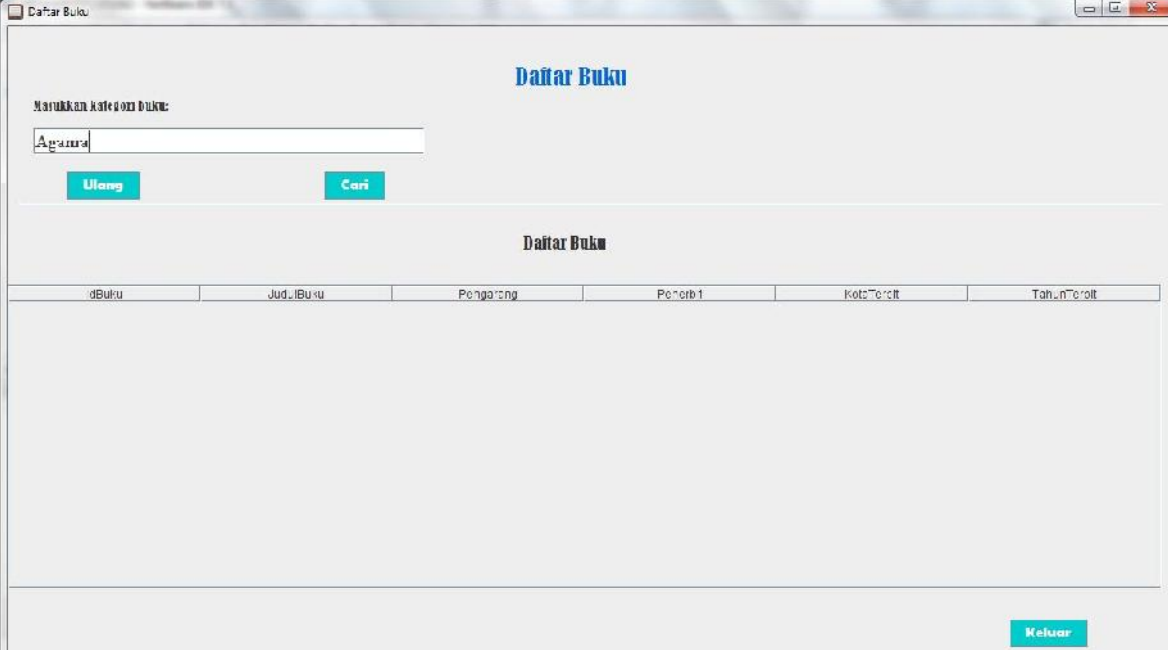


**Gambar 50. Option Pane Tombol Keluar Informasi Buku**

Berdasarkan gambar 50, dapat dilihat bahwa terdapat 2 pilihan ketika pengguna mengklik tombol keluar pada halaman informasi buku yaitu tombol ok dan *cancel*. Apabila pengguna mengklik tombol ok, maka secara otomatis halaman informasi buku akan tertutup. Namun apabila pengguna mengklik tombol *cancel*, maka pengguna akan tetap berada di halaman informasi buku.

#### 4.4.4 Hasil Implementasi Perancangan Daftar Buku

Hasil implementasi perancangan *input* daftar buku dapat dilihat seperti pada gambar 51.



| idBuku | JudulBuku | Penerbit | Kota:Terbit | Tahun:Terbit |
|--------|-----------|----------|-------------|--------------|
|        |           |          |             |              |

**Gambar 51. Hasil Implementasi Input Daftar Buku**

Berdasarkan gambar 51, dapat dilihat pengguna memasukkan salah satu kategori buku yaitu “Agama” pada *text field* kategori. Apabila pengguna mengklik tombol ulang, maka *text field* kategori tersebut akan kembali kosong seperti semula. Apabila pengguna mengklik tombol cari, maka sistem akan menampilkan daftar buku yang termasuk dalam kategori buku “Agama” seperti terlihat pada gambar 52.

Daftar Buku

Memilih kategori buku:

Agama

Ulang Cari

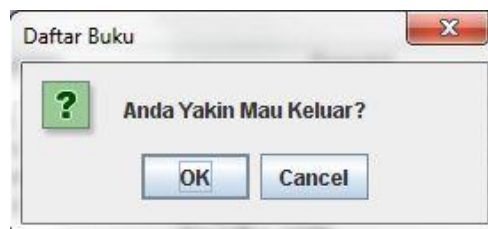
Daftar Buku

| IDBuku | JudulBuku                      | Pengarang                    | Penerbit                  | KotaTglDit | TahunTerbit |
|--------|--------------------------------|------------------------------|---------------------------|------------|-------------|
| 24     | Adam Tak Diusir dari Surga     | Agus Mustafa                 | PADMA press               | Surabaya   | 2007        |
| 25     | Tionghoa Botam                 | Edi Satrio, dkk              | Botam Link Publisher      | Botam      | 2006        |
| 27     | La Takson                      | Dr. Adhal Qomari             | Qisthi Press              | Jakarta    | 2008        |
| 28     | Radhikah Amal                  | Maulana Muhammad Zakariya... | Pustaka Fomation          | Yogyakarta | 2006        |
| 29     | Doa Umam Doa yang Terdapat     | Drs. Imam Ghosali            | Aladamlit pressindo       | Jakarta    | 1991        |
| 32     | Ternyata Adam Ditahirkan       | Agus Mustafa                 | PADMA press               | Surabaya   | 2007        |
| 33     | Fiqih Sunnah Waktu 1           | Abu Malik Kamal              | Pena Pundi Aksara         | Jakarta    | 2007        |
| 43     | Tuntunan Sholat Lima Waktu     | M. Nurhidayah                | CV. Ansh Offset           | Batam      | 1994        |
| 46     | Bukan Yesus yang Saya Kenal    | Philip yancy                 | Gospel Press              | Jakarta    | 1998        |
| 47     | Kekuatan Kata Tuhak            | William Jey                  | Uruk Press                | Jakarta    | 2007        |
| 54     | Hikmah dalam Hidup             | Adulain salam basyirudin     | Gema insani press         | Jakarta    | 1998        |
| 59     | The Choice Dalam Islam Umum... | ahmad Deedat                 | Pustaka Al-Kautsar        | Jakarta    | 1995        |
| 75     | Mempergunakan 60 Orang yang... | Prof. Dr. Muhammad Bakwis    | Al-Qadim                  | Jakarta    | 2011        |
| 92     | Alimiah Sains untuk Madia      | Iskandar ag soenabrata       | Republika                 | Jakarta    | 2006        |
| 94     | Tuntunan Sholat 5 Waktu        | M. Nurhidayah                | PT Gramedia Pustaka Utama | Jakarta    | 1994        |
| 95     | Hukum Umar bin al-Khattab      | Hingaryyah wana maqsood &... | nezzar                    | Jakarta    | 1991        |
| 96     | 441 Hra (Hadist) Allah         | Samar Mahmod al-hussein      | Gema insan                | Jakarta    | 2011        |
| 97     | Hindu Apa Harapan              | Agus Mustafa                 | PADMA press               | Surabaya   | 2004        |

Keluar

Gambar 52. Hasil Implementasi Output Daftar Buku

Berdasarkan gambar 52, dapat dilihat bahwa apabila kategori buku yang diinput pengguna tersedia pada database maka daftar buku yang termasuk dalam kategori buku yang diinput pengguna akan dimunculkan pada tabel daftar buku. Apabila pengguna mengklik tombol keluar pada halaman daftar buku akan muncul pilihan seperti pada gambar 53.



Gambar 53. Option Pane Tombol Keluar Daftar Buku

Berdasarkan gambar 53, dapat dilihat bahwa terdapat 2 pilihan ketika pengguna mengklik tombol keluar pada halaman daftar buku yaitu tombol ok dan cancel. Apabila pengguna mengklik tombol ok, maka secara otomatis halaman daftar buku akan tertutup. Namun apabila pengguna mengklik tombol cancel, maka pengguna akan tetap berada di halaman daftar buku.

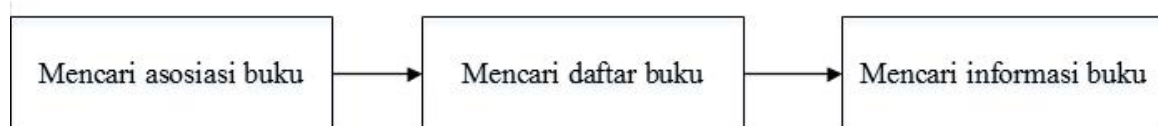
## 4.5 Pengujian

### 4.5.1 Strategi Pengujian

Strategi pengujian pada sistem yang menerapkan analisis data *mining* ini menggunakan metode *black box* yaitu berfokus pada kebutuhan deskripsi fungsional guna untuk mengetahui jalannya sistem secara lengkap.

### 4.5.2 Skenario Pengujian

Skenario pengujian sistem yang menerapkan data *mining* ini dijelaskan seperti pada gambar 54.



Gambar 54. Skenario Pengujian Sistem

Berdasarkan gambar 54, dapat dijelaskan bahwa skenario pengujian yang akan dilakukan pada sistem ada 3 yaitu pengujian mencari asosiasi buku, mencari daftar buku, dan mencari informasi buku.

### 4.5.3 Hasil Pengujian

Pengujian telah dilaksanakan dengan baik sesuai dengan metode *black box* yang digunakan dan hasilnya dapat dilihat di pada tabel 26.

Tabel 26. Hasil Pengujian Sistem

| No. | Use Case               | Fungsi                           | Skenario  | Data Uji  | Target  | Pengujian    | Penguji   |
|-----|------------------------|----------------------------------|---|---|---|--------------|---|
| 1   | Mencari asosiasi buku  | Cari asosiasi dari kategori buku | 1. Memilih kategori buku pada <i>combo box</i><br>2. Menekan tombol cari                    | Kategori 1: Bahasa<br>Kategori 2: Ilmu Terapan Dan Teknologi<br>Kategori 3: Ilmu-Ilmu Murni | - Sistem menampilkan kategori buku yang berhubungan dengan kategori yang dipilih pengguna.  | OK           | Nama : Yasnil Ilyas<br>Umur : 21 Tahun<br>Jenis Kelamin : Laki-laki |
| 2   | Mencari informasi buku | Cari informasi dari judul buku   | 1. Pengguna memasukkan judul buku di <i>text field</i> judul buku<br>2. Menekan tombol cari | <b>Data Benar</b><br>Judul buku : La tahzan<br><br><b>Data Salah</b><br>Judul buku : abcde  | - Sistem menampilkan pesan "Data Ditemukan!" dan menampilkan informasi terkait judul buku.<br>- Sistem menampilkan pesan kesalahan "Judul yang Anda masukkan tidak tersedia." | OK<br><br>OK | Nama : Yasnil Ilyas<br>Umur : 21 Tahun<br>Jenis Kelamin : Laki-laki |

|   |                     |                                |   |   |   |    |  |
|---|---------------------|--------------------------------|---|---|---|----|--|
| 3 | Mencari daftar buku | Cari daftar dari kategori buku | 1. Pengguna memasukkan kategori buku di <i>text field</i> kategori buku<br>2. Menekan tombol cari | <b>Data Benar</b><br>Kategori buku : Agama  | - Tabel buku menampilkan daftar buku terkait kategori tersebut. | OK | Nama : Sulasmono<br>Umur : 60 Tahun<br>Jenis Kelamin : Laki-laki |
|   |                     |                                |   | <b>Data Salah</b><br>Kategori buku : fisika | - Tabel buku tidak menampilkan apa-apa (kosong).                | OK |  |

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan tahapan analisis, perancangan, dan implementasi pada analisis data pola peminjaman buku di perpustakaan Otorita Batam, maka dapat diambil kesimpulan:

1. Aturan asosiasi atau pola data peminjaman buku di Perpustakaan Otorita Batam yang diperoleh melalui pengolahan data menggunakan algoritma *FP-Growth* dari metode asosiasi, yaitu:
  - ) Jika meminjam kategori buku Ilmu-ilmu murni, maka meminjam kategori buku Sejarah, geografi, dan biografi.
  - ) Jika meminjam kategori buku Filsafat dan kategori buku Agama, maka meminjam kategori buku Karya umum.
  - ) Jika meminjam kategori buku Filsafat dan kategori buku Ilmu terapan dan teknologi, maka meminjam kategori buku Ilmu-ilmu sosial.
  - ) Jika meminjam kategori buku Agama dan kategori buku Ilmu-ilmu murni, maka meminjam kategori buku Ilmu-ilmu sosial.
  - ) Jika meminjam kategori buku Sejarah, geografi, dan biografi dan kategori buku Karya umum, maka meminjam kategori buku Ilmu-ilmu sosial.
  - ) Jika meminjam kategori buku Bahasa, kategori buku Ilmu terapan dan teknologi, dan kategori buku Ilmu-ilmu murni, maka meminjam kategori buku Ilmu-ilmu sosial.
  - ) Jika meminjam kategori buku Bahasa, kategori buku Kesusasteraan, dan kategori buku Ilmu-ilmu murni, maka meminjam kategori buku Ilmu-ilmu sosial.
  - ) Jika meminjam kategori buku Karya umum, kategori buku Ilmu terapan dan teknologi, dan kategori buku Ilmu-ilmu murni, maka meminjam kategori buku Ilmu-ilmu sosial.
  - ) Jika meminjam kategori buku Agama, kategori buku Karya umum, dan kategori buku Sejarah, geografi, dan biografi, maka meminjam kategori buku Ilmu-ilmu sosial.

- ) Jika meminjam kategori buku Ilmu-ilmu sosial, kategori buku Ilmu terapan dan teknologi, dan kategori buku Kesusasteraan, maka meminjam kategori buku Bahasa.
2. Sistem ini dapat menampilkan buku-buku yang saling berhubungan, informasi buku, dan daftar buku berdasarkan kategori buku.
  3. Algoritma *FP-Growth* menganalisis data hanya dengan *menscan database* paling banyak 2 kali.
  4. Algoritma *FP-Growth* menemukan *frequent itemset* dengan pembangkitan *FP-tree*.

## 5.2 Saran

Saran-saran yang dianggap dapat menyempurnakan analisis data pola peminjaman buku di perpustakaan Otorita Batam, adalah:

1. Dapat menganalisis data pola peminjaman buku menggunakan algoritma yang lain.
2. Dapat menambahkan fitur untuk memproses aturan asosiasi yang berfungsi seperti WEKA.
3. Dapat menambahkan fitur-fitur yang mempermudah pengguna saat mengoperasikan sistem dan merancang sistem agar lebih *user friendly*.
4. Dapat mempermudah pengguna mengetahui tata letak buku.

## DAFTAR PUSTAKA

- Ananda, S. dan Rouf, I., 2011, *Buku Pintar Menguasai Microsoft Excel Untuk Pemula*, MediaKita, Jakarta.
- Fatihatul, F., dkk, 2012, *Asosiasi Data Mining Menggunakan Algoritma FP-Growth Untuk Market Basket Analysis*, Universitas Padjadjaran, Sumedang.
- Han, J. dan Kamber, M., 2006, *Data mining: Concepts and Techniques*, Morgan-Kaufman, New York.
- Kusrini, dan Luthfi ET., 2009, *Algoritma Data Mining*, Andi Offset, Yogyakarta.
- Larose, D. T., 2005, *Discovering Knowledge in Data: An Introduction to Data Mining*, John Willey & Sons, Kanada.
- Madcoms, 2009, *Menguasai XHTML, CSS, PHP, & MySQL Melalui DREAMWEAVER*, Penerbit Andi, Yogyakarta.
- Milburga, L. C., dkk, 1986, *Membina Perpustakaan Sekolah*, Kanisius, Yogyakarta.
- Moleong, L. J., 2007, *Metodologi Penelitian Kualitatif*, Rosda, Bandung.
- Pascal, S. A., 2007, *Tip & Trik Microsoft Office 2007*, Elex Media Komputindo, Jakarta.
- Putra, K. A., dkk, 2004, *Penerapan Metode Association Rule Dengan Algoritma FP-Growth Untuk Mengetahui Karakteristik Pembeli Mobil*, Universitas Brawijaya, Malang.
- Samuel, D., 2008, *Penerapan Struktur FP-Tree dan Algoritma FP-Growth dalam Optimasi Penentuan Frequent Itemset*, Institut Teknologi Bandung, Bandung.
- Triandini, E. dan Suardika, I. G., 2012. *Step By Step Desain Proyek Menggunakan UML*, Andi Offset, Yogyakarta.
- Turban, E., dkk, 2005, *Decision Support Systems and Intelligent System*, Andi Offset, Yogyakarta.
- Whitten, J. L., dkk, 2002, *Metode Desain & Analis Sistem Ed. 6*, McGraw Hill Education.

Yolanda, R. N. dan Annisa, D. A., 2012, *Analisis Data Pola Peminjaman Buku Perpustakaan Menggunakan Metode Asosiasi*, Politeknik Negeri Batam, Batam.