



**SISTEM PENGKLASIFIKASIAN REKOMENDASI BUKU PERPUSTAKAAN  
UNIVERSITAS JEMBER MENGGUNAKAN ALGORITMA APRIORI**

**SKRIPSI**

Oleh

**MAHYUDHA FATCHUL YAQIEN**

**NIM 112410101054**

**PROGRAM STUDI SISTEM INFORMASI  
UNIVERSITAS JEMBER**

**2015**



**SISTEM PENGKLASIFIKASIAN REKOMENDASI BUKU PERPUSTAKAAN  
UNIVERSITAS JEMBER MENGGUNAKAN ALGORITMA APRIORI**

**SKRIPSI**

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan pendidikan di Program Studi Sistem Informasi Universitas Jember dan mendapat gelar Sarjana Sistem Informasi

Oleh

**MAHYUDHA FATCHUL YAQIEN**

**NIM 112410101054**

**PROGRAM STUDI SISTEM INFORMASI  
UNIVERSITAS JEMBER**

**2015**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang telah memberikan anugerah dan karuniaNya sehingga skripsi saya selesai sesuai dengan yang diharapkan;
2. Ayah saya Alip Susanto serta ibu saya Anik Suryani yang selalu memberikan dukungan dan doanya selama kuliah;
3. Adek Wahyu Ahmad Firdauz dan Mutiara Fadillah yang selalu memberikan motivasi dan dukungan;
4. Guru – guruku sejak taman kanak – kanak hingga perguruan tinggi;
5. Seluruh teman – teman yang selalu memberikan dukungan dan bantuan;
6. Almamater Program Studi Sistem Informasi Universitas Jember.

**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Mahyudha Fatchul Yaqien

NIM : 112410101054

Menyatakan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pengklasifikasian Rekomendasi Buku Perpustakaan Universitas Jember Menggunakan Algoritma Apriori” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar

Jember, Desember 2015

Yang menyatakan,

Mahyudha Fatchul Yaqien

NIM. 112410101054

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Sistem Pengklasifikasian Rekomendasi Buku Perpustakaan Universitas Jember Menggunakan Algoritma Apriori”, telah diuji dan disahkan pada:

Hari, tanggal : Senin, Desember 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Prof. Drs. Slamir, M.Comp.Sc., Ph.D

NIP. 196704201992011001

M. Arief Hidayat, S.Kom.,M.Kom.

NIP. 1981101232010121003



**SKRIPSI**

**SISTEM PENGKLASIFIKASIAN REKOMENDASI BUKU PERPUSTAKAAN  
UNIVERSITAS JEMBER MENGGUNAKAN ALGORITMA APRIORI**

Oleh:

**MAHYUDHA FATCHUL YAQIEN**

NIM. 112410101054

Pembimbing

Pembimbing Utama : Prof. Drs. Slamir, M.Comp.Sc., Ph.D

Pembimbing Anggota : M. Arief Hidayat, S.Kom,.M.Kom.

**PENGESAHAN**

Skripsi berjudul “Sistem Pengklasifikasian Rekomendasi Buku Perpustakaan Universitas Jember Menggunakan Algoritma Apriori”, telah diuji dan disahkan pada:

Hari, tanggal :

Tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji

Penguji II,

Anang Andrianto ST.,MT  
NIP. 196906151997021002

Yanuar Nurdiansyah ST.,M.Cs.  
NIP. 198201012010121004

Mengesahkan  
Ketua Program Studi

Prof. Drs. Slamini, M.Comp.Sc., Ph.D  
NIP. 196704201992011001

## RINGKASAN

**Sistem Pengklasifikasian Rekomendasi Buku Perpustakaan Universitas Jember Menggunakan Algoritma Apriori**; Mahyudha Fatchul Yaqien, 112410101054; 2015; 120 halaman; Program Studi Sistem Informasi Universitas Jember.

Rekomendasi terhadap suatu buku yang terdapat keterkaitan dengan buku lain pada suatu perpustakaan biasanya didapat dari komunikasi yang perlu adanya pengetahuan tentang buku tersebut, belum tentu buku 1 dengan buku lainnya yang memiliki judul yang hampir sama memiliki materi yang berkaitan. Rekomendasi buku yang saling berkaitan bisa didapatkan dengan seberapa sering buku yang dimaksud tersebut sering dipinjam secara bersamaan, semakin sering buku – buku tersebut sering dipinjam bersamaan, maka besar kemungkinan pula *support* dan *confidence* yang didapat, sehingga rekomendasi buku menjadi lebih akurat.

Salah satu alternatif yang dapat digunakan untuk mendapatkan informasi mengenai rekomendasi buku – buku perpustakaan tersebut, yaitu menggunakan salah satu algoritma dari *association rules* pada *Datamining*, algoritma apriori. Dengan perhitungan apriori, program dapat mengolah data transaksi yang telah dilakukan di dalam perpustakaan, dibagi menjadi data – data buku yang saling berkaitan 1 sama lain, dan selanjutnya algoritma akan menghitung seberapa sering buku – buku tersebut sering dipinjam dengan waktu yang sama dan saling terhubung 1 sama lain, dan selanjutnya apriori akan mencari nilai *support* dan *confidence* sehingga peminjam mendapatkan informasi yang lebih mengenai rekomendasi buku dimana terdapat value *support* dan *confidence*, semakin besar nilai value tersebut, besar kemungkinan buku tersebut saling berkaitan 1 sama lain.

## PRAKATA

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Karya Ilmiah Tertulis (Skripsi) berjudul “Sistem Pengklasifikasian Rekomendasi Buku Perpustakaan Universitas Jember Menggunakan Algoritma Apriori.”.

Pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada :

1. Prof. Drs. Slamun, M.Comp.Sc., Ph.D selaku Ketua Program Studi Sistem Informasi Universitas Jember ;
2. Prof. Drs. Slamun, M.Comp.Sc., Ph.D selaku Dosen Pembimbing Utama, Muhammad Arif Hidayat S.Kom,M.Kom selaku Dosen Pembimbing Anggota, yang telah memberikan banyak arahan dan bimbingan dalam penulisan skripsi ini
3. Anang Andrianto ST.,MT, selaku dosen penguji I, dan Yanuar Nurdiansyah ST.,M.Cs. selaku dosen penguji II yang telah memberikan masukan dalam penulisan skripsi ini;
4. Ayah Alip Susanto dan Ibu Anik Suryani serta seluruh pihak keluarga yang telah memberikan dukungan dan doa yang tulus;
5. Sahabat – sahabat terbaikku Program Studi Sistem Informasi angkatan 2011 Nefotion.
6. Semua pihak yang telah membantu baik tenaga maupun pikiran dalam pelaksanaan kegiatan penelitian dan penyusunan skripsi ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, Desember 2015

Penulis



**DAFTAR ISI**

|   |          |
|---|----------|
| PERSEMBAHAN.....  | ii       |
| PERNYATAAN .....  | iii      |
| PENGESAHAN PEMBIMBING .....                                       | iv       |
| PENGESAHAN.....   | vi       |
| RINGKASAN.....  | vii      |
| PRAKATA .....   | viii     |
| DAFTAR ISI.....   | x        |
| DAFTAR TABEL .....  | xiii     |
| DAFTAR GAMBAR.....  | xiv      |
| <b>BAB 1      PENDAHULUAN.....</b>                                | <b>1</b> |
| 1.1. Latar Belakang.....  | 1        |
| 1.2. Rumusan Masalah.....   | 3        |
| 1.3. Tujuan dan Manfaat .....                                     | 3        |
| 1.3.1. Tujuan .....   | 3        |
| 1.3.2. Manfaat .....  | 4        |
| 1.4. Batasan Masalah.....   | 4        |
| 1.5. Sistematika Penulisan .....                                  | 5        |
| <b>BAB 2      TINJAUAN PUSTAKA .....</b>                          | <b>6</b> |
| 2.1. Perpustakaan.....  | 6        |
| 2.2. Sistem Informasi.....  | 7        |
| 2.3. Datamining .....   | 7        |
| 2.4. Algoritma Apriori .....                                      | 8        |
| 2.4.1. Perbandingan Algoritma pada Association Rules .....        | 15       |
| 2.4.2. Kesimpulan terhadap Algoritma pada Association Rules ..... | 18       |
| 2.5. Frequent itemset .....                                       | 20       |
| 2.6. Metode Pengembangan Sistem.....                              | 22       |
| 2.7. Analisis Kebutuhan.....                                      | 23       |

|              |   |           |
|--------------|---|-----------|
| 2.8.         | Perancangan Sistem .....                      | 23        |
| 2.9.         | Implementasi .....                            | 24        |
| 2.10.        | Pengujian .....                               | 24        |
| 2.10.1.      | Pengujian White Box .....                     | 25        |
| 2.10.2.      | Pengujian Black Box .....                     | 26        |
| 2.11.        | Penelitian Terdahulu .....                    | 27        |
| <b>BAB 3</b> | <b>METODOLOGI PENELITIAN .....</b>            | <b>28</b> |
| 3.1.         | Alur Penelitian.....                          | 28        |
| 3.2.         | Jenis Penelitian .....                        | 29        |
| 3.3.         | Pengumpulan Data.....                         | 29        |
| 3.3.1.       | Tempat Penelitian .....                       | 29        |
| 3.3.2.       | Alat Penelitian .....                         | 30        |
| 3.4.         | Pembuatan Sistem.....                         | 30        |
| 3.5.1.       | Analisa Kebutuhan .....                       | 30        |
| 3.5.2.       | Desain Sistem .....                           | 32        |
| 3.5.3.       | Penulisan Kode Program.....                   | 33        |
| 3.5.4.       | Pengujian Sistem .....                        | 33        |
| <b>BAB 4</b> | <b>ANALISIS DAN PENGEMBANGAN SISTEM .....</b> | <b>34</b> |
| 4.1.         | Analisis Kebutuhan Perangkat Lunak.....       | 34        |
| 4.1.1.       | Kebutuhan Fungsional .....                    | 34        |
| 4.1.2.       | Kebutuhan Non-Fungsional .....                | 35        |
| 4.2.         | Desain Sistem .....                           | 35        |
| 4.2.1.       | Business Process.....                         | 35        |
| 4.2.2.       | Usecase Diagram .....                         | 36        |
| 4.2.3.       | Skenario .....                                | 38        |
| 4.2.4.       | Activity Diagram .....                        | 60        |
| 4.2.5.       | Sequence Diagram .....                        | 69        |
| 4.2.6.       | Class Diagram .....                           | 81        |

|        |   |     |
|--------|---|-----|
| 4.2.7. | <i>Entity Relation Diagram</i> .....          | 82  |
| 4.3.   | Implementasi Perancangan Sistem .....         | 82  |
| 4.4.   | Pengujian Sistem .....                        | 83  |
| BAB 5  | HASIL DAN PEMBAHASAN .....                    | 89  |
| 5.1.   | Implementasi LIS .....                        | 89  |
| 5.1.1. | Halaman Utama LIS .....                       | 89  |
| 5.1.2. | Halaman Awal Admin .....                      | 92  |
| 5.1.3. | Manajemen User LIS .....                      | 92  |
| 5.1.4. | Manajemen Data Buku .....                     | 95  |
| 5.1.5. | Halaman Awal User (Pegawai) .....             | 97  |
| 5.1.6. | Pencarian Buku pada Pegawai .....             | 98  |
| 5.1.7. | Menu Peminjaman dan Pengembalian Buku .....   | 99  |
| 5.1.8. | Menu Rekomendasi Buku .....                   | 100 |
| 5.2.   | Implementasi Algoritma Apriori pada LIS ..... | 101 |
| BAB 6  | PENUTUP .....                                 | 109 |
| 6.1.   | Kesimpulan .....                              | 109 |
| 6.2.   | Saran .....                                   | 109 |
|        | DAFTAR PUSTAKA .....                          | 110 |
|        | LAMPIRAN .....                                | 112 |

## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 4.1 Definisi Aktor Usecase .....                                  | 37 |
| Tabel 4.2 Definisi Usecase .....  | 37 |
| Tabel 4.3 Skenario login Admin.....                                     | 38 |
| Tabel 4.4 Skenario Manajemen Data User.....                             | 41 |
| Tabel 4.5 Skenario Manajemen Data Mahasiswa .....                       | 43 |
| Tabel 4.6 Skenario Manajemen Data Buku .....                            | 46 |
| Tabel 4.7 Skenario login pegawai .....                                  | 50 |
| Tabel 4.8 Skenario Pencarian Buku.....                                  | 52 |
| Tabel 4.9 Skenario Manajemen Data Peminjaman dan Pengembalian .....     | 53 |
| Tabel 4.10 Skenario Pencarian Rekomendasi Buku .....                    | 56 |
| Tabel 4.11 Skenario Pencarian Buku oleh Mahasiswa .....                 | 58 |
| Tabel 4.12 <i>Test Case</i> Perhitungan Algoritma Apriori .....         | 86 |
| Tabel 4.13 Pengujian blackbox terhadap pencarian rekomendasi buku ..... | 87 |

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 2.1 FP-Tree dan CFP-Tree atau CT-Pro .....                        | 19 |
| Gambar 2.2 Hasil pengujian algoritma pada <i>Association Rules</i> ..... | 20 |
| Gambar 2.3 Proses Pencarian Maximum Frequent Itemsets.....               | 21 |
| Gambar 2.4 Model Waterfall.....  | 23 |
| Gambar 3.1 <i>Flowchart</i> Alur Penelitian.....                         | 28 |
| Gambar 3.2 <i>Flowchart</i> penerapan algoritma apriori .....            | 31 |
| Gambar 4.1 <i>Business Process</i> Sistem Informasi Perpustakaan.....    | 36 |
| Gambar 4.2 Usecase Sistem Informasi Perpustakaan.....                    | 36 |
| Gambar 4.3 Activity Diagram Login Admin .....                            | 61 |
| Gambar 4.4 Activity Diagram Login User.....                              | 62 |
| Gambar 4.5 Activity Diagram Logout .....                                 | 62 |
| Gambar 4.6 Activity Diagram Manajemen Data User LIS.....                 | 63 |
| Gambar 4.7 Activity Diagram Manajemen Data Mahasiswa.....                | 64 |
| Gambar 4.8 Activity Diagram Manajemen Data Buku .....                    | 65 |
| Gambar 4.9 Activity Diagram Pencarian Buku .....                         | 66 |
| Gambar 4.10 Activity Diagram Peminjaman Buku.....                        | 66 |
| Gambar 4.11 Activity Diagram Pengembalian Buku .....                     | 67 |
| Gambar 4.12 Pencarian Buku oleh Mahasiswa .....                          | 68 |
| Gambar 4.13 Activity Diagram Rekomendasi Buku .....                      | 69 |
| Gambar 4.14 Sequence Diagram Login Admin .....                           | 70 |
| Gambar 4.15 Sequence Diagram Login Pegawai .....                         | 71 |
| Gambar 4.16 Sequence Tambah Data User .....                              | 72 |
| Gambar 4.17 Sequence Edit Data User.....                                 | 73 |
| Gambar 4.18 Sequence Tambah Data Mahasiswa .....                         | 74 |
| Gambar 4.19 Sequence Edit Data Mahasiswa.....                            | 75 |
| Gambar 4.20 Sequence Tambah Data Buku .....                              | 76 |
| Gambar 4.21 Sequence Edit Data Buku.....                                 | 77 |
| Gambar 4.22 Sequence Diagram Peminjaman Buku.....                        | 78 |
| Gambar 4.23 Sequence Diagram Pengembalian Buku .....                     | 79 |
| Gambar 4.24 Sequence Pencarian Buku Beserta Rekomendasi.....             | 80 |
| Gambar 4.25 Class Diagram .....  | 81 |
| Gambar 4.26 Entity Relation Diagram .....                                | 82 |
| Gambar 4.27 Listing Program .....  | 84 |
| Gambar 4.28 Grafik Alir Fungsi Deskripsi.....                            | 85 |

## BAB 1 PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir ini. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan.

### 1.1. Latar Belakang

Dunia pendidikan dan informasi merupakan topik yang tidak asing lagi bagi seluruh umat manusia dimana pendidikan dan informasi tersebut mempengaruhi perkembangan dalam kehidupan. Dengan adanya pendidikan yang sudah ada sejak dahulu hingga sekarang yang selalu berkembang, manusia menjadi lebih pintar dan dapat mengetahui sejarah, logika dan teori yang ada pada alam kehidupannya, dengan informasi – informasi yang didapat, semakin banyak informasi yang diperoleh, semakin luas wawasan yang diketahui oleh umat manusia tersebut. Tentunya dalam pendidikan dan informasi tersebut perlu adanya wadah tertentu sebagai rumah informasi sehingga ketika masyarakat suatu saat membutuhkan pengetahuan, masyarakat tersebut bisa langsung menuju ke rumah informasi tersebut. Muncul lah perpustakaan, rumah atau gudang ilmu yang berisi informasi – informasi dan pendidikan yang bersedia memberikan pengetahuan kepada masyarakat tentang apa yang masyarakat tersebut butuhkan.

Dalam dunia perpustakaan terdapat banyak pengetahuan yang dapat dipelajari baik dari ilmu luar maupun ilmu dalam, menurut UU Perpustakaan pada Bab I pasal 1 menyatakan bahwa Perpustakaan adalah institusi yang mengumpulkan pengetahuan tercetak dan terekam, mengelolanya dengan cara khusus guna memenuhi kebutuhan intelektualitas para penggunanya melalui beragam cara interaksi pengetahuan. Semakin berkembang dunia teknologi, kini perpustakaan pun juga bisa menyimpan data – data informasi berupa data map, cetak, atau hasil seni lainnya, *mikrofilm*, *mikrofiche*, *tape audio*, CD, LP, DVD, *tape video*, dan menyediakan fasilitas umum untuk mengakses gudang data CD-ROM dan internet.

Jadi di setiap tempat atau wilayah, perlu adanya perpustakaan sebagai gudang tempat masyarakat tersebut menggali ilmu pengetahuan. Contohnya di Universitas Jember, terdapat puluhan ribu mahasiswa yang berkumpul guna mencari pengetahuan dan gelar, maka Universitas Jember menyediakan atau membangun perpustakaan Universitas Jember guna sebagai gudang ilmu yang nantinya akan membantu mahasiswa dalam proses mencari pengetahuan dan gelar tersebut.

Setiap tahunnya, mahasiswa dari berbeda angkatan menempuh mata kuliah yang sama, dan tentunya mahasiswa tersebut membutuhkan referensi berupa informasi – informasi dan pengetahuan lebih mengenai mata kuliah yang saat itu mereka tempuh. Terdapat kemungkinan bahwa dalam setiap semesternya, ribuan mahasiswa tersebut melakukan peminjaman buku ke perpustakaan guna mencari informasi tambahan, sehingga muncul lah pengulangan terhadap buku – buku yang dipinjam dalam setiap periodenya, dalam pengulangan atau peminjaman dengan relasi yang sama antara buku 1 dengan buku yang lainnya, bisa diperoleh informasi yang bisa berguna untuk peminjam selanjutnya dalam memilih buku apa saja yang sesuai berupa rekomendasi buku yang sering berkaitan 1 sama lain. Dengan adanya rekomendasi buku yang berkaitan ini, peminjam mendapatkan informasi lebih mengenai buku yang diperoleh.

Namun untuk saat ini, perpustakaan lebih memberikan informasi mengenai buku yang terkait berdasarkan judul ataupun mata pelajaran, tetapi kemungkinan bahwa buku yang pembahasannya mengenai informasi tertentu justru mendapatkan relasi yang lebih terhadap buku lain, dikarenakan luasnya informasi sehingga butuh referensi dalam pengetahuan tersebut. Dalam hal ini, dapat disimpulkan bahwa relasi antar buku tidak hanya berdasarkan judul ataupun tema, dan informasi tersebut bisa didapat dengan memanfaatkan data transaksi sehingga memunculkan persentase seberapa sering relasi buku tersebut terhubung satu sama lain dan seberapa besar kemungkinan buku rekomendasi tersebut mendukung buku yang dipinjam, ini justru sangat bermanfaat bagi mahasiswa yang sedang menempuh suatu mata kuliah, sehingga mahasiswa tersebut mendapatkan informasi lebih mengenai buku – buku

apa saja yang sering dipinjam oleh angkatan sebelumnya yang berelasi dengan buku yang dipinjam oleh mahasiswa tersebut.

Berdasarkan permasalahan di atas, maka peneliti membuat “Sistem Pengklasifikasian Buku Perpustakaan Universitas Jember dengan Rekomendasi Buku” untuk mahasiswa guna membantu memberikan informasi tentang buku – buku yang dipinjam secara bersamaan yang memiliki kemungkinan saling berkaitan pada 1 mata kuliah yang berhubungan dengan buku utama, pada sistem ini menggunakan Algoritma Apriori dengan memanfaatkan *Frequent itemset* sehingga mendapatkan informasi mengenai nilai *Support* dan *Confidence*.

## 1.2. Rumusan Masalah

Berdasarkan uraian di atas, maka dirumuskan permasalahan sebagai berikut :

1. Bagaimana merancang dan menggunakan sebuah sistem pengklasifikasian rekomendasi buku menggunakan algoritma apriori.
2. Bagaimana membangun sistem pengklasifikasian rekomendasi buku perpustakaan pada perpustakaan di Universitas Jember.

## 1.3. Tujuan dan Manfaat

Tujuan dan manfaat pada bab ini berisi tentang tujuan penelitian dari implementasi metode Algoritma Apriori untuk mendapatkan hasil output berupa rekomendasi buku yang berhubungan dengan buku yang dipinjam oleh mahasiswa. Sedangkan pada bagian manfaat berisi tentang manfaat apa saja yang akan diperoleh pada penelitian ini, baik bagi peneliti sendiri maupun bagi objek pada penelitian ini.

### 1.3.1. Tujuan

Dari beberapa rumusan masalah di atas, terdapat tujuan yang ingin dicapai dari penelitian ini, yaitu :

1. Merancang dan menggunakan sebuah sistem klasifikasi rekomendasi buku perpustakaan pada Universitas Negeri Jember menggunakan algoritma apriori dan menghasilkan output berupa rekomendasi buku.

2. Membangun sistem pengklasifikasian rekomendasi buku perpustakaan pada Universitas Jember.

### 1.3.2. Manfaat

Dari project berdasarkan rumusan masalah diatas, terdapat beberapa manfaat diantaranya manfaat akademis, bagi peneliti, dan objek peneliti, yaitu :

1. Manfaat Akademis

Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Universitas Jember.

2. Manfaat bagi peneliti

Mengetahui bagaimana proses penerapan metode Algoritma Apriori untuk mencari hasil rekomendasi suatu buku yg berkaitan serta membantu perpustakaan Universitas Jember dalam memberikan pelayanan peminjaman suatu buku.

3. Manfaat bagi objek penelitian

Memberikan informasi baru kepada perpustakaan Universitas Jember bahwa dengan penggunaan algoritma ini, dapat meningkatkan informasi mengenai buku yang berkaitan, dan dapat membantu perpustakaan Universitas Jember dalam menjalankan perannya.

### 1.4. Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Data rekomendasi buku diperoleh dari hasil perhitungan algoritma apriori terhadap data transaksi.
2. Implementasi sistem dirancang berbasis web.
3. Metode algoritma apriori menggunakan perhitungan nilai *support* dan *confidence* dalam memberikan rekomendasi buku.

4. Akurasi dalam pemberian rekomendasi didapatkan pada sering tidaknya buku utama dengan buku hasil rekomendasi tersebut.

## **1.5. Sistematika Penulisan**

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut:

1. **Pendahuluan**  
Bab ini menjelaskan latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan.
2. **Tinjauan Pustaka**  
Bab ini berisi materi, informasi, dan kajian teori yang digunakan dalam penulisan.
3. **Metodologi Penelitian**  
Bab ini menjelaskan tentang metodologi penelitian yang digunakan selama proses penulisan berlangsung.
4. **Analisis dan Pengembangan Sistem**  
Bab ini menjelaskan tentang analisis dan pengembangan sistem yang dibangun.
5. **Hasil dan Pembahasan**  
Bab ini berisi tentang hasil dan pembahasan dari system yang sudah dibangun.
6. **Penutup**  
Bab ini berisi tentang kesimpulan dari penulis dan saran untuk penulis selanjutnya.

## BAB 2 TINJAUAN PUSTAKA

Tinjauan pustaka akan dijelaskan teori – teori dan pustaka yang digunakan dalam penelitian. Teori yang dibahas adalah teori tentang perpustakaan, sistem informasi, *Datamining*, *Apriori Algorithm*, *Frequent itemset*, metode pengembangan sistem, analisis kebutuhan, desain sistem, implementasi, pengujian, serta penelitian terdahulu tentang penggunaan metode Algoritma Apriori.

### 2.1. Perpustakaan

Dalam UU No.43 tahun 2007 tentang perpustakaan disebutkan bahwa Perpustakaan adalah institusi pengelola koleksi karya tulis, karya cetak,dan/atau karya rekam secara profesional dengan sistem yang baku.

Perpustakaan merupakan unit yang bertugas untuk menghimpun, mengolah dan menyebarkan informasi dan atau literatur kepada masyarakat di lingkungannya. Menurut definisi ini jelas bahwa semua yang berkaitan dengan pengelolaan informasi yang dapat berupa karya tulis, karya cetak dan karya rekam dimandatkan kepada perpustakaan. (Saleh, 2009).

Di Universitas Jember terdapat sebuah perpustakaan yang sangat berperan dalam membagi ilmu kepada mahasiswa di universitas tersebut, pengolahan serta proses peminjaman berjalan lancar sesuai tugas. Disana tidak hanya terdapat buku, melainkan media lain juga tersedia untuk dipelajari bersama. Adapun tujuan dari perpustakaan pada sekolah, diantaranya (Heriyanto, 2013) :

1. Mendorong dan mempercepat proses penguasaan teknik membaca para siswa.
2. Membantu menulis kreatif siswa dengan bimbingan guru dan pustakawan.
3. Menumbuhkan minat baca siswa.
4. Menyediakan berbagai informasi yang sesuai dengan kurikulum sekolah.
5. Mendorong, menggairahkan, memelihara, dan member semangat membaca dan semangat belajar bagi siswa.

6. Memperluas, memperdalam, dan memperkaya pengalaman belajar para siswa dengan membaca buku dan koleksi lain yang mengandung ilmu pengetahuan dan teknologi, yang disediakan oleh perpustakaan.
7. Memberikan hiburan sehat untuk mengisi waktu senggang melalui kegiatan membaca.

Dari beberapa tujuan perpustakaan pada sekolah tersebut, yang dapat kita tarik kesimpulan pada studi kasus perpustakaan untuk mahasiswa di Universitas Negeri Jember bahwa dengan tercapainya tujuan ini perpustakaan telah berhasil melakukan tugasnya, disini penulis merancang sistem guna membantu mewujudkan dan memberi sarana tambahan dari tujuan tersebut.

## **2.2. Sistem Informasi**

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. (Antonio & Safriadi, 2012).

Pada sistem yang akan dikembangkan oleh penulis ini bersifat memberikan informasi yang nantinya akan berguna untuk perpustakaan. Khususnya pada rekomendasi peminjaman buku, dimana sistem akan memberi informasi mengenai buku – buku apa saja yang berkaitan dengan buku yang akan dipinjam oleh peminjam buku tersebut.

## **2.3. Datamining**

Datamining adalah ekstraksi informasi implisit yang tidak mudah, yang sebelumnya tidak diketahui dan sangat berpotensi dalam hal kegunaan. Teknologi datamining memberikan pendekatan yang berorientasi user-oriented baru yang tersembunyi di dalam data. (Srinivas, Rao, & Govardhan, 2011).

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 bukanlah suatu bidang yang sama sekali baru. Salah satu kesulitan untuk mendefinisikan data mining adalah kenyataan bahwa data mining mewarisi banyak aspek dan teknik dari bidang-bidang ilmu yang sudah mapan terlebih dahulu. (Syaifullah, 2010).

Dalam perkembangan teknologi datamining sudah muncul model atau mode – mode yang digunakan untuk melakukan proses penggalian informasi terhadap data – data yang sudah ada, pada studi kasus ini datamining sangatlah berperan penting dikarenakan dibutuhkannya penggalian data – data pada perpustakaan guna mencari hubungan atau relasi buku yang akan dipinjamkan.

## 2.4. Algoritma Apriori

Menurut (Syaifullah, 2010) dalam (Santosa, 2007) Algoritma apriori termasuk jenis aturan asosiasi pada data mining. Selain apriori, yang termasuk pada golongan ini adalah metode *Generalized Rule Induction* dan *Algoritma Hash Based*. Aturan yang menyatakan asosiasi antara beberapa atribut sering disebut *affinity analysis* atau *market basket analysis*. Analisis asosiasi atau association rule adalah teknik data mining untuk menemukan aturan asosiatif antara suatu kombinasi item.

Algoritma Apriori yang bertujuan untuk menemukan frequent itemsets dijalankan pada sekumpulan data. Analisis Apriori didefinisikan suatu proses untuk menemukan semua aturan apriori yang memenuhi syarat minimum untuk support dan syarat minimum untuk confidence.

Support adalah nilai penunjang, atau persentase kombinasi sebuah item dalam database. Rumus support sebagai berikut:

$$Support(A) = \left( \frac{\sum trans(A)}{\sum Trans} \right) \times 100\% \dots \dots \dots \text{persamaan(I)}$$

Keterangan:

- Support(A) : Hasil perhitungan support A
- $\sum A$  : Jumlah transaksi mengandung A
- $\sum Trans$  : Total transaksi

Sedangkan confidence adalah nilai kepastian yaitu kuatnya hubungan antar item dalam sebuah apriori. Confidence bisa dicari setelah pola frekuensi munculnya sebuah item ditemukan. Rumus untuk menghitung confidence adalah sebagai berikut, contoh misal ditemukan aturan  $A \rightarrow B$  maka:

$$\text{Confidence } P(B|A) = \left( \frac{\sum \text{Trans}(A|B)}{\sum \text{Trans}(A)} \right) \times 100 \% \dots \dots \dots \text{persamaan (II)}$$

Keterangan:

- Confidence  $P(B|A)$  : Hasil perhitungan Confidence  $A \rightarrow B$
- $\sum \text{Trans}(A|B)$  : Total transaksi mengandung A dan B
- $\sum \text{Trans}(A)$  : Transaksi mengandung A

Contoh transaksi peminjaman, yang ditampilkan pada tabel di bawah ini.

Tabel 1

| Transaksi | Item yang dipinjam |
|-----------|--------------------|
| 1         | A, B, C, E         |
| 2         | C, D, E            |
| 3         | A, B, C, D         |
| 4         | A, B, E            |
| 5         | D, E               |
| 6         | C, D, E            |

Lalu transaksi tabel 1 dipisahkan masing – masing item yang dipinjam.

Tabel 2

| Item yang dipinjam |
|--------------------|
| A                  |
| B                  |
| C                  |
| D                  |
| E                  |

Kemudian pada tabel 2 dibuat tabel 3 seperti di bawah ini dan kemudian dihitung jumlah transaksi total yang ada.

Tabel 3

| Transaksi | A | B | C | D | E |
|-----------|---|---|---|---|---|
| 1         | 1 | 1 | 1 | 0 | 1 |
| 2         | 0 | 0 | 1 | 1 | 1 |
| 3         | 1 | 1 | 1 | 1 | 0 |
| 4         | 1 | 1 | 0 | 0 | 1 |
| 5         | 0 | 0 | 0 | 1 | 1 |
| 6         | 0 | 0 | 1 | 1 | 1 |
| $\Sigma$  | 3 | 3 | 4 | 4 | 5 |

Tentukan nilai  $\phi$

Misalkan nilai  $\phi$  ditentukan sebesar 3, maka dengan ini dapat menentukan frequent itemset. Dari tabel di atas diketahui total  $\phi$  untuk transaksi  $k = 1$ , semuanya lebih besar dari  $\phi$ , maka:

$$F_1 = \{(A), (B), (C), (D), (E)\}$$

Untuk transaksi  $k = 2$  (2 unsur), diperlukan tabel untuk tiap – tiap pasang item. Himpunan yang mungkin terbentuk adalah:  $\{A,B\}$ ,  $\{A,C\}$ ,  $\{A,D\}$ ,  $\{A,E\}$ ,  $\{B,C\}$ ,  $\{B,D\}$ ,  $\{B,E\}$ ,  $\{C,D\}$ ,  $\{C,E\}$ ,  $\{D,E\}$

Dari hasil di atas dapat dibuat tabel untuk calon 2 item set

Tabel 4-10

| T | A | B        | F |
|---|---|----------|---|
| 1 | 1 | 1        | P |
| 2 | 0 | 0        | S |
| 3 | 1 | 1        | P |
| 4 | 1 | 1        | P |
| 5 | 0 | 0        | S |
| 6 | 0 | 0        | S |
|   |   | $\Sigma$ | 3 |

| T | A | C        | f |
|---|---|----------|---|
| 1 | 1 | 1        | p |
| 2 | 0 | 1        | S |
| 3 | 1 | 1        | P |
| 4 | 1 | 0        | S |
| 5 | 0 | 0        | S |
| 6 | 0 | 1        | S |
|   |   | $\Sigma$ | 2 |

| T | A | D        | F |
|---|---|----------|---|
| 1 | 1 | 0        | S |
| 2 | 0 | 1        | S |
| 3 | 1 | 1        | P |
| 4 | 1 | 0        | S |
| 5 | 0 | 1        | S |
| 6 | 0 | 1        | S |
|   |   | $\Sigma$ | 1 |

| T | A | E        | F |
|---|---|----------|---|
| 1 | 1 | 1        | P |
| 2 | 0 | 1        | S |
| 3 | 1 | 0        | S |
| 4 | 1 | 1        | P |
| 5 | 0 | 1        | S |
| 6 | 0 | 1        | S |
|   |   | $\Sigma$ | 2 |

| T | B | C        | f |
|---|---|----------|---|
| 1 | 1 | 1        | p |
| 2 | 0 | 1        | S |
| 3 | 1 | 1        | P |
| 4 | 1 | 0        | S |
| 5 | 0 | 0        | S |
| 6 | 0 | 1        | S |
|   |   | $\Sigma$ | 2 |

| T | B | D        | F |
|---|---|----------|---|
| 1 | 1 | 0        | S |
| 2 | 0 | 1        | S |
| 3 | 1 | 1        | P |
| 4 | 1 | 0        | S |
| 5 | 0 | 1        | S |
| 6 | 0 | 1        | S |
|   |   | $\Sigma$ | 1 |

| T | B | E        | F |
|---|---|----------|---|
| 1 | 1 | 1        | P |
| 2 | 0 | 1        | S |
| 3 | 1 | 0        | S |
| 4 | 1 | 1        | P |
| 5 | 0 | 1        | S |
| 6 | 0 | 1        | S |
|   |   | $\Sigma$ | 2 |

| T | C | D        | f |
|---|---|----------|---|
| 1 | 1 | 0        | S |
| 2 | 1 | 1        | P |
| 3 | 1 | 1        | P |
| 4 | 0 | 0        | S |
| 5 | 0 | 1        | S |
| 6 | 1 | 1        | P |
|   |   | $\Sigma$ | 3 |

| T | C | E        | F |
|---|---|----------|---|
| 1 | 1 | 1        | P |
| 2 | 1 | 1        | P |
| 3 | 1 | 0        | S |
| 4 | 0 | 1        | S |
| 5 | 0 | 1        | S |
| 6 | 1 | 1        | P |
|   |   | $\Sigma$ | 3 |

| T | D | E        | f |
|---|---|----------|---|
| 1 | 0 | 1        | S |
| 2 | 1 | 1        | P |
| 3 | 1 | 0        | S |
| 4 | 0 | 1        | S |
| 5 | 1 | 1        | P |
| 6 | 1 | 1        | P |
|   |   | $\Sigma$ | 3 |

Pada tabel 4 – 10 di atas, P memiliki arti bahwa item – item tersebut dipinjam bersamaan, sedangkan S berarti tidak ada item yang dipinjam bersamaan atau tidak ada transaksi.

$\sum$  melambangkan jumlah frequent itemset pada setiap tabelnya. Jumlah frequent itemset harus lebih besar atau sama dengan jumlah frequent itemset yang telah ditentukan ( $\sum \geq \phi$ ). Dari tabel di atas, maka didapat:

$$F2 = \{\{A,B\}, \{C,D\}, \{C,E\}, \{D,E\}\}$$

Kombinasi dari itemset pada F2, dapat digabungkan menjadi calon 3-itemset. Itemset – itemset yang dapat digabungkan adalah itemset – itemset yang memiliki kesamaan dalam k-1 item pertama. Untuk k = 3, himpunan yang mungkin terbentuk adalah: {C,D,E}

Dari tabel 11 di atas, didapat F3 = {} dikarenakan jumlah frequent itemset dibawah frequent itemset yang telah ditentukan, sehingga F4, F5, dan F6 juga merupakan himpunan kosong.

Tabel 11

| T | C | D | E      | f |
|---|---|---|--------|---|
| 1 | 1 | 0 | 1      | s |
| 2 | 1 | 1 | 1      | P |
| 3 | 1 | 1 | 0      | s |
| 4 | 0 | 0 | 1      | s |
| 5 | 0 | 1 | 1      | s |
| 6 | 1 | 1 | 1      | P |
|   |   |   | $\sum$ | 2 |

Tentukan (ss-s) sebagai antecedent dan s sebagai consequent dari Fk yang telah didapat. Pada F2 didapat himpunan F2 = {A,B}, {C,D}, {C,E}, {D,E}

Maka dapat disusun:

Untuk {A,B}

Jika (ss-s) = A, Jika s = B, Maka  $\rightarrow$  if pinjam A then pinjam B

Jika (ss-s) = B, Jika s = A, Maka  $\rightarrow$  if pinjam B then pinjam A

Untuk {C,D}

Jika (ss-s) = C, Jika s = D, Maka  $\rightarrow$  if pinjam C then pinjam D

Jika (ss-s) = D, Jika s = C, Maka  $\rightarrow$  if pinjam D then pinjam C

Untuk {C,E}

Jika (ss-s) = C, Jika s = E, Maka  $\rightarrow$  if pinjam C then pinjam E

Jika (ss-s) = E, Jika s = C, Maka  $\rightarrow$  if pinjam E then pinjam C

Untuk {D,E}

Jika (ss-s) = D, Jika s = E, Maka  $\rightarrow$  if pinjam D then pinjam E

Jika (ss-s) = E, Jika s = D, Maka  $\rightarrow$  if pinjam E then pinjam D

Dari langkah di atas, bisa didapatkan 8 rule yang dapat digunakan, yaitu:

If pinjam A then pinjam B

If pinjam B then pinjam A

If pinjam C then pinjam D

If pinjam D then pinjam C

If pinjam C then pinjam E

If pinjam E then pinjam C

If pinjam D then pinjam E

If pinjam E then pinjam D

Hitung Support dan Confidence

$$Support(A) = \left( \frac{\sum trans(A)}{\sum Trans} \right) \times 100\%.$$

Keterangan:

Support(A) : Hasil perhitungan support A

$\sum A$  : Jumlah transaksi mengandung A

$\sum Trans$  : Total transaksi

$$\text{Confidence } P(B|A) = \left( \frac{\sum \text{Trans}(A|B)}{\sum \text{Trans}(A)} \right) \times 100 \%$$

Keterangan:

Confidence  $P(B|A)$  : Hasil perhitungan Confidence  $A \rightarrow B$

$\sum \text{Trans}(A|B)$  : Total transaksi mengandung A dan B

$\sum \text{Trans}(A)$  : Transaksi mengandung A

Sehingga didapat tabel sebagai berikut:

| If antecedent then consequent | Support                        | Confidence                   |
|-------------------------------|--------------------------------|------------------------------|
| If pinjam A then pinjam B     | $(3/6) \times 100\% = 50\%$    | $(3/3) \times 100\% = 100\%$ |
| If pinjam B then pinjam A     | $(3/6) \times 100\% = 50\%$    | $(3/3) \times 100\% = 100\%$ |
| If pinjam C then pinjam D     | $(4/6) \times 100\% = 66,67\%$ | $(3/4) \times 100\% = 75\%$  |
| If pinjam D then pinjam C     | $(4/6) \times 100\% = 66,67\%$ | $(3/4) \times 100\% = 75\%$  |
| If pinjam C then pinjam E     | $(4/6) \times 100\% = 66,67\%$ | $(3/4) \times 100\% = 75\%$  |
| If pinjam E then pinjam C     | $(5/6) \times 100\% = 83,33\%$ | $(3/5) \times 100\% = 60\%$  |
| If pinjam D then pinjam E     | $(4/6) \times 100\% = 66,67\%$ | $(3/4) \times 100\% = 75\%$  |
| If pinjam E then pinjam D     | $(5/6) \times 100\% = 83,33\%$ | $(3/5) \times 100\% = 60\%$  |

Setelah di dapat support dan confidence untuk masing – masing kandidat, lakukan perkalian antara support dan confidence, dimana confidence nya diambil sample 70% ke atas, sehingga mendapatkan tabel seperti berikut:

| If antecedent then consequent | Support | Confidence | Support x Confidence |
|-------------------------------|---------|------------|----------------------|
| If pinjam A then pinjam B     | 50%     | 100%       | 0,5                  |
| If pinjam B then pinjam A     | 50%     | 100%       | 0,5                  |
| If pinjam C then pinjam D     | 66,67%  | 75%        | 0,50003              |

|                           |        |     |         |
|---------------------------|--------|-----|---------|
| If pinjam D then pinjam C | 66,67% | 75% | 0,50003 |
| If pinjam C then pinjam E | 66,67% | 75% | 0,50003 |
| If pinjam D then pinjam E | 66,67% | 75% | 0,50003 |

Setelah didapat hasil perkalian antara confidence dengan support, pilihlah hasil kali yang paling tinggi. Hasil paling tinggi tersebut merupakan rule yang dipakai pada saat melakukan transaksi peminjaman, sehingga rule nya adalah:

Jika meminjam buku C, maka dapat meminjam buku D dengan support 66,67% dan confidence 75%

Jika meminjam buku D, maka dapat meminjam buku C dengan support 66,67% dan confidence 75%

Jika meminjam buku C, maka dapat meminjam buku E dengan support 66,67% dan confidence 75%

Jika meminjam buku D, maka dapat meminjam buku E dengan support 66,67% dan confidence 75%

#### 2.4.1. Perbandingan Algoritma pada Association Rules

Pada *association rules* terdapat beberapa algoritma yang berbeda dalam melakukan proses datamining. Seiring berkembangnya program, muncul algoritma datamining yang baru dengan keunggulan masing – masing sehingga pengguna datamining dapat memilih yang mana yang menurut pengguna cocok dengan program tersebut.

##### 1. Algoritma Apriori

Ide dasarnya adalah menghitung pola kemunculan *item* yang muncul dalam data transaksi dengan beberapa iterasi. Iterasi ke *i* berarti mendapatkan semua *frequent i-itemset* (suatu *itemset* yang jumlah *item* anggotanya sejumlah *i*). Langkah umum tiap iterasi adalah menghasilkan *candidate itemset* kemudian dihitung nilai *support* dari tiap *candidate*. Untuk menghasilkan *candidate*, pada dasarnya dapat dilakukan dengan menyusun kombinasi *item-item* yang sudah ditemukan sebelumnya.

Algoritma ini didasari oleh hukum *apriori*, jika sebuah *itemset* ternyata *infrequent*, maka seharusnya *superset*-nya juga *infrequent* sehingga tidak perlu diperiksa lagi. Pada iterasi pertama, setiap jenis *item* yang ditemukan dalam data dijadikan *candidate* untuk *frequent 1-itemsets*. Sedangkan *candidate* di iterasi berikutnya didapatkan dari *frequent itemset* yang ditemukan di iterasi sebelumnya. Proses akan berhenti jika tidak ada lagi kombinasi *candidate* yang bisa dibuat. Berikut penjelasan tahapan dalam *frequent itemset mining* secara paralel:

1. *Matching* dilakukan dengan membaca tiap *item* yang sudah terurut pada tiap transaksi.
2. Tiap *candidate* memiliki *pointer* ke *item* berikutnya.
3. Bila *item* yang sedang ditunjuk tiap *pointer candidate* sama dengan *item* yang sedang dibaca pada transaksi, maka *pointer* pada *item candidate* tersebut bergeser ke *item candidate* berikutnya.
4. *Pointer* pada *candidate* menandakan sudah berapa banyak *item* yang ditemukan dari hasil membaca *item* pada transaksi
5. Jika selesai membaca semua *item* pada transaksi dan semua *item* di *candidate* ditemukan berarti *candidate* tersebut terdapat dalam transaksi tersebut.
6. Jika suatu *candidate* terdapat dalam suatu transaksi berarti nilai *support candidate* tersebut sama dengan satu untuk transaksi tersebut.

Setelah mendapatkan nilai *support* untuk tiap *candidate*, nilai *support* tersebut dibandingkan dengan nilai *minimum support* yang telah dimasukkan sebelumnya. Struktur data yang digunakan untuk menyimpan *frequent itemset* adalah *FItemset*. Struktur ini adalah struktur yang digunakan untuk menyimpan *candidate* yang telah dihitung nilai *support*-nya dan memenuhi nilai *minimum support*. *FItemset* disimpan referensinya ke sebuah *vector*. Tiap *FItemset* yang ditemukan dimasukkan ke dalam *vector* (hanya referensinya) sesuai ukuran dan *prefix*-nya. Misalnya “1 2 4” mendahului “1 2 5” karena 4 lebih kecil dari 5. Hal ini agar lebih memudahkan proses ketika melakukan *candidates generation*. Dimana bila “1 2 4” dan “1 2 5” diproses,

karena *prefix*-nya sama, maka *candidate* yang baru adalah “1 2” + “4” + “5”. Fitemset diurutkan berdasarkan ukuran dan *prefix*. Untuk penggunaan struktur lain seperti *candidate* hanya berupa *array of integer* yang disimpan referensinya saja pada sebuah *vector* untuk dihitung nilai *support*-nya. (Ruldeviyani & Fahrian, 2008).

## 2. Algoritma FP-Growth

Algoritma FP-Growth merepresentasikan transaksi dengan menggunakan struktur data *FP-Tree*. Proses penyusunan *FP-Tree* dari mulai representasi awal transaksi, pengurutan dengan hanya mempertahankan *frequent 1-itemset*, dan penyimpanannya di *FP-Tree*. Setelah *FP-Tree* terbentuk, langkah selanjutnya adalah memperoleh *frequent itemset* tanpa melakukan *candidates generation*. Berikut adalah tahapan dari algoritma FP-Growth:

### 1. Membuat *Header Item*

*Header* dalam hal ini selain sebagai *header* suatu *item* ke *FP-Tree* juga sebagai jenis *item* dasar yang memenuhi *minimum support*. Setelah mendapatkan *item* dan nilai *support*-nya, maka *item* yang tidak *frequent* dibuang dan *item* diurutkan berdasarkan nilai *support*-nya. *Header* untuk *item*, disiapkan pada suatu *array* tertentu dan ditambahkan ketika membuat *FP-Tree*.

### 2. Membuat *FP-Tree*

*FP-Tree* dibangun dengan mencari *item* sesuai urutan pada *item* yang *frequent*. Data transaksi tidak perlu diurutkan, dan untuk tiap *item* yang ditemukan bisa langsung dimasukkan ke dalam *FP-Tree*. Sesudah membuat *root*, tiap *item* yang ditemukan dimasukkan berdasarkan *path* pada *FP-Tree*. Jika *item* yang ditemukan sudah ada, maka nilai *support item* tersebut yang ditambahkan. Namun jika *path* belum ada, maka dibuat *node* baru untuk melengkapi *path* baru pada *FP-Tree* tersebut. Hal ini dilakukan selama *item* pada transaksi masih ada yang *qualified*, artinya memenuhi nilai *minimum support*. Jadi, *item-item* yang ditemukan dalam transaksi akan berurutan memanjang ke bawah. Dalam struktur *FP-Tree*,

diterapkan alur *path* dari *child* hingga ke *root*. Jadi, suatu *path* utuh dalam *FP-Tree* adalah dari *child* terbawah hingga ke *root*. Tiap *node* pada *FP-Tree* memiliki *pointer* ke *parent*, sehingga pencarian harus dimulai dari bawah.

### 3. *Pattern Extraction*

*Pattern extraction* dilakukan berdasarkan keterlibatan *item* pada suatu *path*. Di setiap *path*, diperiksa semua kombinasi yang mungkin dimana *item* tersebut terlibat. Di iterasi berikutnya dilakukan dengan melibatkan *item* berikutnya, tanpa melibatkan *item* sebelumnya, sehingga *pattern* yang sama tidak akan ditemukan dua kali pada *path* yang sama. Bila *item* pertama suatu hasil kombinasi bukan *item* terakhir (sebelum *root*), maka kombinasi *itemset* tersebut masih bisa dikembangkan lagi.

### 4. Memasukkan setiap *pattern* yang ditemukan ke dalam *PatternTree*

Setelah mengolah *FP-Tree* menjadi *pattern-pattern*, diperlukan proses akumulasi *pattern-pattern* yang ditemukan mengingat *pattern* yang sama dapat ditemukan pada *path* yang berbeda. Untuk itu digunakan struktur data *PatternTree*. Setiap *node* di *PatternTree* merepresentasikan dan menyimpan frekuensi suatu *pattern*. *PatternTree* terdiri atas *PatternTreeNode* yang menyimpan nilai *item*, nilai *support* dan dilengkapi dengan dua *pointer* yaitu untuk horisontal dan vertikal. (Ruldeviyani & Fahrian, 2008).

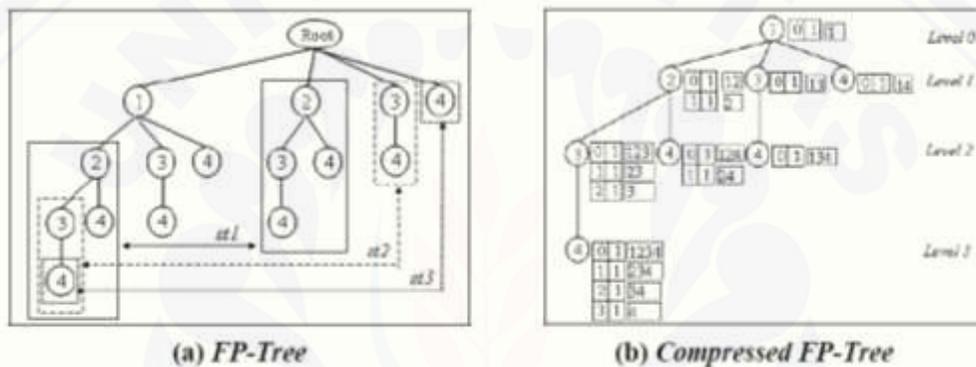
#### 2.4.2. Kesimpulan terhadap Algoritma pada Association Rules

Dapat ditarik kesimpulan bahwa algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. 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. Pada algoritma Apriori diperlukan generate candidate untuk mendapatkan frequent itemsets. Akan tetapi, di algoritma FP-Growth generate candidate tidak dilakukan

karena FP-Growth menggunakan konsep pembangunan tree dalam pencarian frequent itemsets.

### 3. Algoritma CT-Pro

Algoritma ini berakar dari FP-Growth dimana modifikasi yang dilakukan adalah pada struktur data yang digunakan. Struktur data yang digunakan adalah *Compressed FP-Tree (CFP-Tree)* dimana informasi dari sebuah *FP-Tree* diringkas dengan struktur yang lebih kecil, sehingga baik pembentukan *tree* maupun *frequent itemset mining* dapat dilakukan lebih cepat. (Ruldeviyani & Fahrian, 2008).



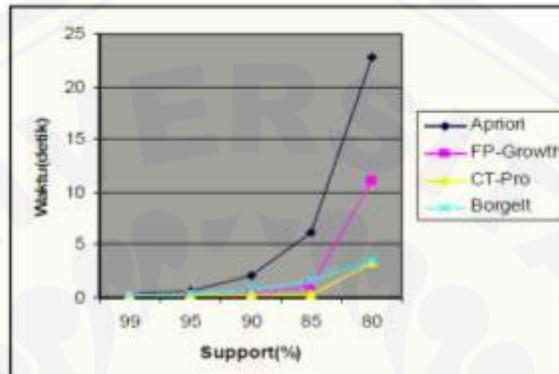
Gambar 2.1 FP-Tree dan CFP-Tree atau CT-Pro

(Ruldeviyani & Fahrian, 2008)

Pada gambar 2.1 terlihat perbedaan antara *FP-Tree*(a) dengan *FP-Tree* yang telah dimodifikasi dengan struktur yang lebih simple sehingga pembentukan *frequent itemsets* dapat dilakukan lebih cepat, disebut dengan *Compressed FP-Tree*(b).

Dari beberapa algoritma diatas, dilakukan pengujian yang terdapat proses mining suatu data di studi kasus tersebut. Dengan membuat beberapa studi kasus menggunakan beberapa input data sehingga algoritma – algoritma pada *Association Rules* bisa dibandingkan satu dengan yang lainnya, terdapat hasil perbandingan antar algoritma baik dari waktu perhitungan maupun dari hasil support yang dihasilkan. Hasil pengujian tersebut dapat dilihat pada gambar 2.2 di bawah ini.

| Support | Apriori | FP-Growth | CT-Pro | Apriori Borgelt |
|---------|---------|-----------|--------|-----------------|
| 99      | 0.36    | 0.27      | 0.22   | 0.27            |
| 95      | 0.66    | 0.28      | 0.23   | 0.42            |
| 90      | 2.07    | 0.34      | 0.24   | 0.8             |
| 85      | 6.2     | 1.09      | 0.26   | 1.57            |
| 80      | 22.81   | 11.07     | 3.27   | 3.63            |



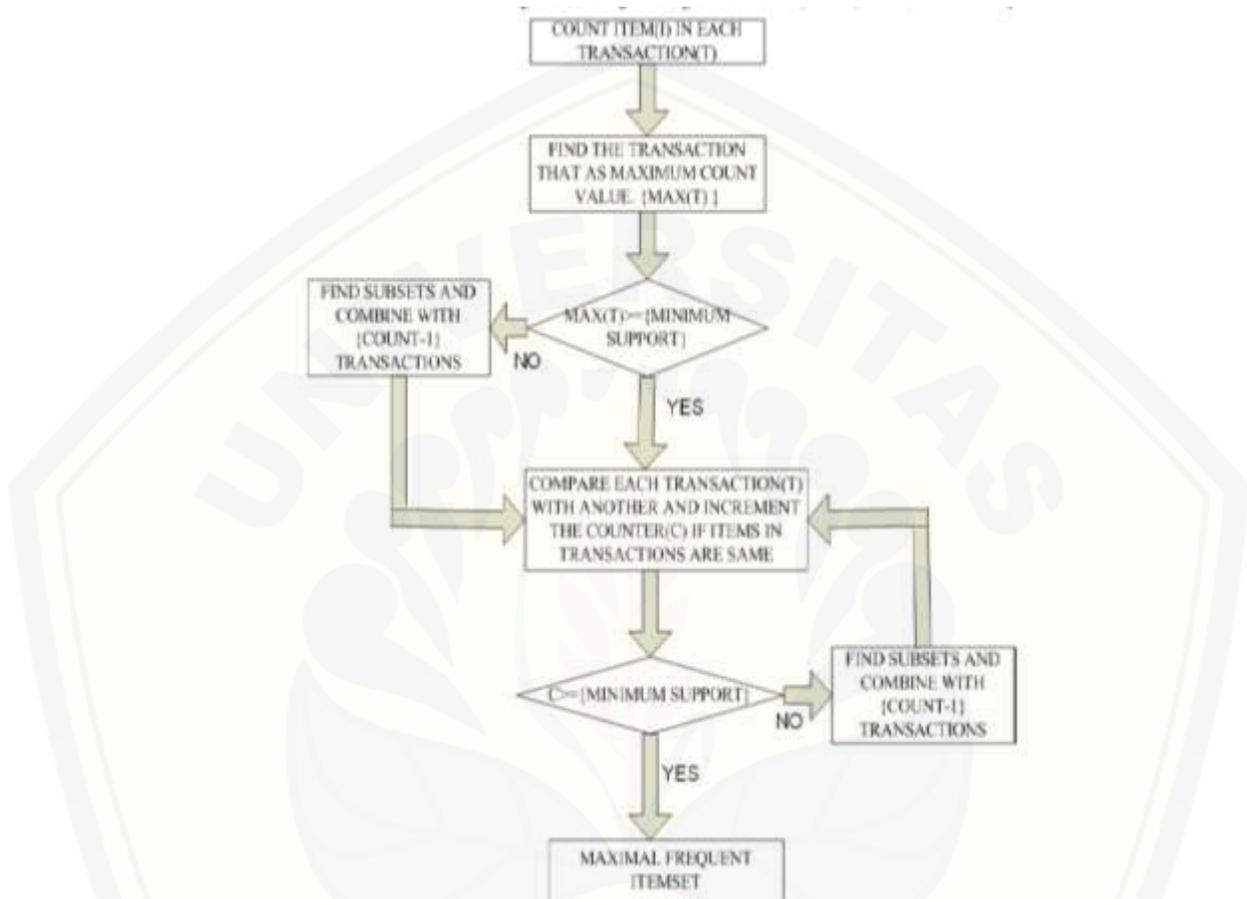
Gambar 2.2 Hasil pengujian algoritma pada *Association Rules* (Ruldeviyani & Fahrian, 2008)

Pada gambar 2.2 dapat dilihat bahwa perbedaan antar algoritma hanya terletak pada perbedaan waktu pada proses perhitungan dan pencarian sedangkan pada nilai support, algoritma – algoritma pada *Association Rules* ini memiliki nilai yang sama.

## 2.5. Frequent itemset

Sebuah item set yang mensupport lebih besar dari minimum support dasar threshold yang diketahui sebagai frequent itemset. (Jnanamurthy, Vishesh, Jain, Kumar, & M.Pai, 2013). Dengan adanya frequent itemset pada algoritma apriori ini, dapat menentukan batasan yang digunakan sebagai rule yang nantinya akan dibuat sebagai rekomendasi berdasarkan nilai minimum rule yang telah ditentukan oleh hasil perhitungan algoritma apriori dengan frequent itemset ini, sehingga sistem dapat memberikan hasil rekomendasi yang lebih tepat (berdasarkan data – data yang semakin banyak dan akurat, maka frequent itemset dapat menjadi lebih baik dalam perhitungannya).

Berikut bagan untuk mencari frequent itemsets yang nantinya akan berguna sebagai proses rekomendasi suatu sistem.



Gambar 2.3 Proses Pencarian Maximum Frequent Itemsets

(Jnanamurthy, Vishesh, Jain, Kumar, & M.Pai, 2013)

Pada Gambar 2.3, terdapat proses pencarian maximum sebuah frequent itemsets dimana proses dimulai dengan:

1. Menghitung item yang ada pada setiap transaksi.
2. Mencari transaksi yang memiliki value perhitungan yang maksimum.
3. Jika nilai maksimum lebih besar atau sama dengan minimum support, bandingkan setiap transaksi dengan yang lain dan kenaikan konter jika item

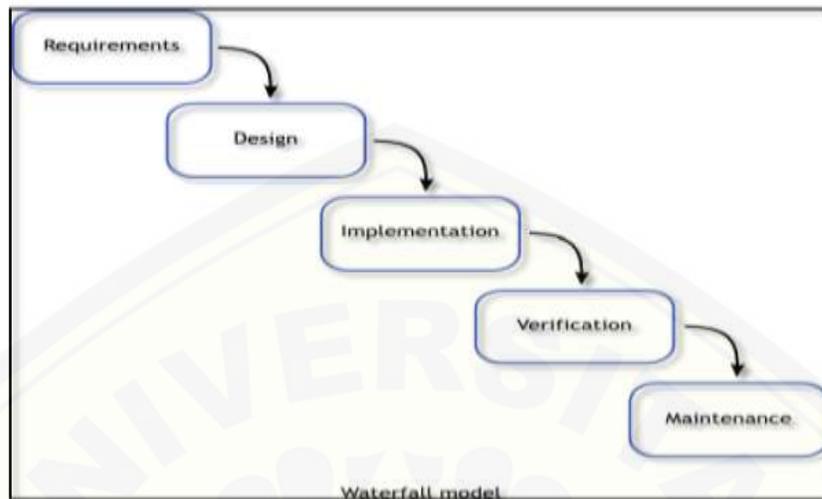
dalam transaksi sama, dan jika lebih kecil cari subsets dan gabungkan dengan transaksi -1 terlebih dahulu.

4. Hasil bandingan tersebut jika lebih kecil dari minimum support, cari subsets dan gabungkan dengan jumlah transaksi -1, lalu kembali pada proses banding, dan jika sudah lebih besar dari minimum support, nilai tersebut merupakan maximal frequent itemsets.

## 2.6. Metode Pengembangan Sistem

Pembuatan sistem informasi pada penelitian ini menggunakan model *waterfall*. Menurut (Verma, 2014) di dalam model *waterfall*, setiap fase harus terselesaikan sebelum melanjutkan ke fase selanjutnya, model ini memiliki 5 fase:

1. *Requirement analysis and Specification Phase*, hasil yang dikeluarkan dari fase ini adalah mengerti apa yang dibutuhkan oleh customer dan dibuat dokumentasi.
2. *Design Phase*, keluaran pada fase ini adalah mengubah spesifikasi kebutuhan awal menjadi struktur yang bisa digunakan untuk implementasi dalam bahasa program.
3. *Implementation and unit testing Design*, keluaran merupakan modul kecil yang telah diuji coba untuk software product.
4. *Integration and system testing phase*, ini adalah fase paling penting, sangat mahal dan mengkonsumsi banyak biaya pada project development.
5. *Operation and maintenance*, maintenance software yang bergelut di 3 aktivitas, yaitu error correction, enchancement of capabilities, dan deletion of capabilities and optimization.



Gambar 2.4 Model Waterfall

(Verma, 2014)

### 2.7. Analisis Kebutuhan

Pada tahap ini penulis menganalisa kebutuhan yang akan digunakan dalam pembuatan sistem. Kebutuhan tersebut dibagi menjadi dua bagian, yaitu kebutuhan fungsional dan kebutuhan non-fungsional dari sistem yang akan dibuat.

### 2.8. Perancangan Sistem

Pembuatan desain sistem pada penelitian ini menggunakan *Unified Modeling Language (UML)* yang dirancang dengan konsep *Object-Oriented Programming (OOP)*. Pemodelan UML yang digunakan adalah sebagai berikut:

#### 1. *Business Process*

*Business Process* merupakan model atau diagram yang menggambarkan sebuah proses lengkap dengan *resources* dan *information* yang dibutuhkan, event yang mendorong terjadinya proses dan goal yang dituju.

#### 2. *Use Case Diagram*

*Use case* merupakan model yang menggambarkan fungsi atau tugas yang dilakukan oleh *user*, baik manusia maupun mesin / komputer. *Use Case*

model ini dapat digunakan untuk menggambarkan *job specification* dan *job description*, serta keterkaitan antar *job*.

### 3. *Scenario*

*Scenario* diagram digunakan untuk menjelaskan atau menceritakan fitur atau isi yang ada di *use case* diagram. *Scenario* menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu event tertentu.

### 4. *Activity Diagram*

*Activity* diagram digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi. *Activity* diagram mempunyai fungsi yang sama dengan *scenario* namun diimplementasikan dalam diagram alir.

### 5. *Sequence Diagram*

*Sequence* diagram digunakan untuk menunjukkan rangkaian pesan yang dikirim antar object juga interaksi antar object.

### 6. *Class Diagram*

*Class* diagram digunakan untuk menggambarkan struktur statis class dalam sistem. *Class Diagram* dibuat untuk memudahkan dalam proses pengkodean.

## 2.9. Implementasi

Tahap ini mengimplementasikan desain yang telah dibuat menjadi sebuah aplikasi berbasis web. Beberapa hal yang dilakukan dalam tahap implementasi adalah menulis code program (*coding*) menggunakan bahasa pemrograman *Java* dan *C++*. Menggunakan database untuk menyimpan data – data yang ada dari studi kasus, dan menggunakan metode algoritma apriori dalam implementasinya.

## 2.10. Pengujian

Tahap pengujian yaitu uji coba terhadap sistem yang dibuat, tujuannya untuk menemukan kesalahan – kesalahan yang mungkin terjadi serta melakukan

*maintenance* (perbaikan) sehingga sistem dapat berjalan secara efisien dan normal. Pengujian dilakukan dengan dua metode, *Whitebox* dan *Blackbox*.

## 2.10.1. Pengujian White Box

Pengujian *White Box* dilakukan oleh tim penguji dari *developer* untuk mengecek masukan dan keluaran (I/O) serta kesesuaian algoritma yang digunakan dengan rancangan sistem yang telah dibuat. Pada pengujian *White Box* juga akan diterapkan pengujian teknik pengujian berbasis alur (*basis path testing*) dimana kompleksitas dari aplikasi yang dibangun akan dihitung menggunakan *Cyclomatic Complexity*. (Ayuliana, 2009). Dengan adanya pengujian white box ini, dapat diketahui apakah input dan output sudah sesuai dengan algoritma apriori sehingga aplikasi ini nantinya akan menjadi lebih kompleksitas dengan adanya pengujian ini.

Menurut Presman (2012) pengujian *white box* merupakan teknik pengujian jalur dasar yang digunakan untuk menentukan kompleksitas logis dengan menentukan rangkaian dasar jalur eksekusinya. Tahapan teknik pengujian jalur dasar meliputi:

### 1. Listing Program

Merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode – kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program.

### 2. Grafik Alir

Menurut Pressman (2012) Grafik alir merupakan Sebuah notasi sederhana yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari listing program. Grafik alir digambarkan dengan *node-node* (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program.

### 3. Kompleksitas Siklomatik

Kompleksitas Siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program (Pressman, 2012). Bila digunakan dalam konteks teknik pengujian jalur

dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jumlah jalur independen dalam basis ser suatu program (Pressman, 2012).

Rumus yang digunakan untuk menghitung kompleksitas siklomatika ditunjukkan pada persamaan berikut:

$$V(G) = E - N + 2 \dots \dots \dots \text{Persamaan (III)}$$

Keterangan:

V(G) : Kompleksitas Siklomatik

E : Jumlah Edge

N : Jumlah Node

## 2.10.2. Pengujian Black Box

Pengujian *Black Box* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Pengetahuan khusus dari kode aplikasi atau struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Metode ini memfokuskan pada keperluan fungsionalitas dari *software* (Wildan Agissa, 2013). Sangat penting pengujian black box pada aplikasi sistem informasi perpustakaan ini dikarenakan harus diperiksa apakah rekomendasi tersebut berfungsi dengan baik dan apakah berkaitan dengan struktur dari perpustakaan itu sendiri, sehingga fungsi aplikasi dapat benar – benar digunakan. Pengujian *blackbox* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan interface.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

## 2.11. Penelitian Terdahulu

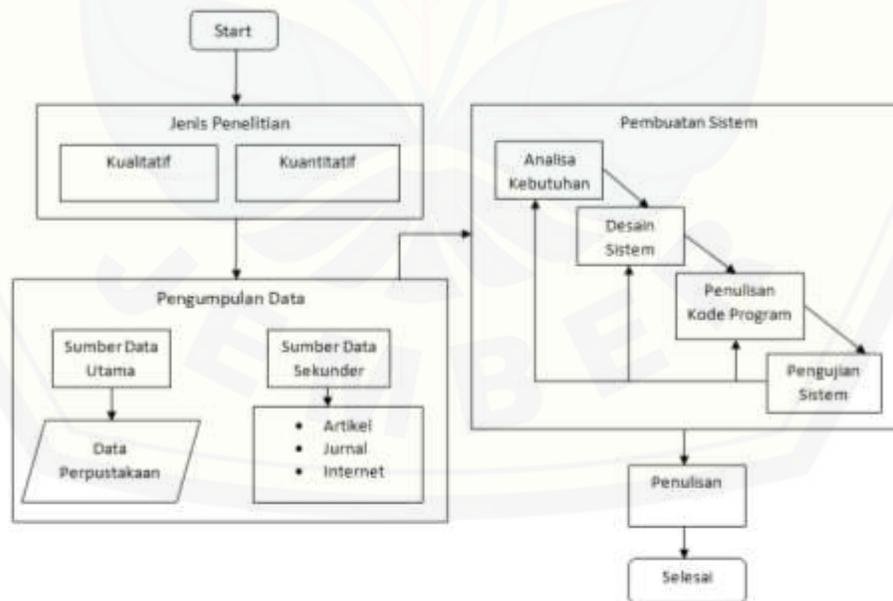
Penelitian yang berjudul “Implementasi Data Mining pada Penjualan Produk Elektronik dengan Algoritma Apriori (Studi Kasus : Kreditplus)” dilakukan oleh Dewi Kartika Pane yang merupakan mahasiswa Program Studi Teknik Informatika, STMIK Budidarma Medan. Penelitian ini mengimplementasikan metode Algoritma Apriori untuk menganalisa dan membuat aplikasi rekomendasi penjualan produk elektronik. Implementasi datamining pada penjualan produk elektronik ini menganalisa seberapa jauh algoritma apriori dapat membantu pengembangan strategi pemasaran, dalam studi kasus membantu perusahaan untuk mengetahui produk elektronik yang paling banyak terjual.

### BAB 3 METODOLOGI PENELITIAN

Pada bagian ini dijelaskan mengenai metodologi penelitian yang digunakan untuk menganalisa sekumpulan peraturan, kegiatan, dan prosedur yang digunakan oleh peneliti pada suatu project. Metode penelitian merupakan langkah dan prosedur yang akan dilakukan dalam mengumpulkan data atau informasi empiris guna memecahkan permasalahan, dan mengumpulkan informasi yang diperlukan untuk menyusun penelitian ini. Alur penelitian awal untuk membuat aplikasi sistem rekomendasi buku berbasis web terjabar menjadi beberapa sub.

#### 3.1. Alur Penelitian

Alur penelitian merupakan urutan langkah penelitian yang dilakukan mulai dari studi literatur, pengumpulan data, pengolahan data, perancangan sistem, dan pengimplementasian rancangan sistem. Diagram alur pada penelitian ini dapat dilihat pada Gambar 3.1 berikut:



Gambar 3.1 Flowchart Alur Penelitian

## 3.2. Jenis Penelitian

Pada penelitian ini menggunakan dua jenis penelitian, yaitu berupa penelitian kualitatif dan penelitian kuantitatif dimana pada kualitatif menganalisa studi literatur dan melakukan wawancara guna mengumpulkan informasi data yang diperlukan serta pada kuantitatif yang digunakan untuk mengkaji teori yang sudah ada sebelumnya.

## 3.3. Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan serta untuk memenuhi kebutuhan penelitian. Data yang digunakan dalam penelitian ini berdasarkan sumber datanya meliputi data utama atau data yang diperoleh dari studi kasus dan data sekunder berupa informasi lain yang membantu dalam penelitian:

### 1. Sumber Data Utama

Sumber data primer diperoleh langsung dari perpustakaan Universitas Jember yang menjadi tempat penelitian. Adapun data yang diperoleh dari sumber data meliputi data buku beserta ciri – ciri buku dan lokasi buku pada perpustakaan Universitas Jember, dan data transaksi yang berfungsi sebagai asal informasi mengenai rekomendasi buku tersebut muncul.

### 2. Sumber Data Sekunder

Dalam penelitian ini penulis memperoleh sumber data sekunder untuk menunjang penelitian seperti literatur yang diperoleh dari buku teks, jurnal yang diperoleh dari internet, dan penelitian sebelumnya.

Pengumpulan data tersebut nantinya akan diproses menjadi sebuah informasi rekomendasi dimana data informasi tersebut diperoleh dari seringnya relasi antar buku dalam transaksi pada perpustakaan.

### 3.3.1. Tempat Penelitian

Penelitian dilakukan di Perpustakaan Universitas Negeri Jember di Kabupaten Jember serta fakultas Sistem Informasi, penulis mengambil data mengenai informasi perpustakaan serta mata kuliah yang ada di fakultas tersebut.

### 3.3.2. Alat Penelitian

Alat yang digunakan untuk penelitian skripsi ini berupa hardware, yaitu sebuah laptop atau komputer dimana di dalamnya terdapat software diantaranya:

1. Windows 7
2. Notepad+++
3. DBMS MySQL
4. XAMPP Control Panel
5. Adobe Photoshop
6. Browser
7. Microsoft Office
8. Visual Paradigm

### 3.4. Pembuatan Sistem

Proses pembuatan aplikasi ini menggunakan model *waterfall* yang dikarenakan sistem klasifikasi ini termasuk sistem dengan skala kecil, yaitu dalam lingkup Universitas Jember saja, selain itu model *waterfall* juga dapat mempermudah dalam pengerjaan atau pembuatan sistem dalam merancang dalam kurun waktu yang lebih singkat. Dengan adanya kebutuhan sistem yang pasti dan membutuhkan waktu yang jelas, maka pemilihan *waterfall* ini menjadi pilihan yang tepat dalam penggunaannya. Pembuatan sistem diawali dengan menganalisa studi kasus guna mendapatkan informasi – informasi mengenai peminjaman dan pengembalian dalam perpustakaan Universitas Jember, setelah itu melanjutkan ke tahap perancangan desain sistem menggunakan aplikasi UML. Setelah desain sistem selesai, dilanjutkan dengan penulisan kode atau bisa disebut sebagai *coding system* yang didasari oleh desain sistem sebelumnya.

#### 3.5.1. Analisa Kebutuhan

Analisa kebutuhan merupakan langkah awal dari tahapan pengembangan perangkat lunak model *waterfall*. Proses yang dilakukan dalam tahap ini adalah mengumpulkan semua data yang dibutuhkan untuk membuat atau mengembangkan

aplikasi. Data tersebut nantinya akan digunakan untuk menentukan kebutuhan fungsional dan nonfungsional dari aplikasi yang akan dibuat. Data-data yang dibutuhkan adalah data buku yang nantinya akan digunakan sebagai sistem peminjaman dan pengembalian, serta beberapa sampel data transaksi sebagai contoh transaksi selama seminggu terakhir.

Peneliti telah melakukan tahap analisis kebutuhan dengan menggunakan metode observasi langsung dan wawancara kepada pihak perpustakaan di Universitas Jember. Hasil dari observasi tersebut adalah informasi mengenai rekomendasi buku untuk saat ini hanyalah menggunakan buku terkait berdasarkan tema ataupun judul. Dengan menggunakan algoritma apriori diharapkan bisa memberikan informasi lebih mengenai rekomendasi buku berdasarkan transaksi buku yang terjadi pada perpustakaan.

Tahap analisa data dimulai dengan menganalisa data – data yang didapat atau yang dikumpulkan. Data yang telah didapat tersebut dilakukan perhitungan menggunakan algoritma apriori. Pada studi kasus penelitian ini, data yang digunakan adalah data buku yang telah melakukan transaksi baik pada peminjaman maupun dari pengembalian.



Gambar 3.2 Flowchart penerapan algoritma apriori

Pada data buku itu sendiri nantinya akan memberikan informasi detail buku sehingga mahasiswa dapat mengetahui data lengkap mengenai buku beserta lokasi

rak buku tersebut berada. Pada data transaksi yang diperoleh, akan dilakukan perhitungan algoritma apriori yaitu dengan menghitung nilai dari *support* dan *confidence* sehingga mendapatkan hasil rekomendasi. Untuk penerapan data yang diproses menggunakan algoritma apriori digambarkan dengan *flowchart* yang dapat dilihat pada gambar 3.2 di atas.

### 3.5.2. Desain Sistem

Tahap selanjutnya setelah menganalisa kebutuhan yang ada pada perpustakaan adalah tahap desain sistem. Tahap ini akan mengimplementasikan data yang didapat kedalam sistem dalam bentuk diagram. Pada desain sistem ini menggunakan bahasa *Unified Modeling Language (UML)* dan menggunakan konsep *Object Oriented Programming (OOP)*. Model desain UML yang digunakan sebagai berikut :

a. Usecase Diagram

Use case diagram adalah gambaran fitur dari sistem yang dijalankan oleh aktor. Pada diagram ini dapat dilihat juga hak akses dari aktor.

b. Usecase Skenario

Usecase skenario adalah deskripsi tentang alur sistem dan user berdasarkan fitur yang berada didalam usecase diagram.

c. Activity Diagram

Activity diagram adalah gambaran tentang aktivitas sistem dan user dalam penggunaan aplikasi dari awal hingga akhir fitur. Menggambarkan implentasi algoritma dalam aplikasi.

d. Squence Diagram

Squance diagram adalah diagram yang menggambarkan interaksi antara objek satu dengan yang lain di dalam sistem yang dibangun pada urutan waktu. Diagram juga menggambarkan interaksi antara aktor, fitur, serta data yang berjalan. Squence diagram menggambarkan aliran logika dalam sebuah sistem yang dimodelkan secara visual dalam bentuk diagram.

e. Class Diagram

Class Diagram merupakan model statis yang menggambarkan struktur dan deskripsi class serta hubungannya antara class. Class Diagram terdiri dari nama class, atribut dan operasi atau method.

f. Entity Relation Diagram

Entity relation diagram (ERD) adalah diagram yang menggambarkan relasi objek-objek dasar data dalam sebuah basis data.

### 3.5.3. Penulisan Kode Program

Implementasi sistem merupakan tahap untuk mengimplementasikan atau mengubah desain sistem yang telah dibuat pada UML ke dalam kode program. Sistem dibuat berdasar desain yang dirancang. Bahasa pemrograman pada penelitian ini menggunakan PHP dengan tool yang digunakan yaitu Notepad++. Proses pengkodean menggunakan cara koding php manual dengan menggunakan desain MVC.

### 3.5.4. Pengujian Sistem

Tahap testing sistem harus dilakukan sebelum sistem dapat digunakan oleh user. Tahap dilakukan agar programmer dapat mengetahui apakah sistem yang dibangun sesuai dengan kebutuhan – kebutuhan yang didapatkan pada saat analisis kebutuhan. Pada tahap testing ini dilakukan guna menyempurnakan sistem sebelum sistem siap beroperasi dimana harus melewati tahap pengujian *white-box* dan *black-box*.

## BAB 4 ANALISIS DAN PENGEMBANGAN SISTEM

Bab ini akan menguraikan tentang proses analisis dan pengembangan sistem untuk implementasi Sistem Pengklasifikasian Rekomendasi Buku menggunakan *Algoritma Apriori*. Proses analisis dan pengembangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, dilanjutkan dengan pembuatan *usecase diagram*, skenario, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relation diagram* (ERD).

### 4.1. Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak dalam penelitian ini yaitu dengan cara mengidentifikasi permasalahan yang ada untuk kemudian dicatat dan dijadikan suatu bahan untuk memulai pembangunan aplikasi Sistem Informasi Perpustakaan dengan rekomendasi buku pada Universitas Jember. Analisis kebutuhan yang dilakukan meliputi proses pengumpulan data kebutuhan fungsional dan kebutuhan non-fungsional.

#### 4.1.1. Kebutuhan Fungsional

Kebutuhan fungsional sistem pada penelitian ini adalah sebagai berikut:

1. Akses sistem terdiri dari admin, pegawai (user), dan umum.
2. Sistem dapat melakukan input update delete data user untuk pengguna aplikasi ini dengan menggunakan akses admin.
3. Sistem dapat melakukan input update delete data mahasiswa untuk digunakan sebagai data peminjam dengan menggunakan akses admin.
4. Sistem dapat melakukan input update delete data buku yang ada pada perpustakaan Universitas Jember dengan menggunakan akses admin.
5. Pada hak akses pegawai atau user, sistem dapat melakukan pencatatan peminjaman dan pengembalian buku oleh mahasiswa.

6. Pada hak akses pegawai atau user, sistem dapat melakukan pencarian buku untuk nantinya digunakan sebagai pencarian barcode.
7. Pada hak akses pegawai atau user, sistem dapat mencari rekomendasi buku yang terkait dari buku yang dipinjam menggunakan *Algoritma Apriori*.
8. Pada hak akses umum, sistem dapat melakukan pencarian buku.

#### 4.1.2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional sistem pada penelitian ini adalah:

1. Tampilan sistem yang user friendly.
2. Sistem berbasis browser.
3. Sistem memiliki hak akses.

## 4.2. Desain Sistem

Desain pengembangan sistem yang akan dibuat menggunakan *Unified Modeling Language* (UML) yang dirancang menggunakan konsep *Object-Oriented Programming* (OOP). Pemodelan UML yang digunakan antara lain :

1. *Business Process*
2. *Use Case Diagram*
3. *Use Case Scenario*
4. *Sequence Diagram*
5. *Activity Diagram*
6. *Class Diagram*
7. *Entity Relationship Diagram* (ERD).

#### 4.2.1. Business Process

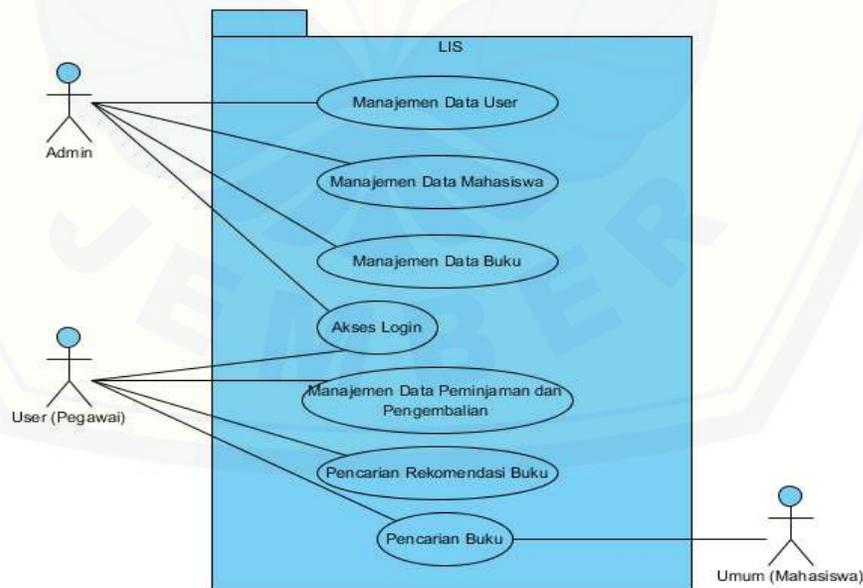
Merupakan serangkaian atau sekumpulan aktifitas dalam sebuah sistem yang dirancang untuk mencapai tujuan sebuah sistem. Pada *business process* terdapat beberapa komponen untuk mencapai tujuan tersebut, yaitu masukan (*input*), media yang digunakan (*uses*) yang kemudian akan menghasilkan tujuan (*goal*). *Business process* sistem informasi perpustakaan ini dapat dilihat pada Gambar 4.1.



Gambar 4.1 *Business Process* Sistem Informasi Perpustakaan

4.2.2. Usecase Diagram

Usecase diagram digunakan untuk menggambarkan fitur yang dibuat dalam sistem dan juga menggambarkan siapa saja yang dapat menggunakan fitur pada sistem informasi perpustakaan tersebut. Usecase diagram sistem informasi perpustakaan ditunjukkan pada Gambar 4.1:



Gambar 4.2 Usecase Sistem Informasi Perpustakaan

Pada usecase diagram tersebut terdapat tiga klasifikasi aktor yang dapat menggunakan sistem informasi perpustakaan. Tiga aktor tersebut diantaranya adalah admin, user (pegawai), dan umum (mahasiswa). Deskripsi aktor pada usecase dijelaskan pada tabel 4.1 dan tabel 4.2 di bawah ini:

Tabel 4.1 Definisi Aktor Usecase

| No | Aktor | Definisi Tugas   |
|----|-------|--|
| 1. | Admin | Manajemen data user, data mahasiswa, data buku yang ada yang meliputi input, update, delete data.  |
| 2. | User  | Manajemen data peminjaman dan pengembalian buku oleh mahasiswa, dapat melakukan pencarian buku serta dapat mencari rekomendasi buku yang dipinjam mahasiswa berdasarkan keterkaitan. |
| 3. | Umum  | Dapat melakukan pencarian buku pada perpustakaan Universitas Jember  |

Tabel 4.2 Definisi Usecase

| No | Usecase                                    | Deskripsi   |
|----|--|---|
| 1. | Manajemen data user                        | Menambah, mengubah dan menghapus data pengguna user pada sistem.  |
| 2. | Manajemen data mahasiswa                   | Melakukan input, update, delete data mahasiswa yang aktif dalam peminjaman buku dalam sistem.                           |
| 3. | Manajemen data buku                        | Melakukan input, update, delete data buku yang ada pada perpustakaan Universitas Jember.                                |
| 4. | Manajemen data peminjaman dan pengembalian | Melakukan pencatatan transaksi peminjaman dan pengembalian buku dalam sistem informasi perpustakaan ini oleh mahasiswa. |
| 5. | Pencarian rekomendasi buku                 | Melakukan pencarian rekomendasi buku yang terkait menggunakan algoritma apriori.  |

|    |                |  |
|----|----------------|--|
| 6. | Pencarian buku | Melakukan pencarian buku pada database buku perpustakaan dalam sistem. |
|----|----------------|--|

### 4.2.3. Skenario

Menjelaskan alur cerita (skenario) pada setiap usecase yang telah digambarkan dalam usecase diagram secara lebih detail. Usecase scenario berisi id, nama usecase, aktor, pre condition, post condition, skenario normal, dan skenario alternatif.

#### a. Skenario Login Admin

Penjelasan urutan reaksi aktor (admin) dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario login admin dapat dilihat pada tabel 4.3.

Tabel 4.3 Skenario login Admin

| Nama                | Manajemen Data User  |
|---------------------|--|
| Participating Actor | Admin.   |
| Entry Condition     | Admin melakukan login ke aplikasi LIS.   |
| Exit Condition      | Admin berhasil masuk ke dalam sistem.  |
| Event Flow          | <ol style="list-style-type: none"> <li>1. Admin membuka aplikasi LIS</li> <li>2. Admin melakukan login ke dalam LIS</li> </ol> |

| Skenario Utama<br>Login Admin      |                                   |
|------------------------------------|-----------------------------------|
| Admin                              | System                            |
| 1. Membuka aplikasi LIS basis web. |                                   |
|                                    | 2. Menampilkan tampilan awal LIS. |
| 3. Klik menu login                 |                                   |

|   |  |
|---|--|
|   | 4. Menampilkan menu login  |
| 5. Melakukan login dengan mengisi id Username dan Password. |  |
| 6. Klik button “Click here to login”.                       |  |
|   | 7. Sistem mengecek ke database user, apakah id user benar dan apakah termasuk Admin atau Pegawai.  |
|   | 8. Sistem menampilkan homepage LIS untuk Admin dengan beberapa menu utama yaitu: <ul style="list-style-type: none"> <li>• Halaman Utama</li> <li>• Tentang Aplikasi</li> <li>• User LIS</li> <li>• Data Mahasiswa</li> <li>• Data Buku.</li> </ul> |

| Skenario Alternatif   |   |
|---|---|
| Username atau Password Salah  |   |
| Admin   | System  |
| 5. Melakukan login dengan mengisi id Username dan Password lalu mengklik “Click here to login”. |   |
|   | 6. System mengecek ke database user, apakah id user benar dan |

|   |   |
|---|---|
|   | apakah termasuk Admin atau Pegawai.   |
|   | 7. System menampilkan warning message “Username tidak terdaftar atau salah input username dan password!” dan button OK. |
| 8. Klik button OK.  |   |
|   | 9. Kembali ke tampilan awal LIS.  |
| Skenario Alternatif<br>Tidak terkoneksi dengan database   |   |
| Admin   | System  |
| 5. Melakukan login dengan mengisi id Username dan Password lalu mengklik “Click here to login”. |   |
|   | 6. Sistem mengecek koneksi ke database.   |
|   | 7. Sistem menampilkan warning message “tidak ada koneksi ke basis data”   |

b. Manajemen Data User

Penjelasan urutan reaksi admin dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen Data User dapat dilihat pada tabel 4.4.

Tabel 4.4 Skenario Manajemen Data User

| Nama                | Manajemen Data User   |
|---------------------|---|
| Participating Actor | Admin.  |
| Entry Condition     | Admin telah login ke dalam LIS untuk mengelola data User LIS.   |
| Exit Condition      | Data User telah di insert, atau update oleh admin LIS.  |
| Event Flow          | <ol style="list-style-type: none"> <li>3. Admin membuka aplikasi LIS</li> <li>4. Admin melakukan login ke dalam LIS</li> <li>5. Admin memilih menu User LIS.</li> </ol> |

| Skenario Utama<br>Manajemen User LIS  |   |
|---|---|
| Admin   | System                                    |
| 1. Memilih menu User LIS.   |   |
|   | 2. Sistem menampilkan interface User LIS. |
| Skenario Utama<br>Insert User LIS   |   |
| Admin   | System                                    |
| 1. Memilih button “Tambah User”.  |   |
|   | 2. Menampilkan form tambah User.          |
| 3. Mengisi form tambah User, diantaranya: <ul style="list-style-type: none"> <li>• Username</li> <li>• Password User</li> </ul> |   |

|  |  |
|--|--|
| <ul style="list-style-type: none"> <li>• Nama User</li> <li>• Hak Akses User.</li> </ul> |  |
| 4. Klik button “Input Data User”.  |  |
|  | 5. Menyimpan user LIS yang baru ke dalam database user.      |
| Skenario Alternatif  |  |
| Pengisian data User LIS belum lengkap  |  |
| Admin  | System   |
| 4. Klik button “Input Data User”.  |  |
|  | 5. Menampilkan warning message “Please fill out this field”. |
| Skenario Utama   |  |
| Update User LIS  |  |
| Admin  | System   |
| 1. Memilih button “Edit User”.   |  |
|  | 2. Menampilkan data User LIS pada database ke interface      |
| 3. Memilih user yang akan di update datanya, lalu klik button “Edit”                     |  |
|  | 4. Menampilkan form Edit User.                               |
| 5. Mengedit user yang ada pada form dengan sesuai indikasi.                              |  |
| 6. Klik button “Update Data User”.   |  |
|  | 7. Menyimpan User LIS yang telah ter edit.                   |

| Skenario Alternatif   |  |
|---|--|
| Pengisian data User LIS saat Edit belum lengkap             |  |
| Admin   | System   |
| 4. Memilih button “Update Data User”.                       |  |
|   | 5. Menampilkan warning message “please fill out this field”. |
| Skenario Utama  |  |
| Hapus User LIS  |  |
| Admin   | System   |
| 1. Memilih button “Hapus User” pada User yang akan dihapus. |  |
|   | 2. Menghapus data User LIS yang telah dipilih oleh Admin     |

c. Manajemen Data Mahasiswa

Penjelasan urutan reaksi admin dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen Data Mahasiswa dapat dilihat pada tabel 4.5.

Tabel 4.5 Skenario Manajemen Data Mahasiswa

| Nama                | Manajemen Data Mahasiswa                                      |
|---------------------|---|
| Participating Actor | Admin.  |
| Entry Condition     | Admin telah login ke dalam LIS untuk mengelola data User LIS. |
| Exit Condition      | Data Mahasiswa telah di insert, atau update oleh admin LIS.   |
| Event Flow          | 1. Admin membuka aplikasi LIS                                 |

|  |  |
|--|--|
|  | <ol style="list-style-type: none"> <li>2. Admin melakukan login ke dalam LIS</li> <li>3. Admin memilih menu User LIS.</li> </ol> |
|--|--|

| Skenario Utama<br>Manajemen User LIS   |   |
|--|---|
| Admin  | System  |
| 1. Memilih menu Data Mahasiswa.  |   |
|  | 2. Sistem menampilkan interface Data Mahasiswa.         |
| Skenario Utama<br>Insert User LIS  |   |
| Admin  | System  |
| 1. Memilih button “Tambah Mahasiswa”.  |   |
|  | 2. Menampilkan form tambah Mahasiswa.                   |
| 3. Mengisi form tambah User, diantaranya: <ul style="list-style-type: none"> <li>• Nama</li> <li>• Tempat Tinggal</li> <li>• Tempat, tanggal lahir</li> <li>• Status Kuliah</li> </ul> |   |
| 4. Klik button “Input Data Mahasiswa”.   |   |
|  | 5. Menyimpan user LIS yang baru ke dalam database user. |
| Skenario Alternatif  |   |

| Pengisian data User LIS belum lengkap   |   |
|---|---|
| Admin   | System  |
| 6. Klik button “Input Data Mahasiswa”.  |   |
|   | 7. Menampilkan warning message “Please fill out this field”.        |
| Skenario Utama<br>Update User LIS   |   |
| Admin   | System  |
| 8. Memilih button “Edit Mahasiswa”.   |   |
|   | 9. Menampilkan data mahasiswa LIS pada database ke interface        |
| 10. Memilih nama mahasiswa yang akan di update datanya, lalu klik button “Edit” |   |
|   | 11. Menampilkan form Edit Mahasiswa.                                |
| 12. Mengedit data mahasiswa yang ada pada form dengan sesuai indikasi.          |   |
| 13. Klik button “Update Data Mahasiswa”.  |   |
|   | 14. Menyimpan data mahasiswa yang telah ter edit ke dalam database. |
| Skenario Alternatif   |   |

| Pengisian data User LIS saat Edit belum lengkap                            |  |
|--|--|
| Admin  | System   |
| 6. Memilih button “Update Data Mahasiswa”.                                 |  |
|  | 7. Menampilkan warning message “please fill out this field”.             |
| Skenario Utama Hapus User LIS  |  |
| Admin  | System   |
| 3. Memilih button “Hapus Mahasiswa” pada data mahasiswa yang akan dihapus. |  |
|  | 4. Menghapus data mahasiswa yang telah dipilih oleh Admin pada database. |

d. Manajemen Data Buku

Penjelasan urutan reaksi aktor dan sistem pada skenario normal dan skenario alternatif usecase skenario Manajemen Data Buku LIS dapat dilihat pada tabel 4.6.

Tabel 4.6 Skenario Manajemen Data Buku

| Name                | Manajemen Data Buku   |
|---------------------|---|
| Participating Actor | Admin.  |
| Entry Condition     | Admin telah login ke dalam LIS untuk mengelola data User LIS.   |
| Exit Condition      | Data Buku telah di insert, atau update oleh admin.  |
| Event Flow          | <ol style="list-style-type: none"> <li>1. Admin membuka aplikasi LIS</li> <li>2. Admin melakukan login ke dalam LIS</li> <li>5. Admin memilih menu Data Buku</li> </ol> |

| Skenario Utama<br>Manajemen Data Buku  |  |
|--|--|
| Admin  | System                                 |
| 1. Memilih menu Data Buku.   |  |
|  | 2. Menampilkan interface Data Buku.    |
| Skenario Utama<br>Insert Data Buku   |  |
| Admin  | System                                 |
| 1. Memilih button “Tambah Data Buku”.  |  |
|  | 2. Menampilkan form Tambah Data Buku”. |
| 3. Mengisi form tambah data buku sesuai dengan kolom yaitu: <ul style="list-style-type: none"> <li>• Nomor Kelas</li> <li>• Nomor Barcode</li> <li>• Pengarang</li> <li>• Judul</li> <li>• Penerbit</li> <li>• Item Type</li> <li>• Lokasi</li> <li>• Harga</li> <li>• Tanggal Input.</li> </ul> |  |
| 4. Klik button “Input Data Buku”   |  |
|  | 5. Menyimpan data buku ke              |

|  |   |
|--|---|
|  | database.   |
| Skenario Alternatif  |   |
| Insert Data Buku belum lengkap   |   |
| Admin  | System  |
| 4. Klik button “Input Data Buku”.                                      |   |
|  | 5. Menampilkan warning message “please fill out this field”.        |
| Skenario Utama   |   |
| Update Data Buku   |   |
| Admin  | System  |
| 1. Memilih button “Edit Data Buku” pada menu Data Buku.                |   |
|  | 2. Menampilkan data buku yang ada pada database ke dalam interface. |
| 3. Memilih data buku yang akan di edit datanya, lalu klik button Edit. |   |
|  | 4. Menampilkan form edit data.                                      |
| 5. Edit data yang ada pada form edit data sesuai dengan kolom.         |   |
| 6. Klik button “Update Data Buku”                                      |   |
|  | 7. Menyimpan data yang telah ter update ke dalam database.          |
| Skenario Alternatif  |   |

| Entry Edit Data belum lengkap      |  |
|------------------------------------|--|
| Admin                              | System   |
| 4. Klik button “Update Data Buku”. |  |
|                                    | 5. Menampilkan warning message “please fill out this field”.     |
| Skenario Utama<br>Search Data Buku |  |
| Admin                              | System   |
| 1. Mengisi field Search.           |  |
| 2. Klik button “Search”.           |  |
|                                    | 3. Menampilkan data buku yang sesuai dengan text yang di search. |
| Skenario Utama<br>Hapus Data Buku  |  |
| Admin                              | System   |
| 1. Memilih data yang akan dihapus. |  |
| 2. Klik button “Hapus”.            |  |
|                                    | 3. Menghapus data yang dipilih oleh user pada database.          |

e. Skenario Login Pegawai

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase login pegawai dapat dilihat pada Tabel 4.7.

Tabel 4.7 Skenario login pegawai

|                     |  |
|---------------------|--|
| Nama                | Manajemen Data User  |
| Participating Actor | Pegawai.   |
| Entry Condition     | Pegawai melakukan login ke aplikasi LIS.   |
| Exit Condition      | Pegawai berhasil masuk ke dalam sistem.  |
| Event Flow          | <ol style="list-style-type: none"> <li>1. Pegawai membuka aplikasi LIS</li> <li>2. Pegawai melakukan login ke dalam LIS</li> </ol> |

| Skenario Utama<br>Login Pegawai                             |   |
|---|---|
| Pegawai   | System  |
| 1. Membuka aplikasi LIS basis web.                          |   |
|   | 2. Menampilkan tampilan awal LIS.   |
| 3. Klik menu login  |   |
|   | 4. Menampilkan menu login   |
| 5. Melakukan login dengan mengisi id Username dan Password. |   |
| 6. Klik button "Click here to login"                        |   |
|   | 7. System mengecek ke database user, apakah id user benar dan apakah termasuk Admin atau Pegawai. |
|   | 8. Sistem menampilkan homepage LIS untuk Pegawai  |

|  |   |
|--|---|
|  | <p>dengan beberapa menu utama yaitu:</p> <ul style="list-style-type: none"> <li>• Halaman Utama</li> <li>• Tentang Aplikasi</li> <li>• Pencarian Buku</li> <li>• Peminjaman dan Pengembalian Buku</li> <li>• Rekomendasi Buku.</li> </ul> |
| <p>Skenario Alternatif</p> <p>Username atau Password Salah</p>   |   |
| <p>Pegawai</p>   | <p>System</p>   |
| <p>3. Melakukan login dengan mengisi id Username dan Password lalu mengklik “Click here to login”.</p> |   |
|  | <p>4. System mengecek ke database user, apakah id user benar dan apakah termasuk Admin atau Pegawai.</p>  |
|  | <p>5. System menampilkan warning message “Username tidak terdaftar atau salah input username dan password!” dan button OK.</p>  |
| <p>6. Klik button OK.</p>  |   |
|  | <p>7. Kembali ke tampilan awal LIS.</p>   |
| <p>Skenario Alternatif</p>   |   |

| Tidak terkoneksi dengan database  |   |
|---|---|
| Pegawai   | System  |
| 3. Melakukan login dengan mengisi id Username dan Password lalu mengklik “Click here to login”. |   |
|   | 4. Sistem mengecek koneksi ke database.                                 |
|   | 5. Sistem menampilkan warning message “tidak ada koneksi ke basis data” |

f. Pencarian Buku

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal usecase skenario pada pencarian buku oleh user atau pegawai dapat dilihat pada tabel 4.8.

Tabel 4.8 Skenario Pencarian Buku

| Name                | Pencarian Buku   |
|---------------------|--|
| Participating Actor | Pegawai.   |
| Entry Condition     | Pegawai telah melakukan login ke dalam LIS sebagai user Pegawai.   |
| Exit Condition      | Sistem menampilkan hasil pencarian buku pada database data buku.   |
| Event Flow          | <ol style="list-style-type: none"> <li>1. Pegawai membuka aplikasi LIS</li> <li>2. Pegawai menginputkan username dan password untuk login ke dalam LIS</li> <li>3. Pegawai memilih menu Pencarian Buku.</li> </ol> |

Skenario Utama

| Pencarian Buku   |  |
|--|--|
| Pegawai  | System   |
| 1. Pegawai memilih menu Pencarian Buku.  |  |
|  | 2. Menampilkan interface menu pencarian buku dengan 3 field, yaitu: <ul style="list-style-type: none"> <li>• Barcode Buku</li> <li>• Judul Buku</li> <li>• Pengarang.</li> </ul> |
| 3. Pegawai menuliskan kriteria buku yang akan dicari pada field yang tersedia. |  |
| 4. Klik button “Cari Buku”   |  |
|  | 5. Menampilkan buku yang dicari pada database ke tab baru pada browser.  |

g. Manajemen Data Peminjaman dan Pengembalian Buku

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario manajemen peminjaman buku oleh user atau pegawai melalui LIS dapat dilihat pada Tabel 4.9.

Tabel 4.9 Skenario Manajemen Data Peminjaman dan Pengembalian

| Name                | Manajemen Data Peminjaman dan Pengembalian                       |
|---------------------|--|
| Participating Actor | Pegawai.   |
| Entry Condition     | Pegawai telah melakukan login ke dalam LIS sebagai user Pegawai. |
| Exit Condition      | Sistem dapat mencatat transaksi yang dilakukan oleh              |

|            |  |
|------------|--|
|            | customer ke dalam database berupa peminjaman dan pengembalian buku.  |
| Event Flow | <ol style="list-style-type: none"> <li>4. Pegawai membuka aplikasi LIS</li> <li>5. Pegawai menginputkan username dan password untuk login ke dalam LIS</li> <li>6. Pegawai memilih menu Peminjaman dan Pengembalian Buku.</li> </ol> |

| Skenario Utama<br>Manajemen Data Peminjaman dan Pengembalian |   |
|--|---|
| Pegawai  | System  |
| 1. Pegawai memilih menu Peminjaman dan Pengembalian Buku.    |   |
|  | 2. Menampilkan interface menu dengan 2 pilihan, yaitu: <ul style="list-style-type: none"> <li>• Peminjaman Buku</li> <li>• Pengembalian Buku</li> </ul> |
| Skenario Utama<br>Peminjaman Buku                            |   |
| Pegawai  | System  |
| 1. Klik button “Peminjaman Buku”.                            |   |
|  | 2. Menampilkan interface menu peminjaman buku.  |
| 3. Pilih NIM mahasiswa yang akan meminjam buku               |   |
| 4. Pilih id buku yang akan                                   |   |

|  |   |
|--|---|
| dipinjam menggunakan id yang didapat pada pencarian buku sebelumnya. |   |
| 5. Klik button “Pinjam Buku”   |   |
|  | 6. Menyimpan data peminjaman ke dalam database peminjaman.  |
| <b>Skenario Utama</b><br><b>Pengembalian Buku</b>                    |   |
| Pegawai  | System  |
| 1. Klik button “Pengembalian Buku”.                                  |   |
|  | 2. Menampilkan interface menu pengembalian buku beserta data peminjaman dan history pengembalian. |
| 3. Klik button “History Peminjaman”                                  |   |
| 4. Klik button “Kembalikan” pada buku yang akan dikembalikan         |   |
|  | 5. Sistem memunculkan javascript persetujuan untuk pengembalian buku.                             |
| 6. Klik OK   |   |
|  | 7. Menyimpan data yang telah diubah ke dalam database.  |
|  | 8. Menampilkan menu   |

|   |   |
|---|---|
|   | pengembalian.   |
| Skenario Alternatif<br>Melihat History Pengembalian |   |
| Pegawai   | System  |
| 1. Klik button “Pengembalian Buku”                  |   |
|   | 2. Menampilkan interface menu pengembalian buku beserta data peminjaman dan history pengembalian. |
| 3. Klik button “History Pengembalian”               |   |

#### h. Pencarian Rekomendasi Buku

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario pencarian rekomendasi buku oleh user atau pegawai melalui LIS dapat dilihat pada Tabel 4.10.

Tabel 4.10 Skenario Pencarian Rekomendasi Buku

|                     |  |
|---------------------|--|
| Name                | Pencarian Rekomendasi Buku   |
| Participating Actor | Pegawai.   |
| Entry Condition     | Pegawai telah melakukan login ke dalam LIS sebagai user Pegawai.   |
| Exit Condition      | Sistem dapat mencari hasil rekomendasi buku berdasarkan association rules yang berlaku.  |
| Event Flow          | <ol style="list-style-type: none"> <li>1. Pegawai membuka aplikasi LIS</li> <li>2. Pegawai menginputkan username dan password untuk login ke dalam LIS</li> <li>3. Pegawai memilih menu Rekomendasi Buku.</li> </ol> |

| Skenario Utama                                |   |
|---|---|
| Pencarian Rekomendasi Buku                    |   |
| Pegawai                                       | System  |
| 1. Pegawai memilih menu Rekomendasi Buku.     |   |
|   | 2. Menampilkan interface menu rekomendasi relasi antar buku dengan 3 field, yaitu: <ul style="list-style-type: none"> <li>• Minimum Confidence</li> <li>• Minimum Support</li> <li>• Periode Awal</li> <li>• Periode Akhir</li> </ul> |
| 3. Pegawai menginputkan data yang akan dicari |   |
| 4. Klik button “Cari Association Rule”        |   |
|   | 5. Sistem menampilkan hasil rekomendasi buku berdasarkan tanggal periode yang ditentukan dan minimum support dan confidence nya.  |
| Skenario Alternatif                           |   |
| Pengisian field rekomendasi belum lengkap     |   |
| Admin   | System  |
| 4. Klik button “Cari Association Rule”.       |   |
|   | 5. Menampilkan warning  |

|  |                                       |
|--|---------------------------------------|
|  | message “Please fill out this field”. |
|--|---------------------------------------|

i. Pencarian Buku oleh Mahasiswa

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario pencarian buku oleh mahasiswa melalui LIS dapat dilihat pada Tabel 4.11.

Tabel 4.11 Skenario Pencarian Buku oleh Mahasiswa

| Name                | Pencarian Rekomendasi Buku   |
|---------------------|--|
| Participating Actor | Umum (Mahasiswa).  |
| Entry Condition     | User telah membuka aplikasi LIS dan membuka menu Pencarian Buku.   |
| Exit Condition      | Sistem dapat mencari hasil pencarian buku serta rekomendasi buku berdasarkan association rules yang berlaku.                             |
| Event Flow          | <ol style="list-style-type: none"> <li>1. User membuka aplikasi LIS</li> <li>2. User menginputkan data buku yang akan dicari.</li> </ol> |

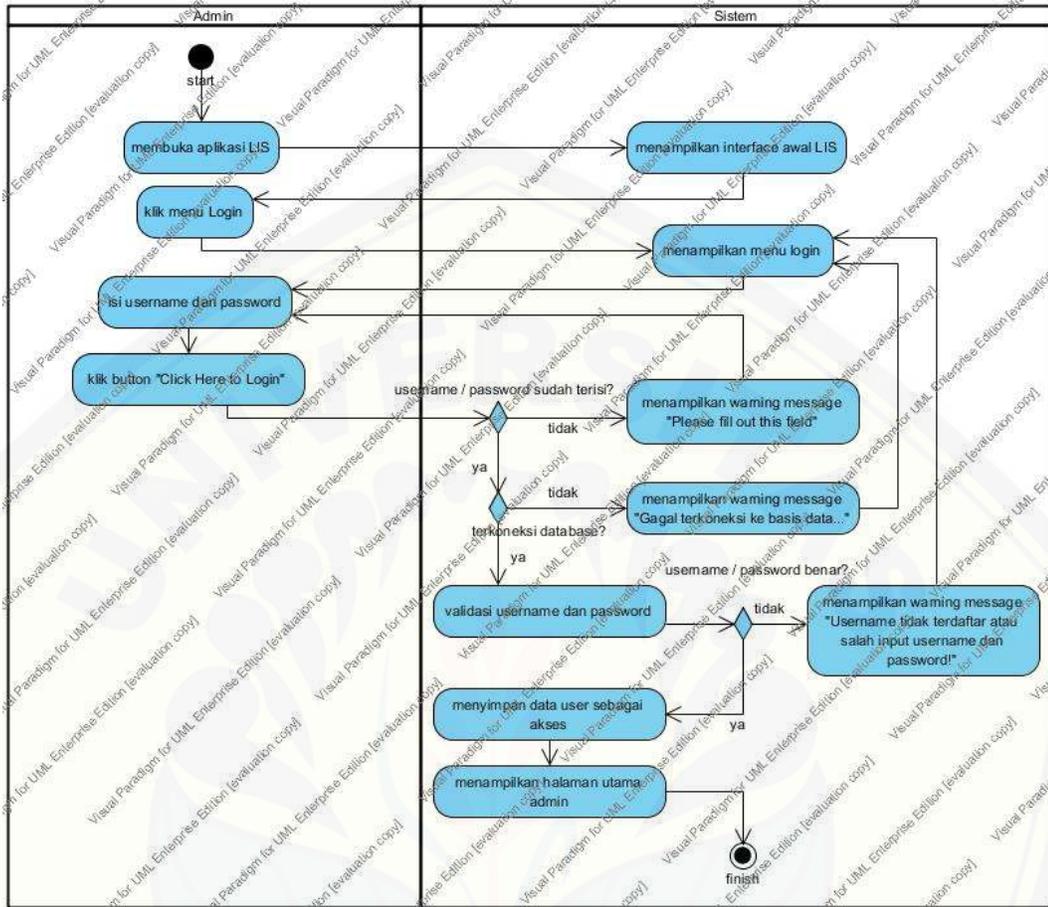
| Skenario Utama                       |   |
|--------------------------------------|---|
| Pencarian Rekomendasi Buku           |   |
| Mahasiswa                            | System  |
| 1. User memilih menu Pencarian Buku. |   |
|                                      | <ol style="list-style-type: none"> <li>2. Menampilkan interface menu pencarian buku dengan 2 opsi pencarian, yaitu: <ul style="list-style-type: none"> <li>• Cari buku berdasar ciri</li> </ul> </li> </ol> |

|  |   |
|--|---|
|  | <p>buku</p> <ul style="list-style-type: none"> <li>• Cari buku berdasar ID</li> </ul>           |
| 3. User menginputkan data yang akan dicari berdasar ciri buku. |   |
| 4. Klik button “Cari Buku”                                     |   |
|  | 5. Sistem menampilkan hasil pencarian buku pada database ke dalam tabel.                        |
| 6. Klik “Detail Buku” pada buku yang akan dicek detailnya.     |   |
|  | 7. Menampilkan detail buku beserta rekomendasi buku yang terkait berdasarkan algoritma apriori. |
| 8. Klik button “Selesai” untuk menutup tab.                    |   |
|  | 9. Aplikasi menutup tab hasil pencarian buku.   |
| <p>Skenario Alternatif<br/>Pencarian Buku Berdasar ID Buku</p> |   |
| Mahasiswa  | System  |
| 3. User menginputkan data yang akan dicari berdasar id buku.   |   |
| 4. Klik button “Cari Buku”                                     |   |
|  | 5. Sistem menampilkan hasil pencarian buku pada database ke dalam tabel.                        |
| 6. Klik “Detail Buku” pada buku                                |   |

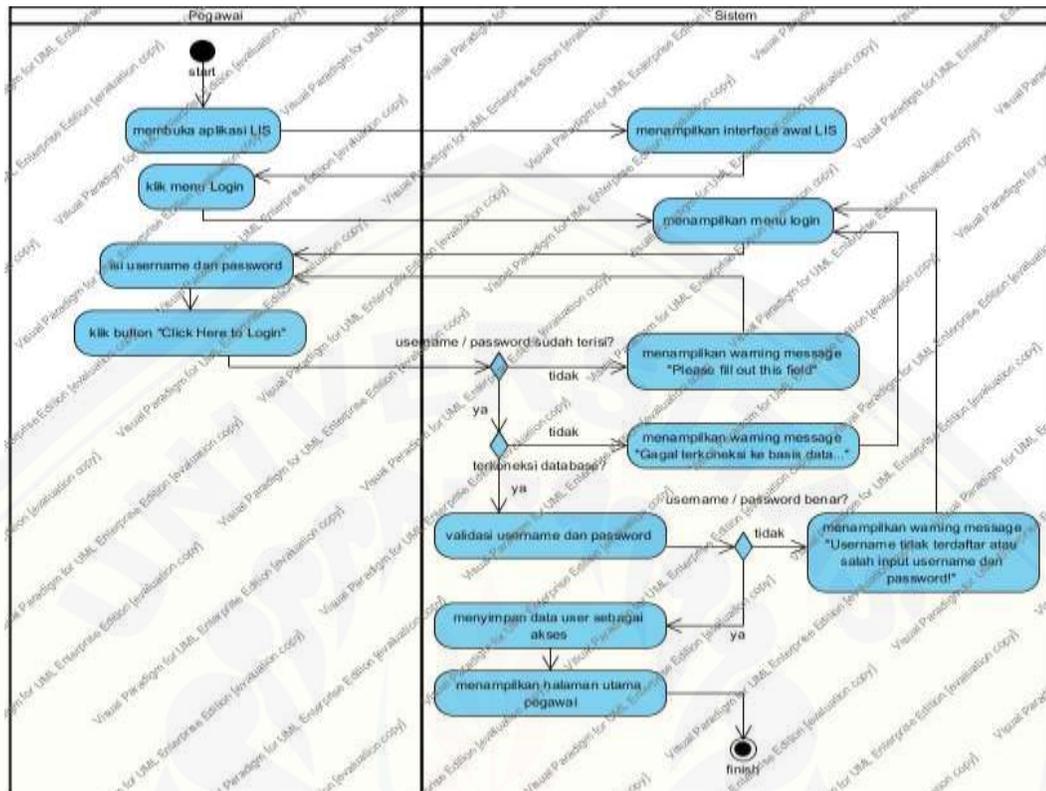
|  |   |
|--|---|
| yang akan dicek detailnya.                   |   |
|  | 7. Menampilkan detail buku beserta rekomendasi buku yang terkait berdasarkan algoritma apriori. |
| 8. Klik button “Selesai” untuk menutup tab.  |   |
|  | 9. Aplikasi menutup tab hasil pencarian buku.   |
| Skenario Alternatif                          |   |
| Pengisian field pencarian buku belum lengkap |   |
| Mahasiswa                                    | System  |
| 4. Klik button “Cari Buku”.                  |   |
|  | 5. Menampilkan warning message “Please fill out this field”.                                    |

#### 4.2.4. Activity Diagram

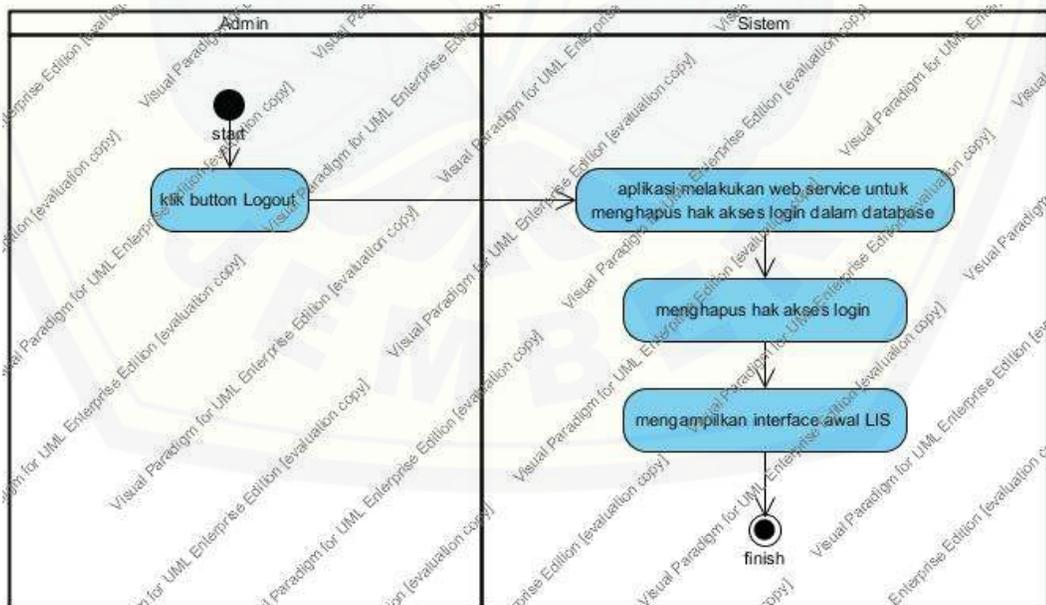
Activity Diagram aplikasi sistem klasifikasi rekomendasi buku perpustakaan ini berfungsi menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing – masing alir berawal, decision yang mungkin terjadi, hingga bagaimana alir tersebut berakhir. Activity diagram pada sistem ini ditunjukkan pada beberapa gambar di bawah ini.



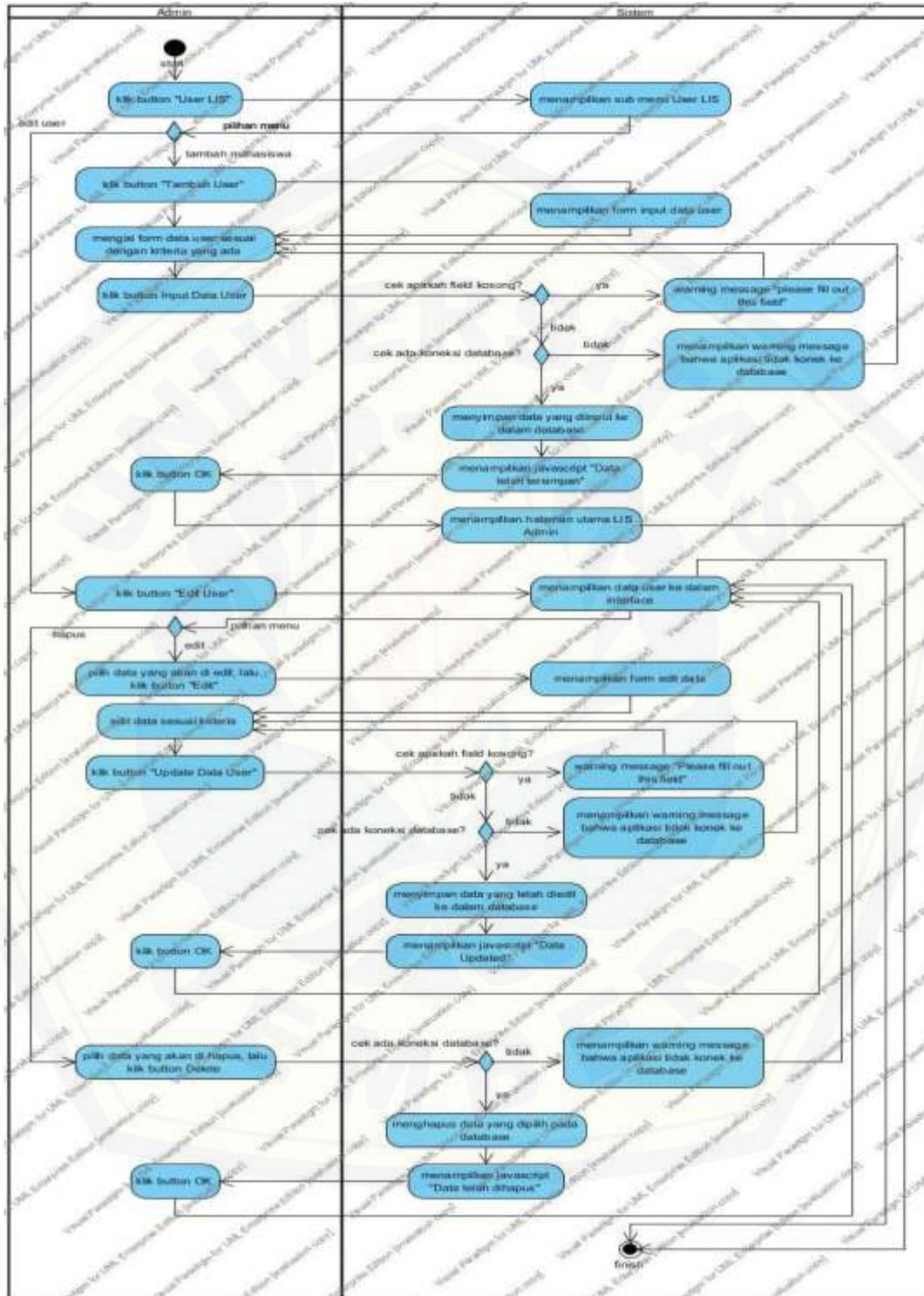
Gambar 4.3 Activity Diagram Login Admin



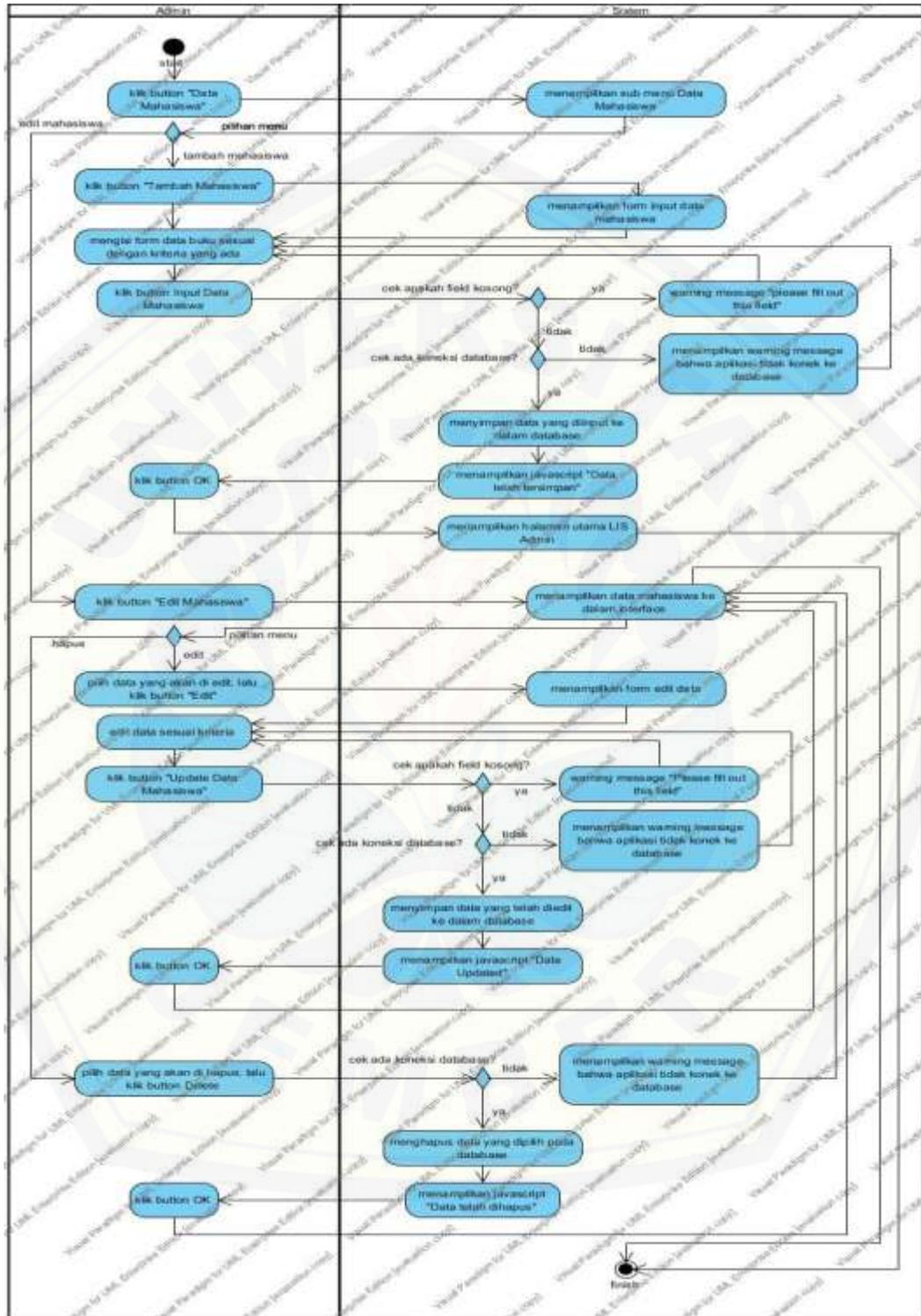
Gambar 4.4 Activity Diagram Login User



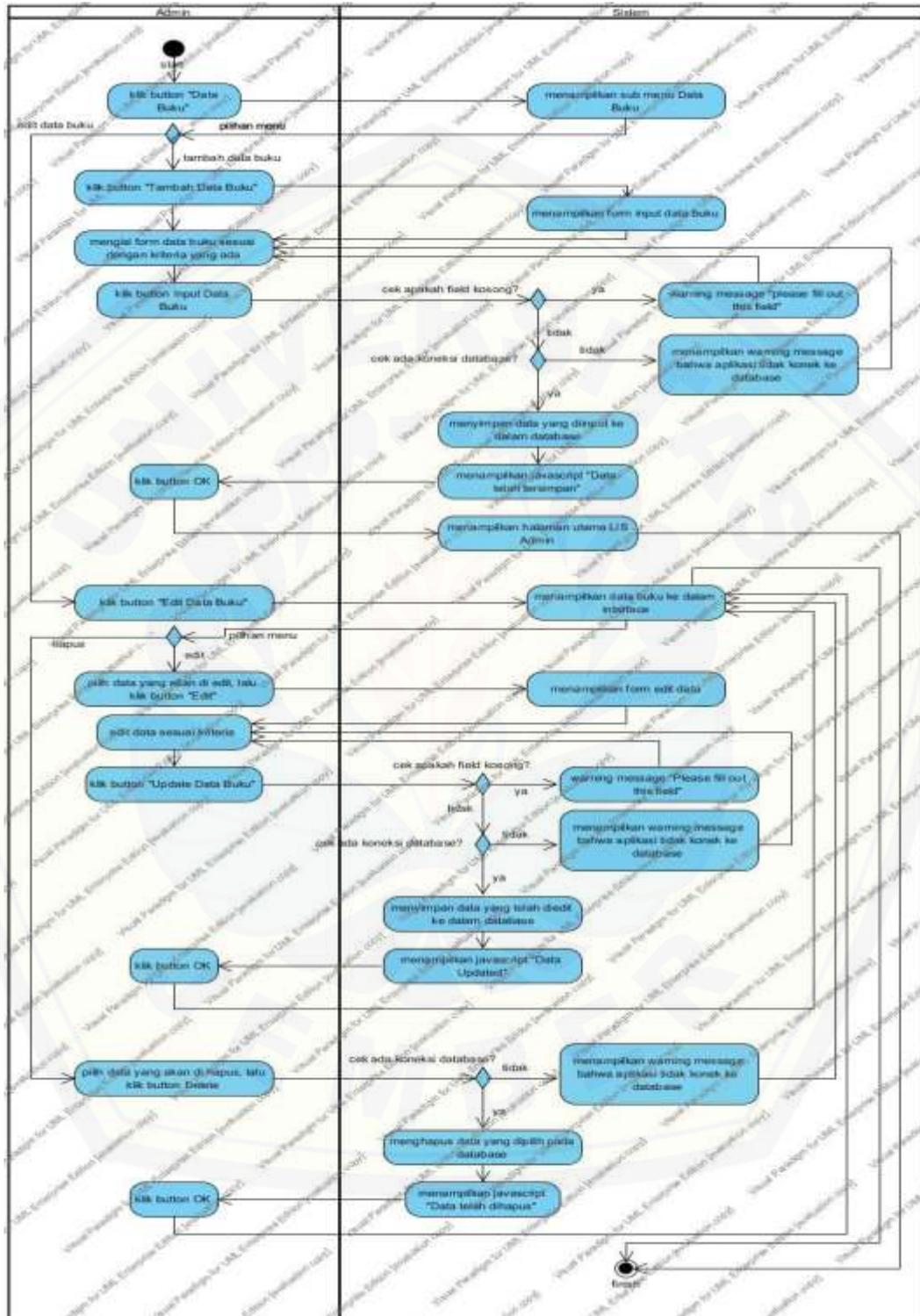
Gambar 4.5 Activity Diagram Logout



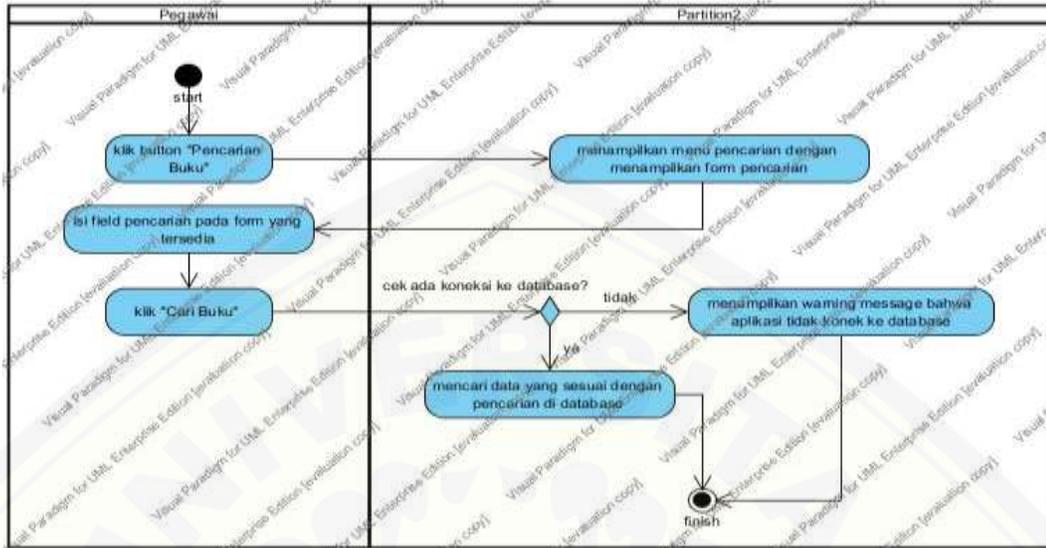
Gambar 4.6 Activity Diagram Manajemen Data User LIS



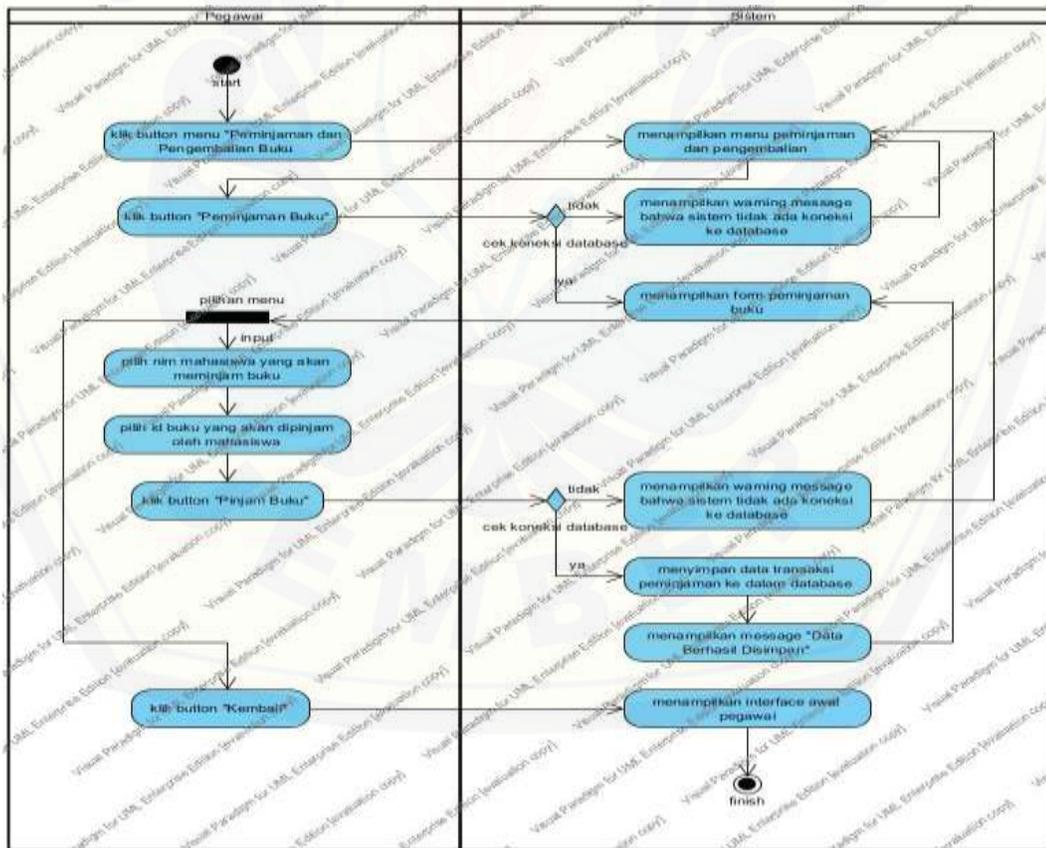
Gambar 4.7 Activity Diagram Manajemen Data Mahasiswa



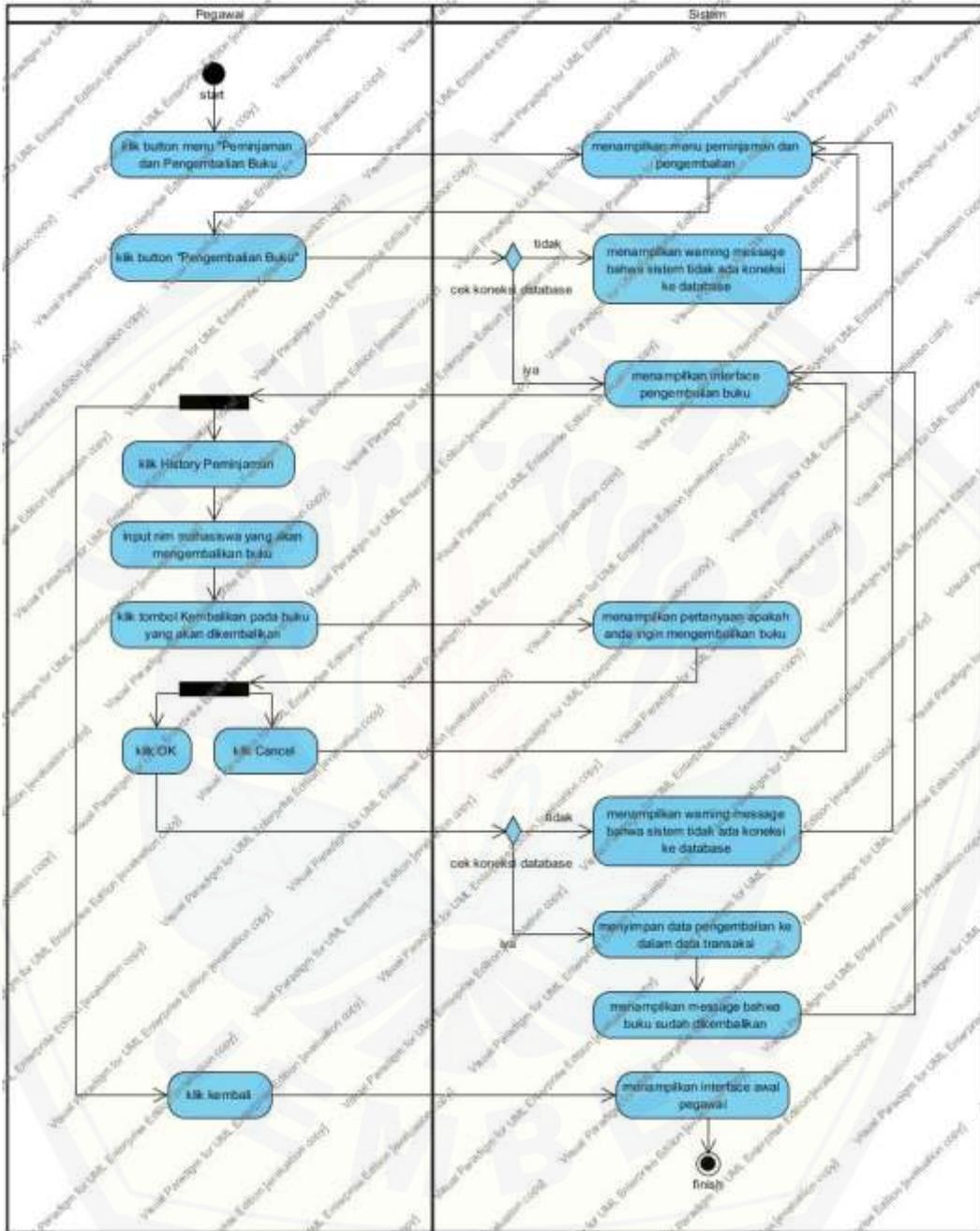
Gambar 4.8 Activity Diagram Manajemen Data Buku



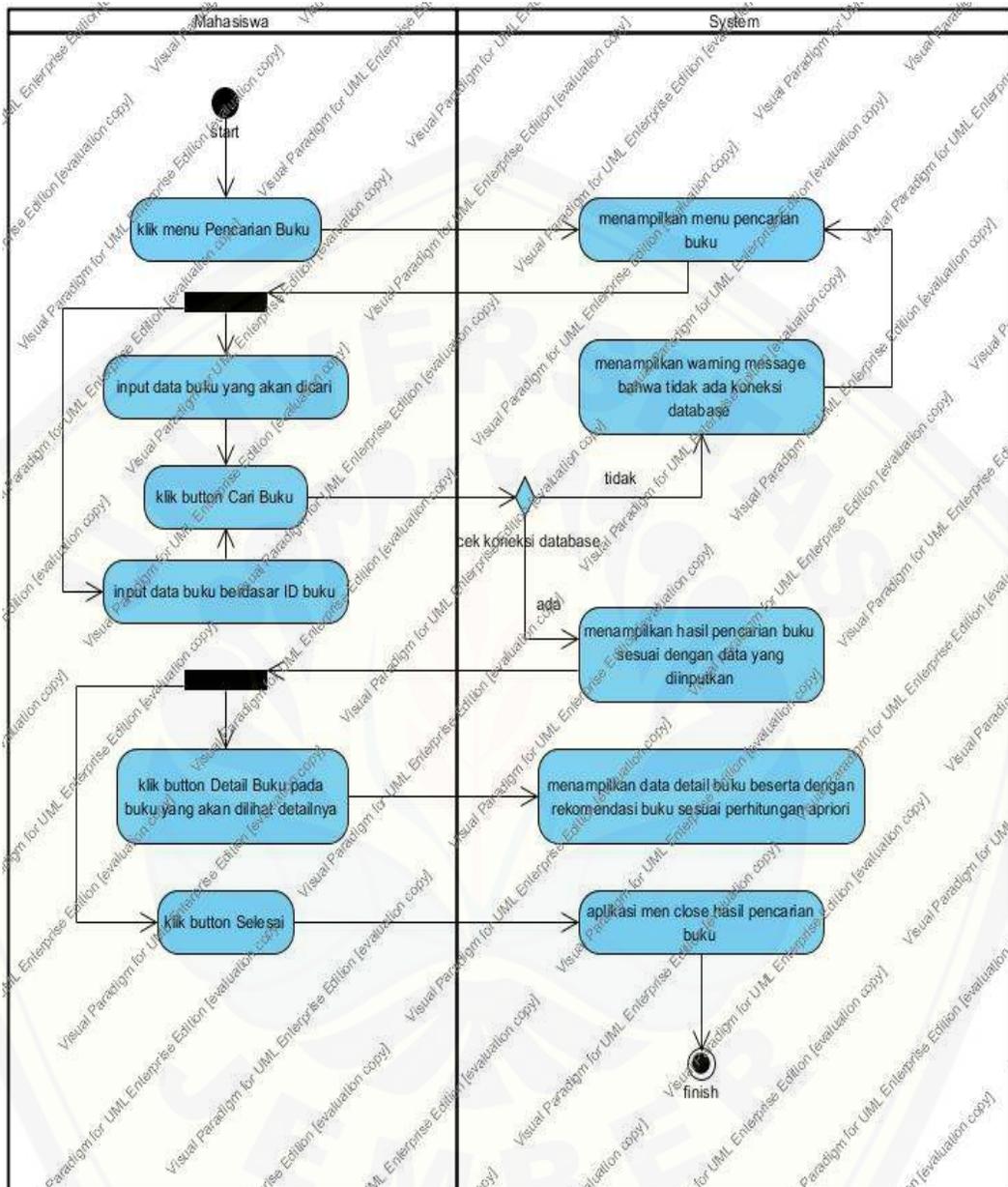
Gambar 4.9 Activity Diagram Pencarian Buku



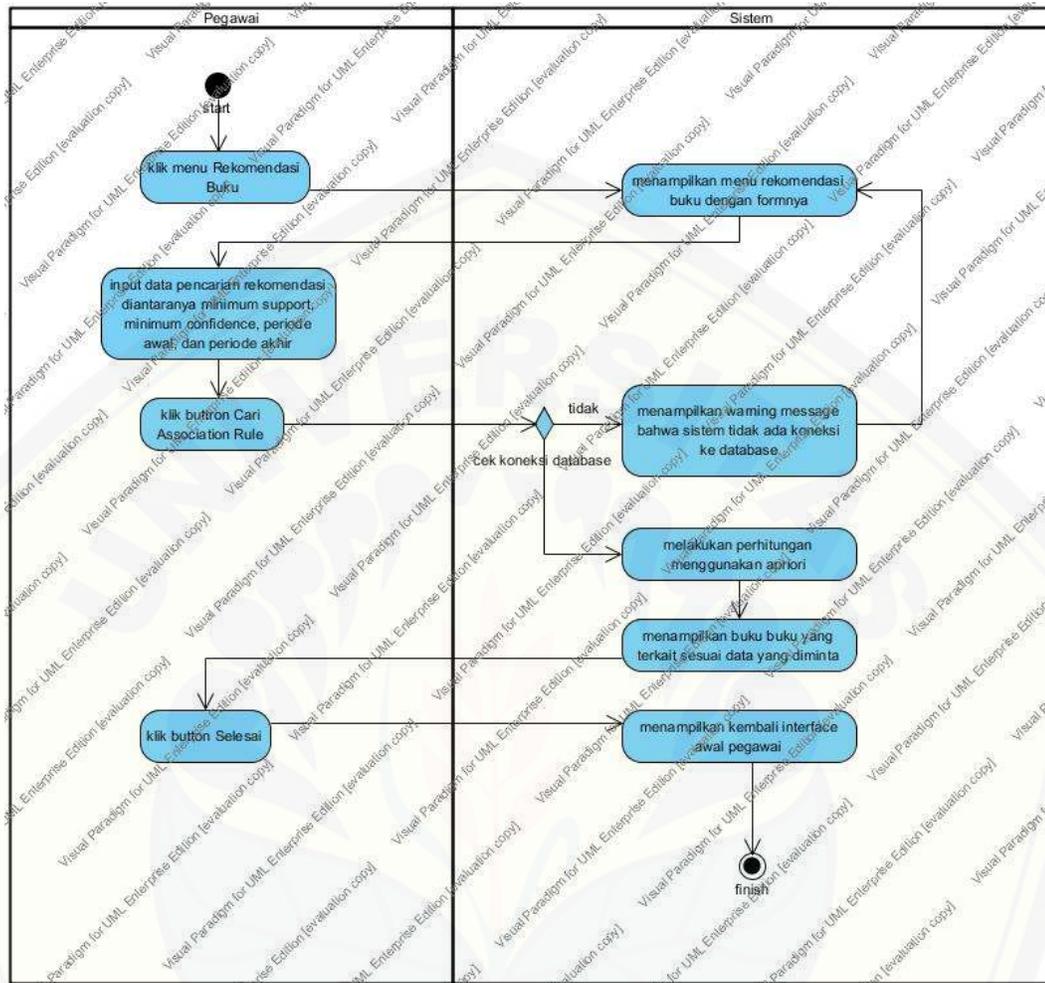
Gambar 4.10 Activity Diagram Peminjaman Buku



Gambar 4.11 Activity Diagram Pengembalian Buku



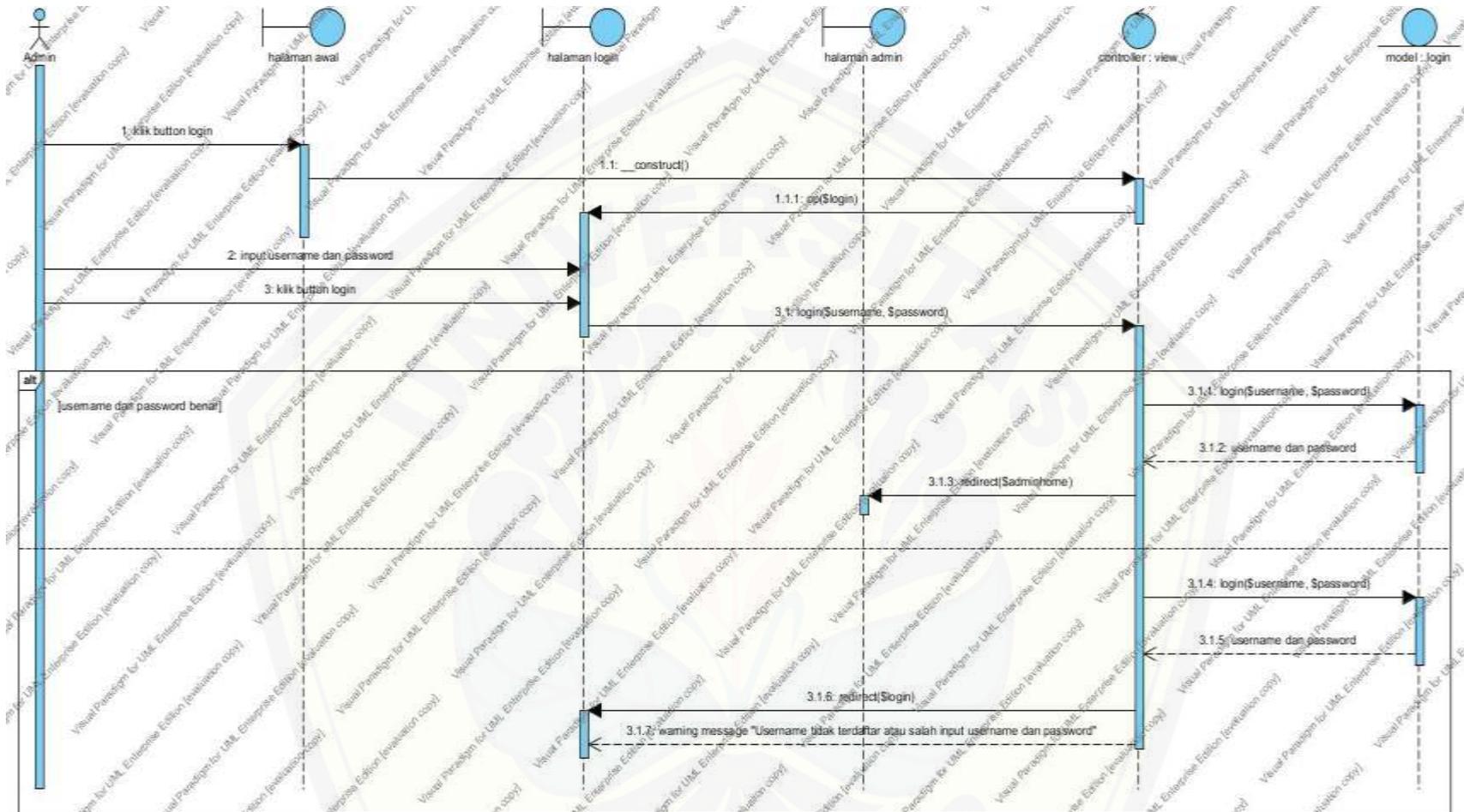
Gambar 4.12 Pencarian Buku oleh Mahasiswa



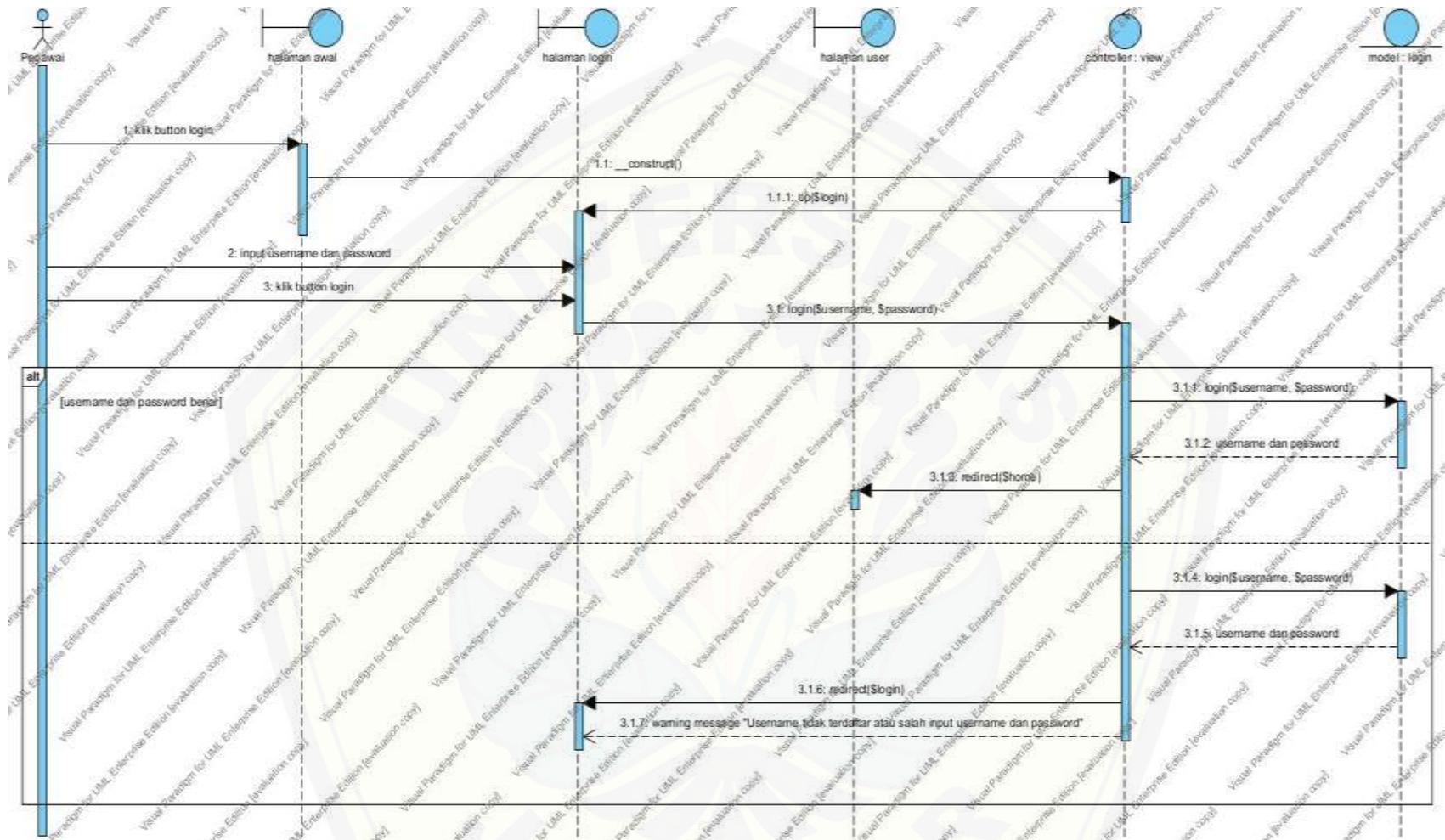
Gambar 4.13 Activity Diagram Rekomendasi Buku

#### 4.2.5. Sequence Diagram

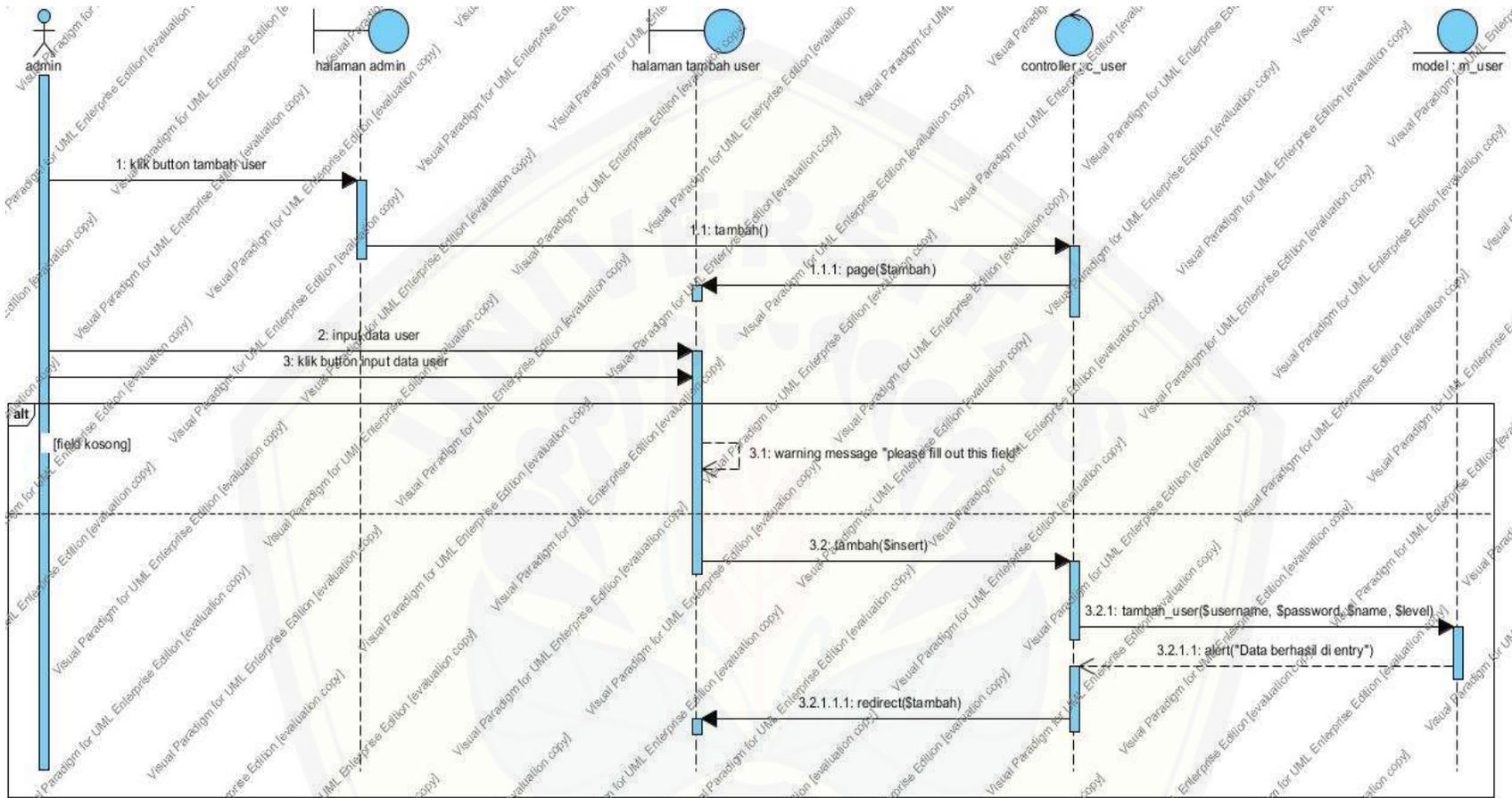
Proses perancangan selanjutnya yaitu dengan membuat sequence diagram. Sequence diagram merupakan dokumentasi suatu diagram terurut tentang interaksi antar objek mulai dari awal pada sampai akhir pada sebuah sistem. Sequence diagram digunakan untuk menggambarkan skenario dan memodelkan aliran logika pada skenario tersebut ke dalam sistem dengan menggunakan gambaran visual. Berikut penjelasan sistem klasifikasi rekomendasi buku perpustakaan menggunakan sequence diagram:



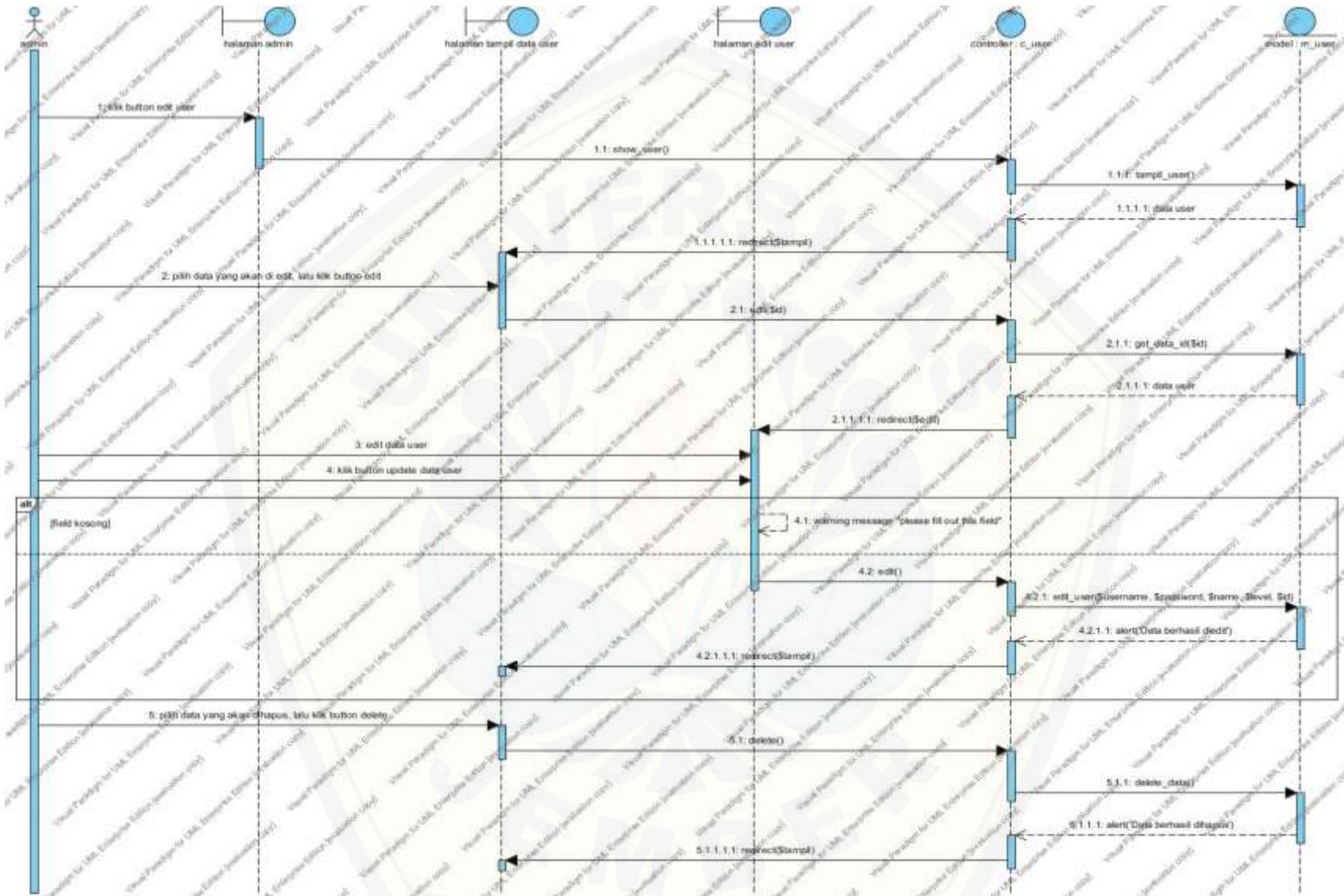
Gambar 4.14 Sequence Diagram Login Admin



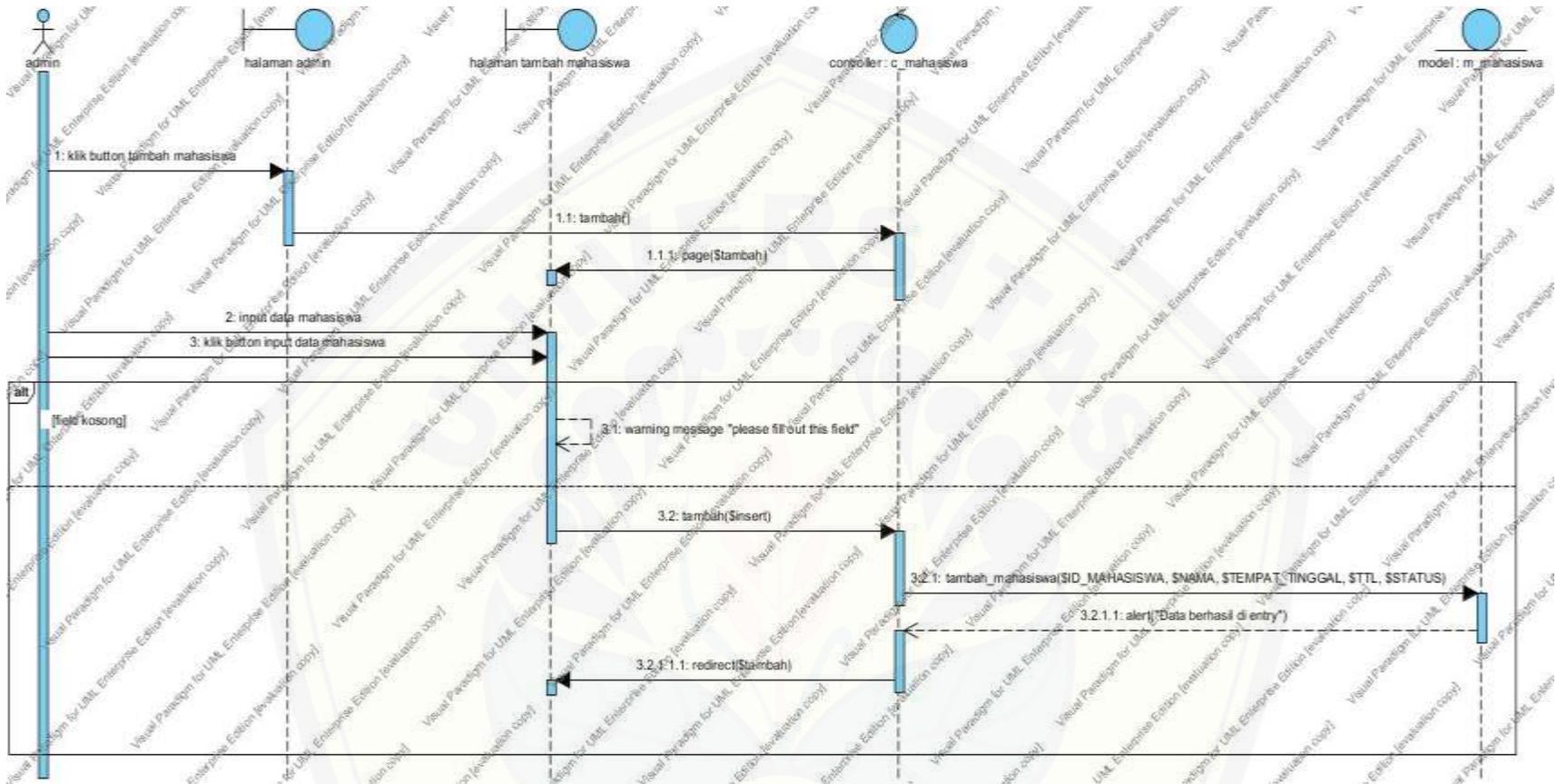
Gambar 4.15 Sequence Diagram Login Pegawai



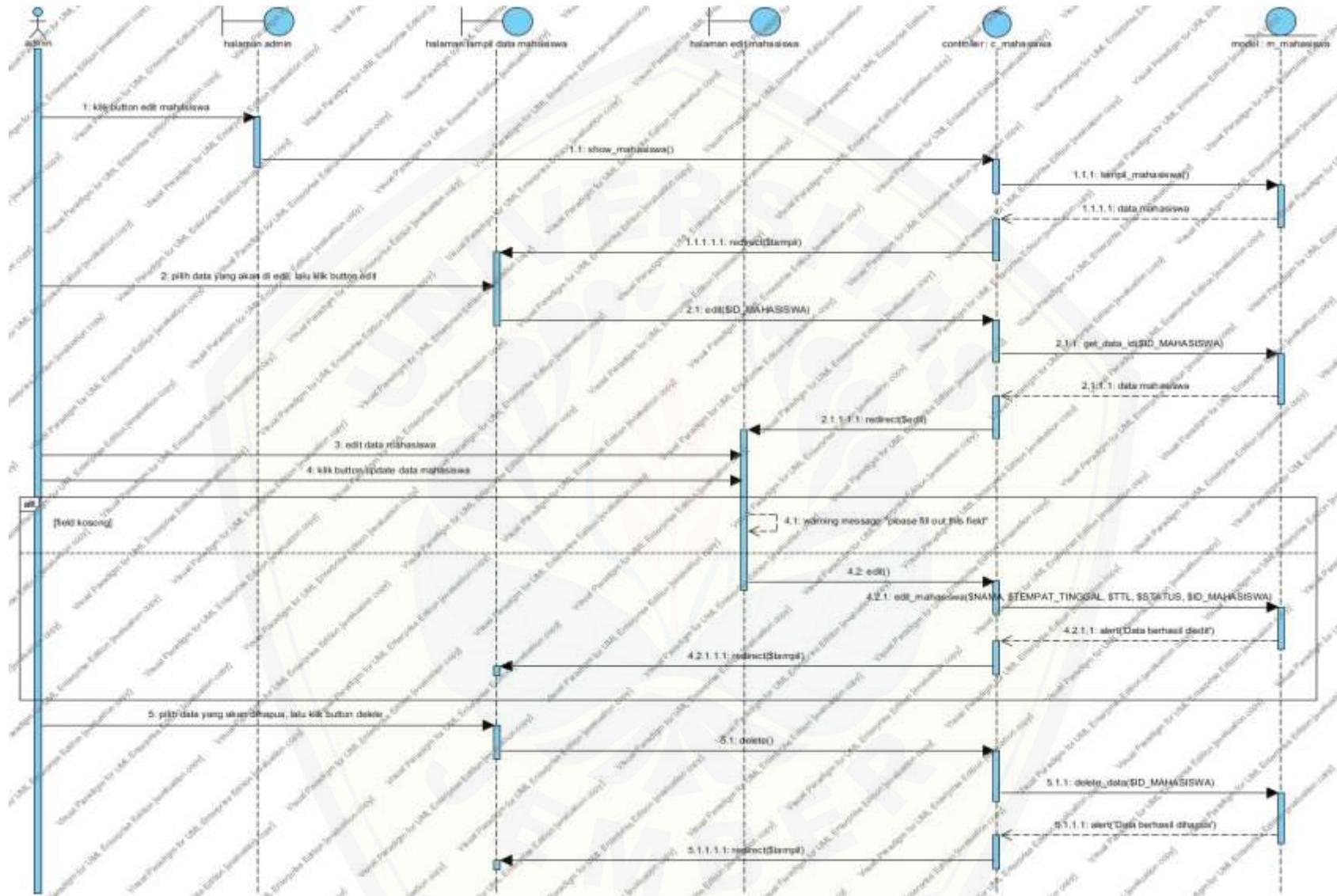
Gambar 4.16 Sequence Tambah Data User



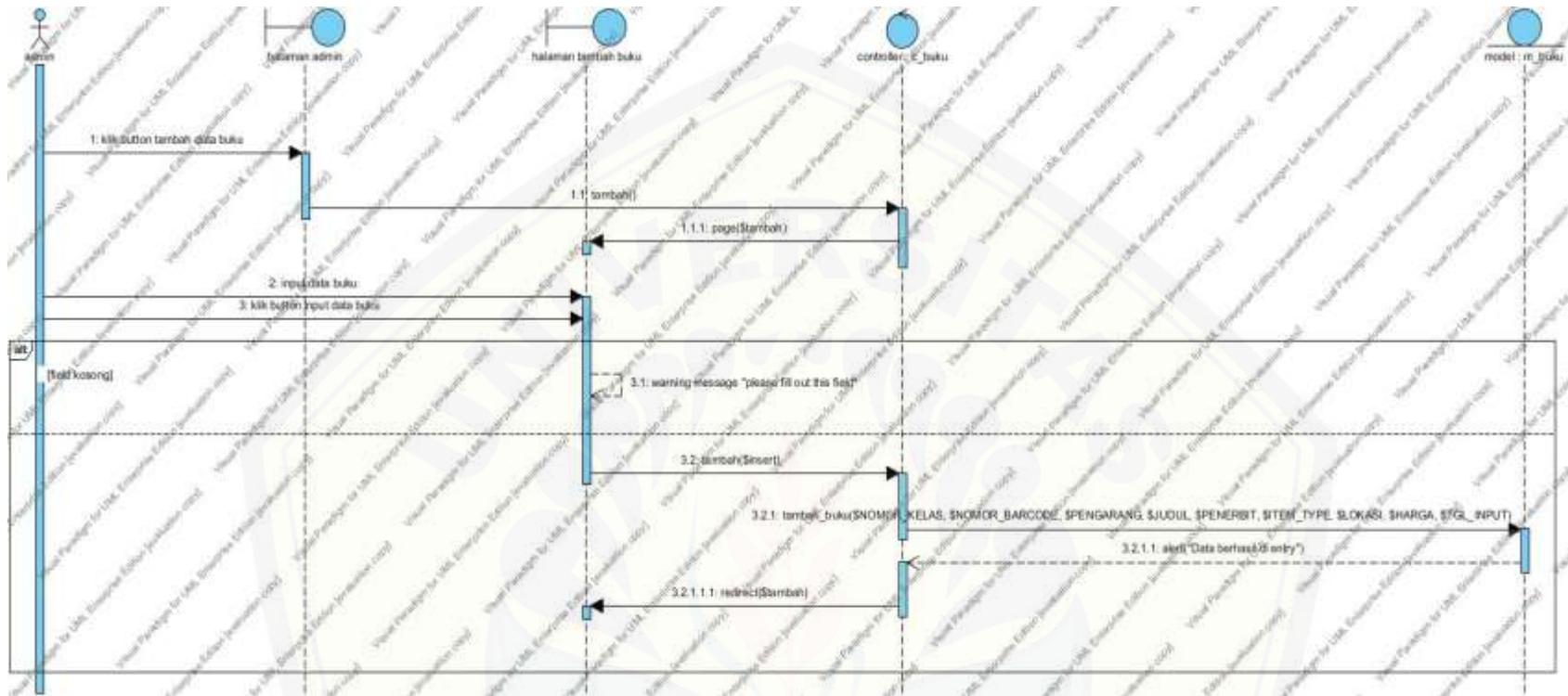
Gambar 4.17 Sequence Edit Data User



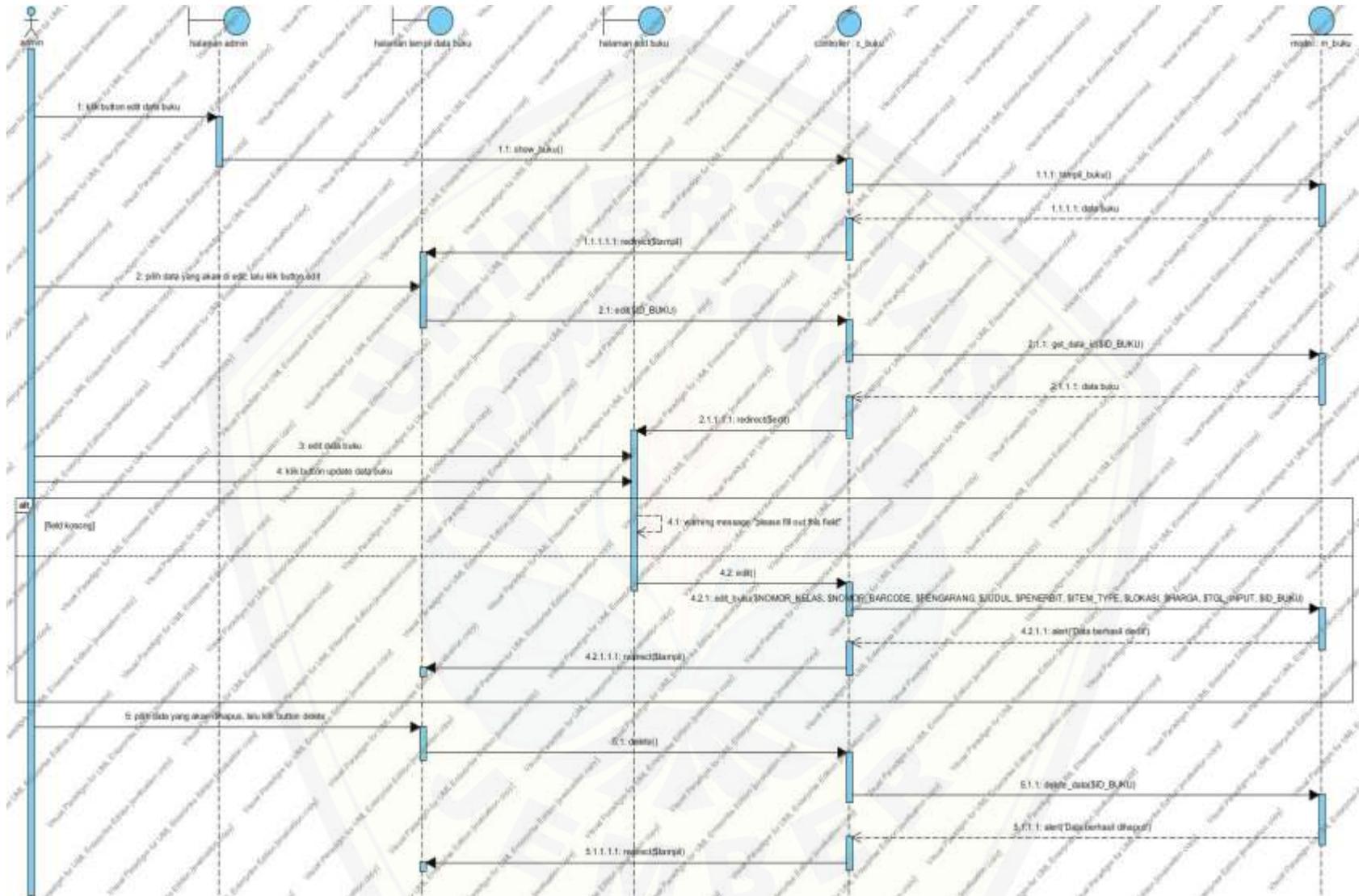
Gambar 4.18 Sequence Tambah Data Mahasiswa



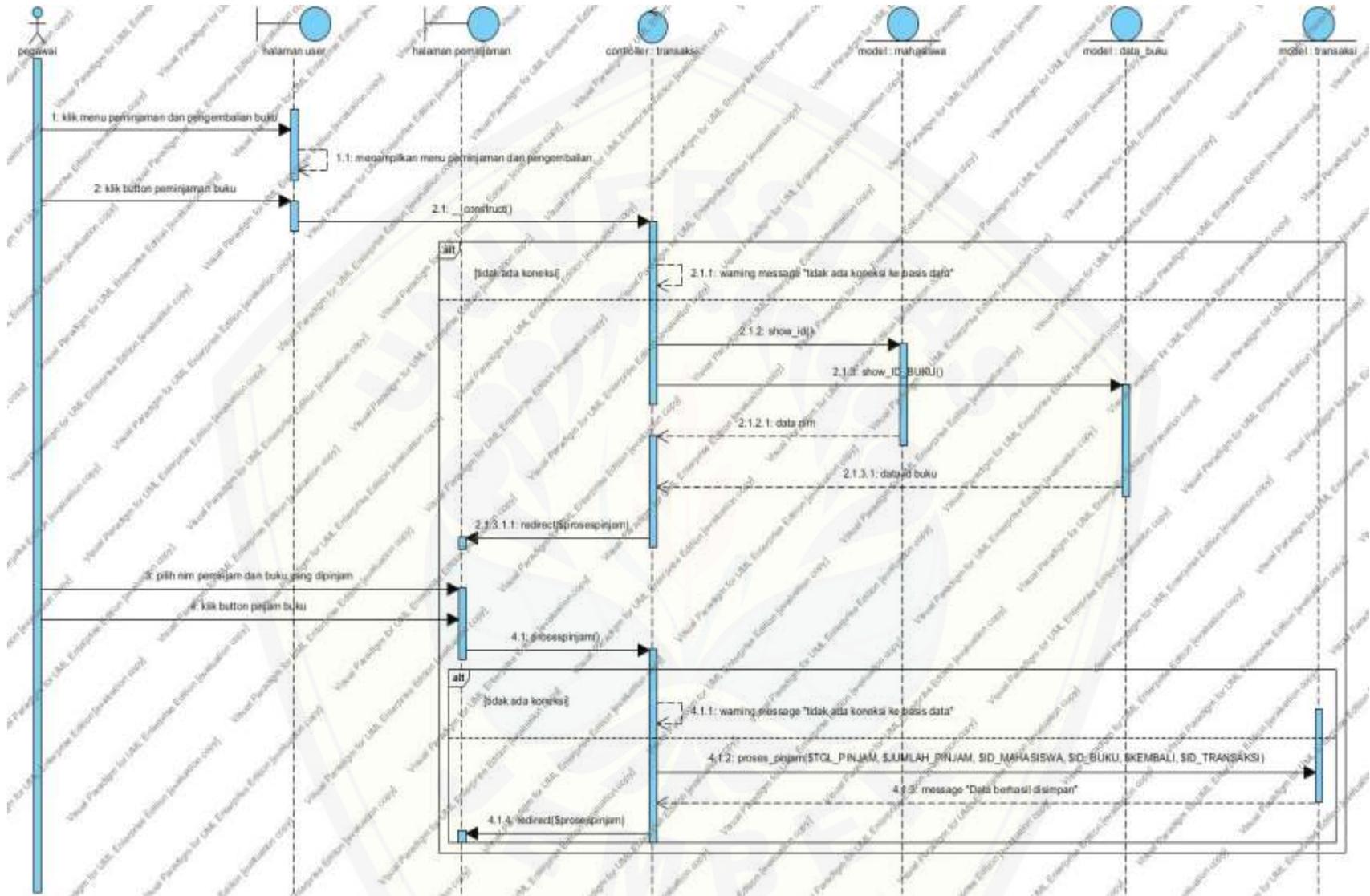
Gambar 4.19 Sequence Edit Data Mahasiswa



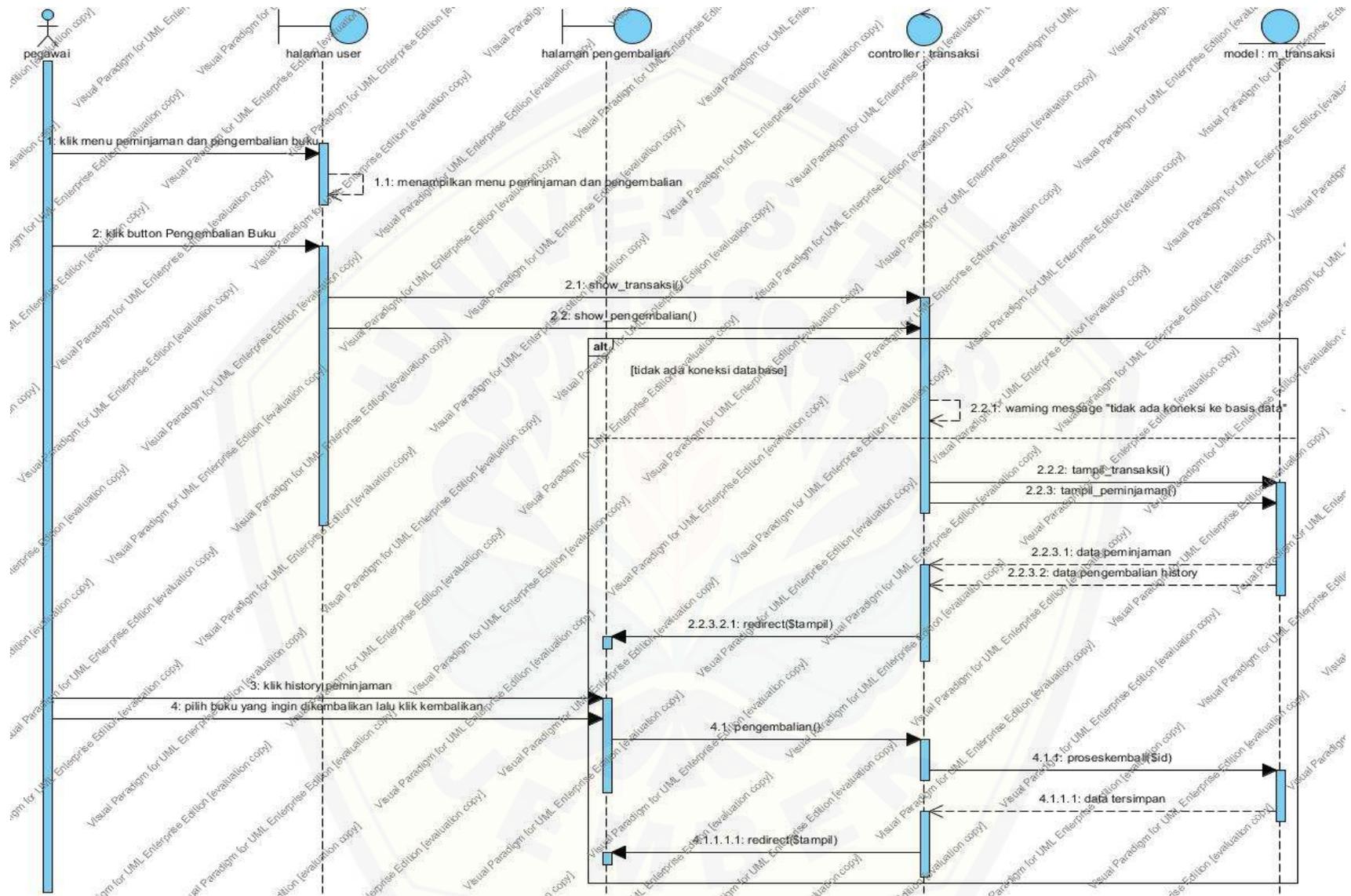
Gambar 4.20 Sequence Tambah Data Buku



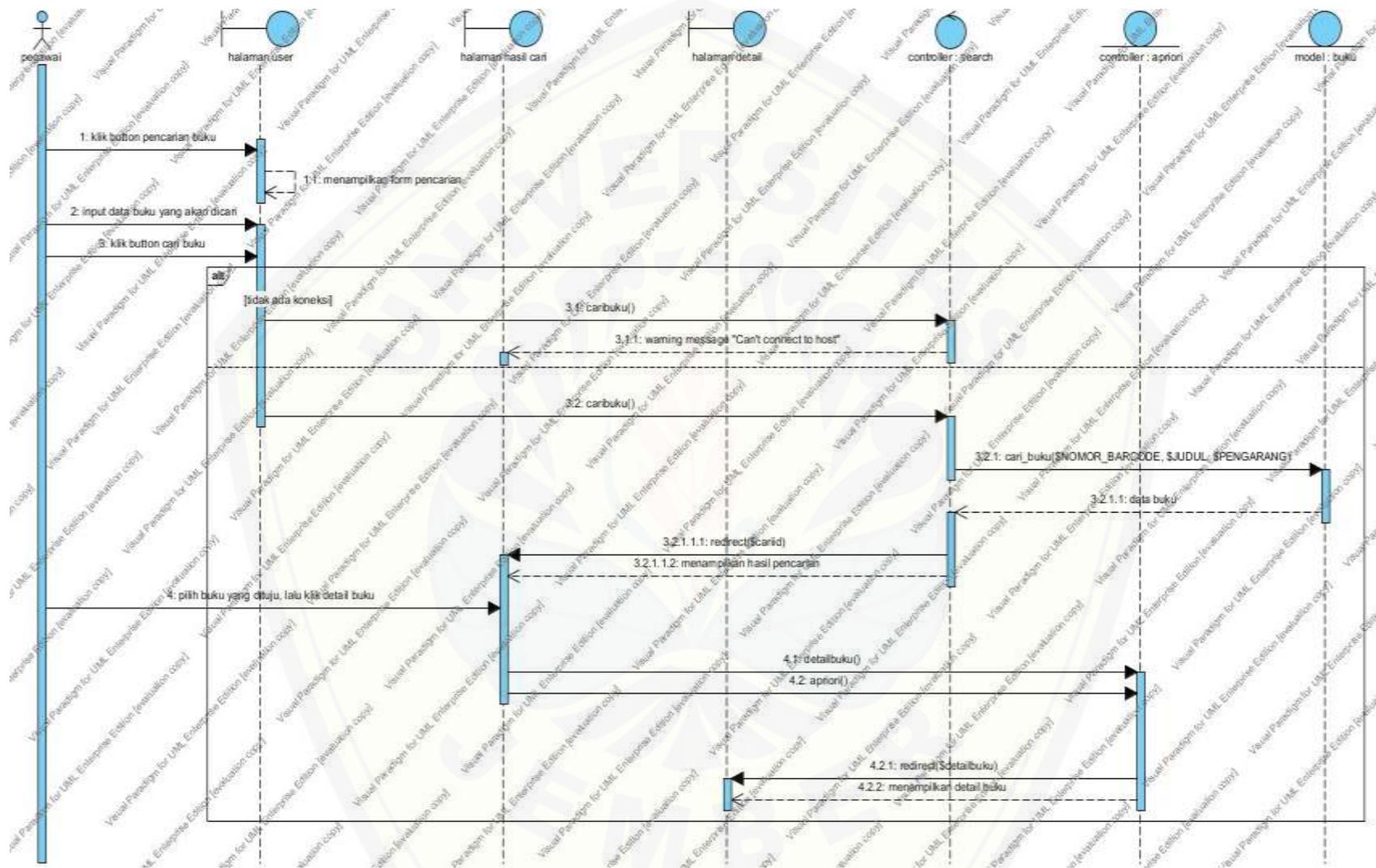
Gambar 4.21 Sequence Edit Data Buku



Gambar 4.22 Sequence Diagram Peminjaman Buku



Gambar 4.23 Sequence Diagram Pengembalian Buku

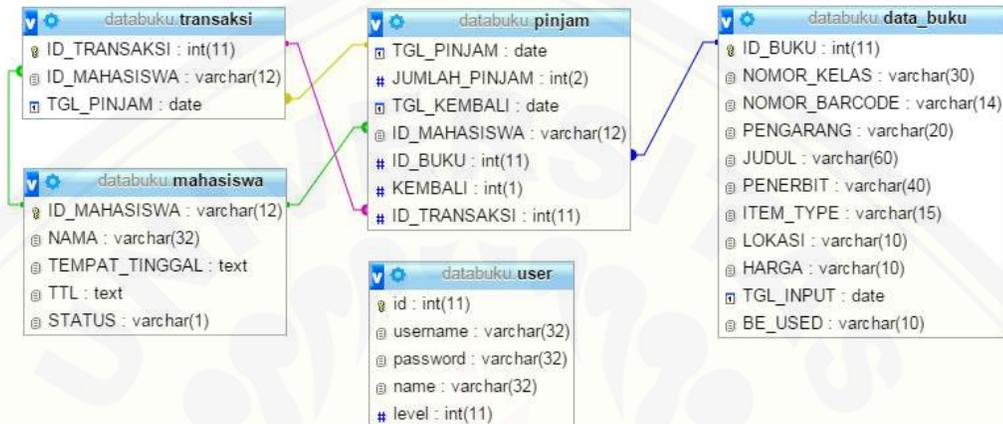


Gambar 4.24 Sequence Pencarian Buku Beserta Rekomendasi



## 4.2.7. Entity Relation Diagram

Setelah melalui tahap pembuatan desain class diagram, tahap selanjutnya yaitu membuat desain Entity Relation Diagram (ERD) yang akan dijelaskan pada gambar 4.26.



Gambar 4.26 Entity Relation Diagram

(Sumber: Hasil Analisis, 2015)

## 4.3. Implementasi Perancangan Sistem

Setelah tahap perancangan desain sistem selesai, tahap selanjutnya dalam penelitian sistem yaitu implementasi perancangan sistem ke dalam bahasa pemrograman. Bahasa pemrograman yang dipakai adalah bahasa pemrograman PHP (*Hypertext Preprocessor*) dan menggunakan *database* MySQL.

Pada tahap implementasi perancangan ini menjelaskan tentang fitur – fitur yang terdapat pada Sistem Klasifikasi Rekomendasi Buku Perpustakaan. Fitur-fitur tersebut meliputi sistem informasi perpustakaan ,managemen data user, managemen data buku, manajemen data mahasiswa, peminjaman dan pengembalian, rekomendasi buku dan lain-lainnya. Di dalam tahap ini juga mengimplementasikan algoritma apriori di dalam barisan kode program.

#### 4.4. Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian whitebox terlebih dahulu, lalu akan dilanjutkan dengan pengujian blackbox. Pengujian whitebox yang dilakukan pada penelitian ini diawali dengan pembuatan diagram alir dari listing program yang diujikan. Listing program yang diujikan dapat dilihat pada Gambar 4.27. Sedangkan untuk diagram alir pengujian dapat dilihat pada Gambar 4.28.

```

172 <?php
173
174 include 'koneksi.php';
175 include 'querySQL.php';
176
177 class tambahRule {
178     }
179     private $query;
180     private $result;
181
182     public function __construct() {
183         $conn = new connect();
184         $this->query = new query_sql();
185         $this->tambahRule();
186     }
187
188     function tambahRule() {
189         include '..\libraries\apriori\lib\apriori.class.php';
190
191         $q=mysql_query("
192             SELECT p.ID_TRANSAKSI as '0', p.ID_BUKU as '1' from data_transaksi p, data_buku db
193             WHERE p.ID_TRANSAKSI = t.ID_TRANSAKSI AND p.ID_BUKU=db.ID_BUKU");
194
195         $queryget = mysql_fetch_array(mysql_query("select * from data_buku where ID_BUKU=" . $i . $GET['id']));
196
197         $a = array();
198         while ($data = mysql_fetch_array($q)) {
199             array_push($a, array($data['0'], $data['1']));
200         }
201
202         //variables
203         $minSupp = '1'; //minimal support
204         $minConf = '1'; //minimal confidence
205         $type = Apriori::SRC_DB; //data type
206         $recomFor = 'beer'; //recommendation for
207
208         $dataFile = 'data.json.gz'; //file for saving of state
209
210         //some example of data source
211         $data = array();
212         switch ($type) {
213             case Apriori::SRC_PLAIN:
214                 break;
215             case Apriori::SRC_DB:
216                 $data = $a;
217                 // $data = $koneksi->result_array();
218                 // print_r($data);
219                 break;
220             case Apriori::SRC_CSV:
221                 break;
222         }
223
224         try {
225             $apri = new Apriori($type, $data, $minSupp, $minConf);
226             $apri->displayTransactions();
227             $apri->solve();
228             $apri->saveState($dataFile); //saving of state without rules
229             //print_r($apri);
230             unset($apri);
231
232             $sb = new Apriori(Apriori::SRC_LOAD, $dataFile); //load state and generate rules
233             $sb->generateRules();
234             $sb->displayRules();
235             $sb->displayRecommendations($recomFor);
236             $sb->saveState($dataFile); //save state with rules
237             $sprint = Apriori::loadAndPrintStateFile($dataFile, true);
238             //print_r($dataFile);
239             // $print = $dataFile;
240             // $data_pola['id_rule'] = $ID_POLA;
241
242             $stabel = array();

```

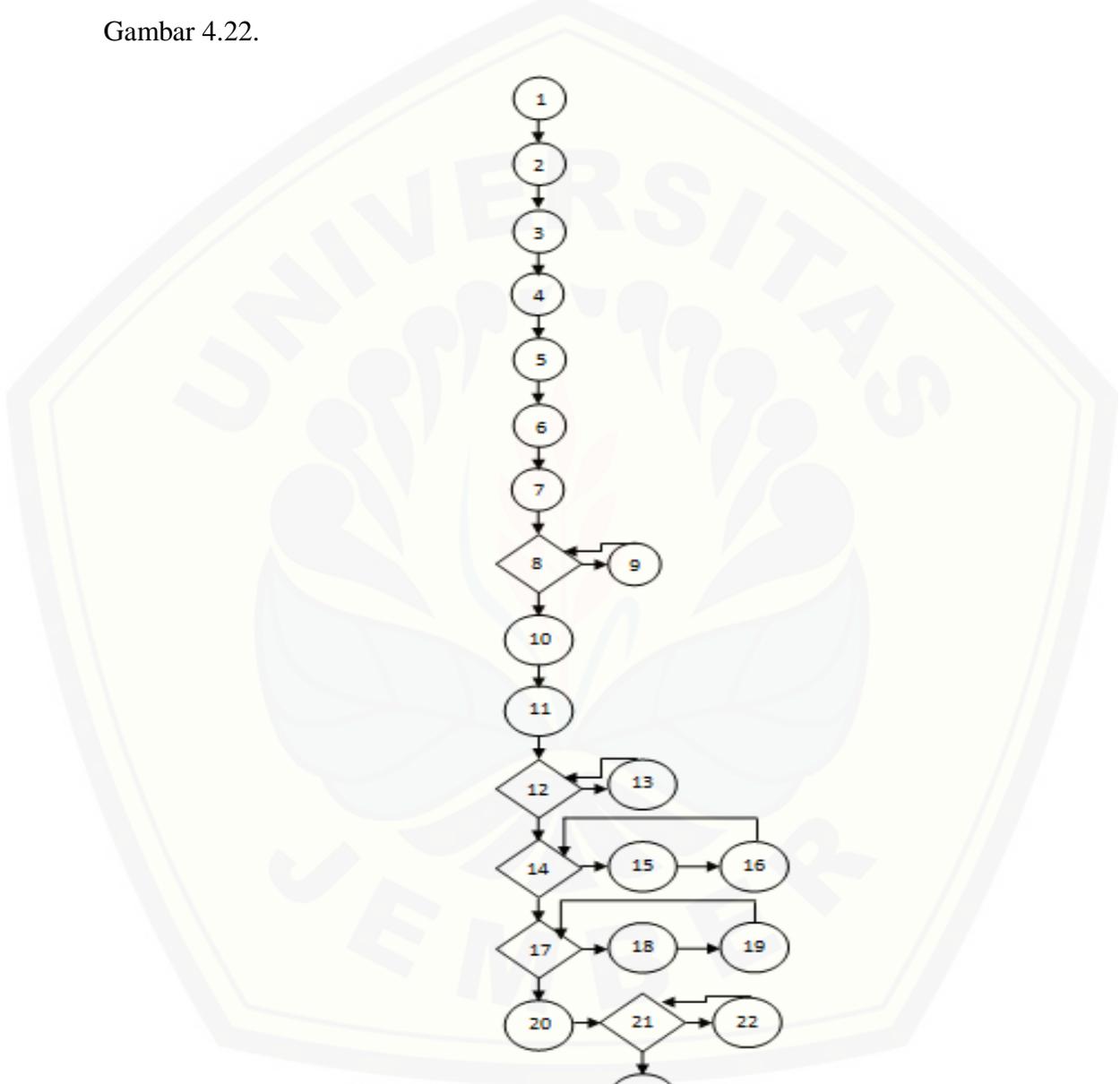
```

242     foreach ($print[4] as $key => $value) { } 14
243         foreach ($print[4][$key] as $keys => $values) { } 15
244             $i=1;
245             foreach ($print[4][$key][$keys] as $keyes => $values) {
246                 $nilai[$i] = $value;
247                 $i++;
248             }
249
250             $data_pola['buku_1'] = $key;
251             $data_pola['buku_2'] = $nilai[3];
252             $data_pola['nilai_support'] = $nilai[1];
253             $data_pola['nilai_confidence'] = $nilai[2];
254             //$data_pola['id_kategori'] = $rows['id_kategori'];
255             //$this->m_pola->simpan_detail_pola_apriori($data_pola); } 16
256
257             array_push($tabel, $data_pola);
258             //echo '<br /><pre>';
259             //print_r($data_pola);
260             //echo '</pre>';
261
262             $i=0;
263         }
264     }
265
306     foreach($tabel as $t) { } 17
307         $a = explode(',', $t['buku_1']); } 18
308         $bool = FALSE;
309         for ($i = 0, $ln = count($a); $i < $ln; $i++) { } 19
310             if ($a[$i] == $_GET['id']) {
311                 $bool = TRUE;
312             }
313
314             if ($bool) { } 20
315                 echo("<tr>");
316                 echo("<td>" . $queryget['JUDUL'] . "</td>");
317                 // echo("<td>" . $t['id_rule'] . "</td>");
318                 echo("<td>" . $t['buku_1'] . "</td>");
319                 echo("<td>" . $t['buku_2'] . "</td>");
320                 echo("<td>" . $t['nilai_support'] . "</td>");
321                 echo("<td>" . $t['nilai_confidence'] . "</td>");
322                 echo("</tr>");
323             }
324         }
325
326         foreach ($print[3] as $key => $value) { } 21
327             foreach ($print[3][$key] as $keys => $values) { } 22
328             }
329         }
330
331     } catch (Exception $exc) { } 23
332         echo $exc->getMessage();
333     }
334
335
336
337
338
339
340 $tambahRule = new tambahRule();

```

Gambar 4.27 Listing Program

Dari listing program diatas selanjutnya dapat dibuat tahap pengujian kedua, yaitu menggambar diagram alir dari penomoran pada listing program yang telah dilakukan pada tahap pertama. Diagram alir dari m\_detail.php dapat dilihat pada Gambar 4.22.



Gambar 4.28 Grafik Alir Fungsi Deskripsi

(Sumber: Hasil Analisis, 2015)

Untuk perhitungan *Cyclomatic Complexity* menggunakan perhitungan:

$$V(G) = E - N + 2$$

$$V(G) = 27 - 23 + 2$$

$$V(G) = 6$$

Dari perhitungan *Cyclomatic Complexity* di atas, maka basis set pengujian di atas terdapat 6 basis set, maka terdapat 6 jalur, diantaranya:

Jalur 1:

1,2,3,4,5,6,7,8,10,11,12,14,17,20,21,23

Jalur 2:

1,2,3,4,5,6,7,8,9,10,11,12,14,17,20,21,23

Jalur 3:

1,2,3,4,5,6,7,8,10,11,12,13,14,17,20,21,23

Jalur 4:

1,2,3,4,5,6,7,8,10,11,12,14,15,16,17,20,21,23

Jalur 5:

1,2,3,4,5,6,7,8,10,11,12,14,17,18,19,20,21,23

Jalur 6:

1,2,3,4,5,6,7,8,10,11,12,14,17,20,21,22,23

Setelah mendapatkan 6 jalur tersebut, selanjutnya dicek menggunakan tabel test case untuk mendeteksi apakah sistem benar – benar berjalan atau tidak saat menjalankan ke 6 jalur tersebut.

Tabel 4.12 *Test Case* Perhitungan Algoritma Apriori

| Jalur | Kasus Uji                       | Luaran yang diharapkan | Kesimpulan                |
|-------|---------------------------------|------------------------|---------------------------|
| 1     | Jalur dengan kondisi semua true | Hasil rekomendasi buku | [√] Berhasil<br>[ ] Gagal |

|   |   |  |                           |
|---|---|--|---------------------------|
| 2 | Jalur dengan kondisi mengubah database menjadi data array                   | Data array yang telah diubah                     | [√] Berhasil<br>[ ] Gagal |
| 3 | Jalur mengambil data array untuk diproses dengan perhitungan apriori        | Data array beserta rules                         | [√] Berhasil<br>[ ] Gagal |
| 4 | Jalur kondisi mengubah data array menjadi value                             | Value data tersimpan dengan nama yang ditentukan | [√] Berhasil<br>[ ] Gagal |
| 5 | Jalur kondisi sortir data buku_1 untuk menampilkan data ganda dalam 1 tabel | Data ganda dengan dipisah oleh “,” pada 1 tabel  | [√] Berhasil<br>[ ] Gagal |
| 6 | Jalur kondisi cetak data yang telah diproses ke dalam tampilan              | Data yang sudah terproses ditampilkan            | [√] Berhasil<br>[ ] Gagal |

Selanjutnya dilakukan pengujian blackbox yang pada tabel 4.13 ini merupakan blackbox pada bagian metode apriori, dan untuk pengujian blackbox yang lain tertera di lampiran.

Tabel 4.13 Pengujian blackbox terhadap pencarian rekomendasi buku

| No | User    | Fitur                    | Kasus Uji                      | Luaran yang diharapkan     | Status       |
|----|---------|--------------------------|--------------------------------|----------------------------|--------------|
| 1. | Pegawai | Mencari Rekomendasi Buku | Klik menu                      | Menampilkan form pencarian | [√] Berhasil |
|    |         |                          | Rekomendasi Buku               | rekomenadasi buku          | [ ] Gagal    |
|    |         |                          | Input data ke dalam field form | Melakukan perhitungan      | [√] Berhasil |

|    |           |                                |  |  |  |
|----|-----------|--------------------------------|--|--|--|
|    |           |                                | pencarian rekomendasi buku, lalu klik cari association rules | apriori dan selanjutnya menampilkan data sesuai dengan inputan                     | <input type="checkbox"/> Gagal   |
|    |           |                                | Salah 1 field kosong   | Menampilkan warning message “please fill out this field”                           | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
| 2. | Mahasiswa | Mencari buku serta rekomendasi | Input data buku lalu klik cari                               | Menampilkan buku yang dicari   | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |
|    |           |                                | Klik detail buku untuk melihat rekomendasi buku              | Menampilkan detail buku beserta rekomendasi buku sesuai dengan perhitungan apriori | <input checked="" type="checkbox"/> Berhasil<br><input type="checkbox"/> Gagal |

## BAB 6 PENUTUP

Pada bab ini merupakan bagian akhir di dalam penulisan skripsi, berisi tentang kesimpulan dan saran. Kesimpulan yang ditulis merupakan kesimpulan dari hasil penelitian yang telah dilakukan dan saran lanjutan untuk dilakukan pada penelitian selanjutnya.

### 6.1. Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Aplikasi Sistem Pengklasifikasian Rekomendasi Buku Perpustakaan Universitas Jember yang dibuat mampu memberikan fungsi lebih ke dalam perpustakaan Universitas Jember dalam hal informasi, yaitu membantu mahasiswa Universitas Jember ataupun peminjam umum dalam mendapatkan hasil rekomendasi buku sesuai dengan buku yang dipinjam.
2. Algoritma Apriori dapat memberikan hasil rekomendasi dan bisa menghasilkan data yang lebih akurat menggunakan data transaksi sebelumnya. Hasil rekomendasi buku dihasilkan dari perhitungan apriori dimana semakin sering data – data yang bersangkutan saling berinteraksi dalam transaksi, semakin besar persentase *support* dan semakin sering transaksi buku 1 dengan yang lainnya dibanding dengan buku rekomendasi lain, maka nilai *confidence* yang didapat juga akan menjadi lebih besar.

### 6.2. Saran

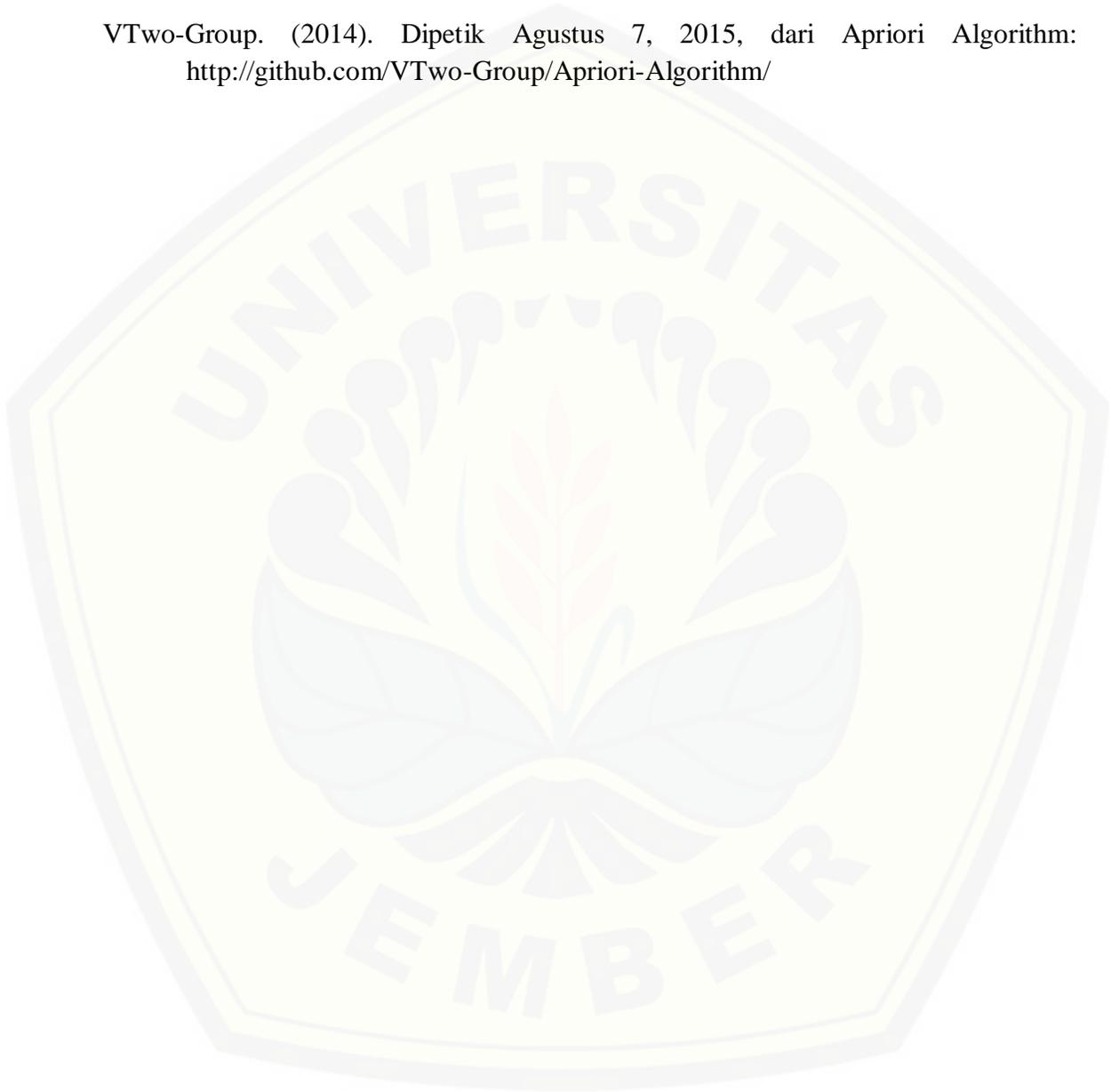
Pengembangan lebih lanjut untuk penelitian ini dapat dilakukan dengan membangun aplikasi sistem pengklasifikasian rekomendasi buku perpustakaan Universitas Jember ini menggunakan platform lain, diantaranya untuk smartphone sebagai aplikasi, sehingga mahasiswa dapat mengakses rekomendasi buku melalui smartphone mereka masing – masing. Dapat juga dikembangkan sistem ini dengan mengkombinasikan metode apriori ini dengan metode lainnya sehingga dapat memunculkan informasi yang lebih lengkap dari rekomendasi buku.

**DAFTAR PUSTAKA**

- Antonio, H., & Safriadi, N. (2012). Rancang Bangun Sistem Informasi Administrasi Informatika (SI-ADIF). *Jurnal ELKHA Vol 4 No 2* , 12-15.
- Heriyanto, A. P. (2013). ANALISIS PEMANFAATAN BUKU ELEKTRONIK (E-BOOK) OLEH PEMUSTAKA DI PERPUSTAKAAN SMA NEGERI 1 SEMARANG. *JURNAL ILMU PERPUSTAKAAN Volume 2 Nomor 2* , 1-9.
- Jnanamurthy, Vishesh, Jain, V., Kumar, P., & M.Pai, R. (2013). Discovery of Maximal Frequent Item Sets using Subset Creation. *International Journal of Data Mining & Knowledge Management Proccess(IJDKP) Vol 3 No 1* , 27-38.
- Pane, D. K. (2013). IMPLEMENTASI DATA MINING PADA PENJUALAN PRODUK ELEKTRONIK DENGAN ALGORITMA APRIORI (STUDI KASUS: KREDITPLUS). *Pelita Informatika Budi Darma Vol IV No 3* , 25-29.
- Ruldeviyani, Y., & Fahrian, M. (2008). Implementasi Algoritma - Algoritma Association Rules Sebagai Bagian Dari Pengembangan Data Mining Algorithms Collection. *Konferensi Nasional Sistem dan Informatika KNS&I08-043* .
- Saleh, A. R. (2009). PERAN PUSTAKAWAN DALAM DISSEMINASI INFORMASI KEPADA PENELITIAN VIA JURNAL ELEKTRONIK LOKAL: KASUS PERPUSTAKAAN IPB. *VISI PUSTAKA Volume 11 Nomor 3* .
- Santosa, B. (2007). Data Mining Teknik Pemanfaatan Data Untuk Keperluan Bisnis.
- Srinivas, K., Rao, G. R., & Govardhan, A. (2011). SURVEY ON PREDICTION OF HEART MORDIBITY USING DATA MINING TECHNIQUES. *International Journal of Data Mining & Knowledge Management Proccess (IJDKP) Vol 1 No 3* , 14-34.
- Syaifullah, M. A. (2010). IMPLEMENTASI DATAMINING ALGORITMA APRIORI PADA SISTEM PENJUALAN.

Verma, S. (2014). Analysis of Strengths and Weakness of SDLC Models. *International Journal of Advance Research in Computer Science and Management Studies Vol 2 Issue 3* , 235-240.

VTwo-Group. (2014). Dipetik Agustus 7, 2015, dari Apriori Algorithm: <http://github.com/VTwo-Group/Apriori-Algorithm/>



LAMPIRAN

Tabel 1. Pengujian blackbox pada aplikasi LIS untuk User Admin

| No | Fitur | Aksi                              | Hasil  | Status                       |
|----|-------|-----------------------------------|--|------------------------------|
| 1. | Login | Username dan password benar       | Menampilkan halaman utama admin dengan menampilkan fitur: <ul style="list-style-type: none"> <li>• Halaman utama</li> <li>• Tentang Aplikasi</li> <li>• User LIS</li> <li>• Data Mahasiswa</li> <li>• Data Buku</li> </ul> | [√]<br>Berhasil<br>[ ] Gagal |
|    |       | Username dan password salah       | Menampilkan warning message:<br>“Username tidak terdaftar atau salah input username dan password!”   | [√]<br>Berhasil<br>[ ] Gagal |
|    |       | Username atau password kosong     | Menampilkan warning message:<br>“Please fill out this field”   | [√]<br>Berhasil<br>[ ] Gagal |
|    |       | Username benar dan password salah | Menampilkan warning message:<br>“Username tidak terdaftar atau salah input username dan password!”   | [√]<br>Berhasil<br>[ ] Gagal |
|    |       | Username salah dan password benar | Menampilkan warning message:<br>“Username tidak terdaftar  | [√]<br>Berhasil<br>[ ] Gagal |

|    |                       |  |   |                              |
|----|-----------------------|--|---|------------------------------|
|    |                       |  | atau salah input username dan password!”                  |                              |
| 2. | Insert data user LIS  | Tambah User  | Menampilkan form tambah user                              | [√]<br>Berhasil<br>[ ] Gagal |
|    |                       | Mengisi form tambah user lalu klik input data user | Data tersimpan ke dalam database                          | [√]<br>Berhasil<br>[ ] Gagal |
|    |                       | Salah 1 field pada form tambah user tidak terisi   | Menampilkan warning message: “Please fill out this field” | [√]<br>Berhasil<br>[ ] Gagal |
| 3. | Update data user LIS  | Klik menu Edit User                                | Menampilkan data user LIS                                 | [√]<br>Berhasil<br>[ ] Gagal |
|    |                       | Klik icon edit pada data yang akan diupdate        | Menampilkan form edit data user                           | [√]<br>Berhasil<br>[ ] Gagal |
|    |                       | Mengupdate data user lalu klik update data user    | Data yang telah terupdate disimpan ke dalam database      | [√]<br>Berhasil<br>[ ] Gagal |
|    |                       | Salah 1 form pada form edit data user tidak terisi | Menampilkan warning message “Please fill out this field”  | [√]<br>Berhasil<br>[ ] Gagal |
|    |                       | Klik icon delete pada data yang akan dihapus       | Data terhapus dari database                               | [√]<br>Berhasil<br>[ ] Gagal |
| 4. | Insert data Mahasiswa | Tambah Mahasiswa                                   | Menampilkan form tambah mahasiswa                         | [√]<br>Berhasil              |

|    |                           |  |  |   |
|----|---------------------------|--|--|---|
|    | LIS                       |  |  | <input type="checkbox"/> Gagal  |
|    |                           | Mengisi form tambah mahasiswa lalu klik input data mahasiswa | Data tersimpan ke dalam database                             | <input checked="" type="checkbox"/><br>Berhasil<br><input type="checkbox"/> Gagal |
|    |                           | Salah 1 field pada form tambah mahasiswa tidak terisi        | Menampilkan warning message:<br>“Please fill out this field” | <input checked="" type="checkbox"/><br>Berhasil<br><input type="checkbox"/> Gagal |
| 5. | Update data mahasiswa LIS | Klik menu Edit Mahasiswa                                     | Menampilkan data mahasiswa LIS                               | <input checked="" type="checkbox"/><br>Berhasil<br><input type="checkbox"/> Gagal |
|    |                           | Klik icon edit pada data yang akan diupdate                  | Menampilkan form edit data mahasiswa                         | <input checked="" type="checkbox"/><br>Berhasil<br><input type="checkbox"/> Gagal |
|    |                           | Mengupdate data mahasiswa lalu klik update data mahasiswa    | Data yang telah terupdate disimpan ke dalam database         | <input checked="" type="checkbox"/><br>Berhasil<br><input type="checkbox"/> Gagal |
|    |                           | Salah 1 form pada form edit data mahasiswa tidak terisi      | Menampilkan warning message “Please fill out this field”     | <input checked="" type="checkbox"/><br>Berhasil<br><input type="checkbox"/> Gagal |
|    |                           | Klik icon delete pada data yang akan dihapus                 | Data terhapus dari database                                  | <input checked="" type="checkbox"/><br>Berhasil<br><input type="checkbox"/> Gagal |
| 6. | Insert data Buku LIS      | Tambah Buku  | Menampilkan form tambah buku                                 | <input checked="" type="checkbox"/><br>Berhasil                                   |

|    |                      |  |  |                              |
|----|----------------------|--|--|------------------------------|
|    |                      |  |  | [ ] Gagal                    |
|    |                      | Mengisi form tambah buku lalu klik input data buku | Data tersimpan ke dalam database                             | [√]<br>Berhasil<br>[ ] Gagal |
|    |                      | Salah 1 field pada form tambah buku tidak terisi   | Menampilkan warning message:<br>“Please fill out this field” | [√]<br>Berhasil<br>[ ] Gagal |
| 7. | Update data buku LIS | Klik menu Edit data buku                           | Menampilkan data buku LIS                                    | [√]<br>Berhasil<br>[ ] Gagal |
|    |                      | Klik icon edit pada data yang akan diupdate        | Menampilkan form edit data buku                              | [√]<br>Berhasil<br>[ ] Gagal |
|    |                      | Mengupdate data buku lalu klik update data buku    | Data yang telah terupdate disimpan ke dalam database         | [√]<br>Berhasil<br>[ ] Gagal |
|    |                      | Salah 1 form pada form edit data buku tidak terisi | Menampilkan warning message “Please fill out this field”     | [√]<br>Berhasil<br>[ ] Gagal |
|    |                      | Klik icon delete pada data yang akan dihapus       | Data terhapus dari database                                  | [√]<br>Berhasil<br>[ ] Gagal |
| 8. | Logout               | Klik button logout                                 | Hak akses diputus lalu kembali ke halaman utama LIS          | [√]<br>Berhasil<br>[ ] Gagal |

Tabel 2. Pengujian blackbox pada aplikasi LIS untuk User Pegawai

| No | Fitur | Aksi | Hasil | Status |
|----|-------|------|-------|--------|
|----|-------|------|-------|--------|

|    |       |                                   |  |                                 |
|----|-------|-----------------------------------|--|---------------------------------|
| 1. | Login | Username dan password benar       | Menampilkan halaman utama admin dengan menampilkan fitur: <ul style="list-style-type: none"> <li>• Halaman utama</li> <li>• Tentang Aplikasi</li> <li>• Pencarian Buku</li> <li>• Peminjaman dan Pengembalian Buku</li> <li>• Rekomendasi</li> </ul> | [√]<br>Berhasil<br>[ ]<br>Gagal |
|    |       | Username dan password salah       | Menampilkan warning message:<br>“Username tidak terdaftar atau salah input username dan password!”   | [√]<br>Berhasil<br>[ ]<br>Gagal |
|    |       | Username atau password kosong     | Menampilkan warning message:<br>“Please fill out this field”   | [√]<br>Berhasil<br>[ ]<br>Gagal |
|    |       | Username benar dan password salah | Menampilkan warning message:<br>“Username tidak terdaftar atau salah input username dan password!”   | [√]<br>Berhasil<br>[ ]<br>Gagal |
|    |       | Username salah dan password benar | Menampilkan warning message:<br>“Username tidak terdaftar atau salah input username dan password!”   | [√]<br>Berhasil<br>[ ]<br>Gagal |

|    |                   |  |   |                                 |
|----|-------------------|--|---|---------------------------------|
| 2. | Pencarian Buku    | Mengisi form pencarian lalu klik cari buku     | Menampilkan halaman hasil pencarian sesuai dengan input yang dicari                                       | [√]<br>Berhasil<br>[ ]<br>Gagal |
|    |                   | Klik button detail buku pada buku yang dituju  | Menampilkan hasil detail buku beserta perhitungan apriori berisi buku rekomendasi, support dan confidence | [√]<br>Berhasil<br>[ ]<br>Gagal |
|    |                   | Klik button selesai pada tabel hasil pencarian | Close tab hasil pencarian dan kembali ke halaman pencarian  | [√]<br>Berhasil<br>[ ]<br>Gagal |
| 3. | Peminjaman Buku   | Klik button Peminjaman Buku                    | Menampilkan form peminjaman buku  | [√]<br>Berhasil<br>[ ]<br>Gagal |
|    |                   | Mengisi form peminjaman dan klik pinjam buku   | Menyimpan data peminjaman / data transaksi peminjaman ke dalam database lalu kembali ke form peminjaman   | [√]<br>Berhasil<br>[ ]<br>Gagal |
|    |                   | Klik button kembali                            | Menampilkan halaman awal pegawai  | [√]<br>Berhasil<br>[ ]<br>Gagal |
| 4. | Pengembalian Buku | Klik button Pengembalian buku                  | Menampilkan halaman pengembalian buku   | [√]<br>Berhasil<br>[ ]<br>Gagal |

|    |                  |   |   |                                 |
|----|------------------|---|---|---------------------------------|
|    |                  | Klik history peminjaman   | Menampilkan tabel transaksi peminjaman  | [√]<br>Berhasil<br>[ ]<br>Gagal |
|    |                  | Klik button<br>Kembalikan pada buku yang akan dikembalikan                  | Menyimpan data transaksi peminjaman menjadi data buku yang sudah dikembalikan   | [√]<br>Berhasil<br>[ ]<br>Gagal |
|    |                  | Klik button history pengembalian  | Menampilkan tabel transaksi pengembalian  | [√]<br>Berhasil<br>[ ]<br>Gagal |
|    |                  | Klik button kembali   | Menampilkan halaman awal pegawai  | [√]<br>Berhasil<br>[ ]<br>Gagal |
| 5. | Rekomendasi buku | Memasukkan data yang diperlukan pada form, lalu klik cari association rules | Melakukan perhitungan lalu ditampilkan ke dalam tabel hasil rekomendasi secara keseluruhan berdasar periode yang ditentukan | [√]<br>Berhasil<br>[ ]<br>Gagal |