

**PENERAPAN ALGORITMA APRIORI UNTUK PENCARIAN POLA
PEMINJAMAN BUKU DI PERPUSTAKAAN
STMIK PALANGKARAYA**

TUGAS AKHIR

Disusun untuk Memenuhi Syarat Kelulusan Program Strata I
Pada Sekolah Tinggi Manajemen Informatika Dan Komputer
(STMIK) Palangkaraya



OLEH

MARTIN

NIM C1755201079

PROGRAM STUDI TEKNIK INFORMATIKA

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2021**

**PENERAPAN ALGORITMA APRIORI UNTUK PENCARIAN POLA
PEMINJAMAN BUKU DI PERPUSTAKAAN
STMIK PALANGKARAYA**

TUGAS AKHIR

Disusun untuk Memenuhi Syarat Kelulusan Program Strata I
Pada Sekolah Tinggi Manajemen Informatika Dan Komputer
(STMIK) Palangkaraya

OLEH

MARTIN

NIM C1755201079

PROGRAM STUDI TEKNIK INFORMATIKA

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA
2021**

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama Mahasiswa : **MARTIN**

NIM : **C1755201079**

Menyatakan bahwa Tugas Akhir dengan judul :

**PENERAPAN ALGORITMA APRIORI UNTUK PENCARIAN POLA
PEMINJAMAN BUKU DI PERPUSTAKAAN
STMIK PALANGKARAYA**

Adalah hasil karya saya dan bukan merupakan duplikasi sebagian atau seluruhnya dari karya orang lain.

Pernyataan ini dibuat dengan sebenar-benarnya secara sadar dan bertanggung jawab dan saya bersedia menerima sanksi pembatalan Tugas Akhir apabila terbukti melakukan duplikasi terhadap Tugas Akhir atau karya ilmiah lain yang sudah ada.

Palangka Raya, Juli 2021

Yang membuat pernyataan,


MARTIN

PERSETUJUAN

PENERAPAN ALGORITMA APRIORI UNTUK PENCARIAN POLA PEMINJAMAN BUKU DI PERPUSTAKAAN STMIK PALANGKARAYA

Tugas Akhir ini telah disetujui untuk
diujikan pada Tanggal 5 juli 2021

Dosen Pembimbing I,



Hotmian Sitohang, M.Kom
NIK. 198503282008002

Dosen Pembimbing II,



Elok Faiqotul Himmah, S.Si., M.Sc
NIK. 198503092009003

Mengetahui :

Kepala STMIK Palangkaraya,



Suparno, M.Kom
NIK. 196901041995105

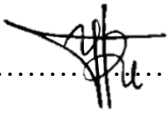
PENGESAHAN

PENERAPAN ALGORITMA APRIORI UNTUK PENCARIAN POLA PEMINJAMAN BUKU DI PERPUSTAKAAN STMIK PALANGKARAYA

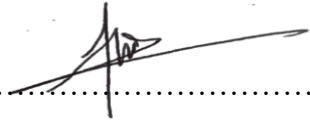
Tugas Akhir ini telah Diuji, Dinilai dan Disahkan
Oleh Tim Penguji Pada Tanggal 14 juli 2021

Tim Penguji :

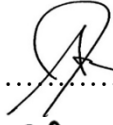
1. Sulistyowati, S.Kom., M.Cs
Ketua

: 

2. Herkules, S.Kom., M.Cs.
Sekretaris

: 

3. Sam'ani, S.T, M.Kom
Anggota

: 

4. Hotmian Sitohang, M.Kom.
Anggota

: 

5. Elok Faiqotul Himmah, S.Si., M.Sc.
Anggota

: 

MOTTO DAN PERSEMBAHAN

Aku tahu, bahwa engkau sanggup melakukan segala sesuatu, dan tidak ada rencana-Mu yang gagal. (Ayub 42:2)

Tugas Akhir Ini saya persembahkan untuk

- Ayah dan Ibu saya, yang telah mendidik, mendukung, dan membesarkan saya hingga sekarang, serta selalu mendo'akan yang terbaik untuk saya.
- Diri saya sendiri yang telah berjuang demi mendapatkan gelar sarjana yang saya yakini akan berguna untuk masa depan saya yang akan datang.
- Sahabat saya Yuniuss yang telah menemani, meluangkan waktu dan memberikan dukungan untuk saya dari awal perkuliahan sampai saya bisa menyelesaikan Tugas Akhir ini.
- Teman-teman Teknik Informatika kelas B Angkatan 2017, yang tidak kenal lelah untuk saling membantu dan memberikan dukungannya.
- Dosen-dosen STMIK Palangkaraya, yang telah memberikan ilmu serta membimbing saya dari awal kuliah sampai pada akhirnya saya lulus dan mendapatkan gelar sarjana komputer (S.Kom)

INTISARI

Martin, C1755201079,2021, *Penerapan Algoritma Apriori Untuk Pencarian Pola Peminjaman Buku Di Perpustakaan STMIK Palangkaraya*, Pembimbing I Hotmian Sitohang, M.Kom. Pembimbing II Elok Faiqotul Himmah, S.Si., M.Sc.

Perpustakaan merupakan tempat yang menyediakan begitu banyak buku, namun dalam kenyataannya peminjam sering kali juga kebingungan dalam memilih buku sehingga perlu dibuat suatu sistem aplikasi yang dapat melakukan analisis agar dapat merekomendasikan buku-buku untuk dipinjam untuk melakukan hal tersebut diperlukan suatu penerapan algoritma ke dalam sebuah sistem aplikasi, algoritma apriori menggunakan *association rule mining* adalah salah satu metode data mining yang dapat mengidentifikasi hubungan kesamaan antar item. Adapun tujuan dari penelitian ini adalah membuat suatu aplikasi yang dapat melakukan analisis dari hasil data transaksi peminjaman serta memberikan informasi mengenai rekomendasi buku yang sering dipinjam yang diterapkan pada perangkat komputer.

Dalam melakukan penelitian Tugas Akhir ini ada metode yang digunakan oleh penulis yang meliputi metode pengumpulan data dan metode pengembangan sistem. Metode pengumpulan data tersebut adalah observasi, wawancara dan studi literatur. Sedangkan metode pengembangan sistem yang penulis gunakan adalah metode *Rapid Application Development* (RAD).

Hasil akhir dari penelitian ini adalah berupa aplikasi Penerapan Algoritma Apriori Untuk Pencarian Pola Peminjaman Buku Di Perpustakaan STMIK Palangkaraya dengan mengimplementasikan algoritma apriori ke dalam sebuah aplikasi untuk mendapatkan informasi rekomendasi buku untuk dipinjam dari hasil analisis.

KATA KUNCI : Apriori, *Data mining*, *Rapid Application Development* (RAD).

ABSTRACT

Martin, C1755201079,2021, *Penerapan Algoritma Apriori Untuk Pencarian Pola Peminjaman Buku Di Perpustakaan STMIK Palangkaraya*, Pembimbing I Hotmian Sitohang, M.Kom. Pembimbing II Elok Faiqotul Himmah, S.Si., M.Sc.

Library is a place that provides so many books, but in reality borrowers are often also confused in choosing a book so it is necessary to create an application system that can do analysis in order to recommend books to be borrowed to do so is required an application algorithm into an application system, a priori algorithm using association rule mining is one of the methods of data mining that can identify h similarity between items. The purpose of this research is to create an application that can analyze the results of loan transaction data as well as provide information about the recommendations of books that are often borrowed that are applied to computer devices.

In conducting this Final Task research there are two methods used by the author which include data collection methods and system development methods. The methods of collecting such data are observations, interviews and literature studies. While the system development method that the author uses is the Rapid Application Development (RAD) method.

The final result of this study is in the form of application of A priori algorithm application for finding book lending patterns in STMIK Palangkaraya Library by implementing a priori algorithm into an application to get book recommendation information to borrow from the analysis results.

KEYWORDS: A priori, Data mining, Rapid Application Development(RAD).

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadiran Tuhan yang maha esa yang telah memberikan rahmat dan karunia-nya, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik. Tugas Akhir ini ditujukan untuk memenuhi salah satu syarat kelulusan program studi Teknik Informatika di STMIK Palangkaraya.

Dalam proses penulisan tugas akhir ini penulis menyadari bahwa ada begitu banyak pihak yang telah memberikan perhatian dan bantuan dengan caranya masing masing sehingga tugas akhir ini dapat diselesaikan. Oleh karena itu penulis ingin mengucapkan terima kasih antara lain kepada :

1. Suparno, M.Kom selaku Ketua Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya.
2. Hotmian Sitohang, M.Kom selaku dosen pembimbing I yang telah memberikan bimbingan, dukungan, motivasi dan waktu untuk membimbing penulis sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik;
3. Elok Faiqotul Himmah, S.Si., M.Sc selaku dosen pembimbing II yang telah memberikan waktu-nya untuk penulis dalam memberikan saran untuk penulisan tugas akhir ini demi hasil yang baik untuk tugas akhir ini;
4. Pihak perpustakaan STMIK Palangkaraya yang memperbolehkan penulis melakukan penelitian di Perpustakaan;

Semoga tugas akhir ini dapat bermanfaat bagi penulis sendiri dan pembaca.

Palangka Raya, juli 2021

Penulis

DAFTAR ISI

LEMBAR PERNYATAAN	Error! Bookmark not defined.
PERSETUJUAN.....	Error! Bookmark not defined.
PENGESAHAN	iii
MOTTO DAN PERSEMBAHAN.....	iv
INTISARI.....	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat.....	3
1.5 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	7
2.1 Tinjauan Pustaka	7
2.1.1 Kajian Penelitian Yang Relevan	7
2.2 Kajian Teori.....	10
2.2.1 Perpustakaan	10
2.2.2 <i>Data Mining</i>	11
2.2.3 <i>Association Rules</i>	12
2.2.4 Algoritma Apriori.....	14
2.2.5 Basis Data (<i>Database</i>).....	16
2.2.6 <i>Database Management System (DBMS)</i>	17
2.2.7 MySQL.....	18
2.2.8 Java.....	19
2.2.9 <i>Netbeans IDE</i>	20
2.2.10 XAMPP	21
2.2.11 <i>Unified Modeling Language (UML)</i>	21
BAB III METODE PENELITIAN	26
3.1 Lokasi Penelitian	26

3.2	Teknik Pengumpulan Data	26
3.3	Analisis	27
3.3.1	Analisis Proses	27
3.3.2	Analisis Kelemahan Sistem.....	30
3.3.3	Analisis Kebutuhan	33
3.3.4	Analisis Kelayakan Sistem.....	35
3.4	Desain Sistem	36
3.4.2	Desain Proses	36
3.4.3	Desain Perangkat Lunak	48
3.4.4	Desain Basis Data	51
BAB IV HASIL DAN PEMBAHASAN.....		55
4.1	Hasil.....	55
4.1.1	Implementasi.....	55
4.1.2	Pengujian.....	61
4.2	Pembahasan	70
4.2.1	Pembahasan Listing Program.....	70
4.4.2	Pembahasan <i>Interface</i>	75
BAB V KESIMPULAN DAN SARAN		81
DAFTAR PUSTAKA		
LAMPIRAN		

DAFTAR TABEL

Tabel 1. Kajian Penelitian Yang Relevan	8
Tabel 2. Simbol- Simbol Use Case Diagram	22
Tabel 3. Simbol-Simbol Activity Diagram	24
Tabel 4. Simbol Simbol Sequence Diagram	25
Tabel 5. perbandingan kinerja.....	31
Tabel 6. Perbandingan informasi	31
Tabel 7. Perbandingan Ekonomi	32
Tabel 8. Perbandingan Pengendalian	32
Tabel 9. Perbandingan Efisiensi.....	33
Tabel 10. Perbandingan Layanan	33
Tabel 11. Spesifikasi Perangkat Keras.....	34
Tabel 12. Perangkat Lunak	34
Tabel 13. Activity Diagram Menu Home	39
Tabel 14. Activity Diagram Menu Buku.....	40
Tabel 15. Activity Diagram Menu Transaksi.....	41
Tabel 16. Activity Diagram Menu Tambah Transaksi.....	41
Tabel 17. Activity Diagram Algoritma Apriori	42
Tabel 18. Activity Diagram Menu Tentang Aplikasi.....	43
Tabel 19. Sturktur Tabel Buku.....	52
Tabel 20. Struktur Tabel Transaksi.....	52
Tabel 21. Struktur Tabel Kategori	53
Tabel 22. Struktur Tabel Nomor Transaksi.....	54
Tabel 16. Listing Program Analisis	63
Tabel 23. Pengujian Basis Path.....	66
Tabel 24. Uji black box.....	66
Table 25. Perbandingan hasil pengujian	68

DAFTAR GAMBAR

Gambar 1. Rapid Application Development (RAD)	29
Gambar 2. <i>Use Case Diagram</i> Aplikasi.....	37
Gambar 3. <i>Sequence Diagram</i> menu <i>home</i>	44
Gambar 5. <i>Squence Diagram</i> Menu Transaksi	45
Gambar 6. <i>Sequence Diagram</i> Apriori.....	46
Gambar 8. <i>Class Diagram</i>	48
Gambar 9. Tampilan Menu Utama Aplikasi	48
Gambar 10. Tampilam Menu Daftar Buku	49
Gambar 11. Menu Tambah Transaksi Peminjaman Buku	50
Gambar 12. Tampilam Menu Daftar Transaksi	50
Gambar 13. Tampilam Menu Analisis	51
Gambar 14. Menu Tentang Aplikasi	51
Gambar 15. Membuatan <i>Project</i>	56
Gambar 16. Desain <i>Jframe Form</i> Menu <i>Home</i>	57
Gambar 17. Membuat Desain Aplikasi	57
Gambar 18. Menambahkan <i>Icon</i> Pada Menu Bar	58
Gambar 19. Menambahkan <i>Coding</i> Pada Fitur Menu	59
Gambar 20. <i>Connect Database</i>	60
Gambar 21. Menambahkan MySQL JDBC Driver	61
Gambar 22. Diagram Alir (<i>Flowchart</i>) Analisis	62
Gambar 23. Grafik Alir (<i>Flowgraph</i>) Analisis	62
Gambar 24. Listing Program Menu Tambah Transaksi.....	72
Gambar 25. <i>Listing Program</i> Menu Tambah Buku	73
Gambar 26. <i>Listing Program</i> Analisis Algoritma Apriori	75
Gambar 27. Menu Utama Aplikasi	76
Gambar 28. Menu Daftar Buku.....	77
Gambar 29. Menu Tambah Daftar Buku.....	77
Gambar 30. Menu Daftar Transaksi.....	78
Gambar 31. Menu Tambah Transaksi Peminjaman	79
Gambar 32. Menu Analisis	79
Gambar 33. Menu Tentang Aplikasi	80

DAFTAR LAMPIRAN

- Lampiran 1. Surat tugas pembimbing Tugas Akhir
- Lampiran 2. Lembar konsultasi bimbingan Tugas Akhir
- Lampiran 3. Surat izin penelitian
- Lampiran 4. Lembar Wawancara
- Lampiran 5. Dokumentasi wawancara
- Lampiran 6. Surat Tugas Penguji Tugas Akhir
- Lampiran 7. Berita Acara ujian Tugas Akhir
- Lampiran 8. Listing Program
- Lampiran 9. Hasil Pengujian Black Box

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi yang semakin pesat memberikan dampak yang signifikan dalam berbagai bidang, pemanfaatan teknologi informasi dalam menganalisa suatu data sangat membantu manusia karena kemampuan komputer dalam mengingat dan menyimpan informasi dapat dimanfaatkan tanpa harus bergantung kepada hambatan-hambatan seperti yang dimiliki pada manusia. Dengan menyimpan informasi dan sehimpunan aturan penalaran yang memadai memungkinkan komputer memberikan kesimpulan atau mengambil keputusan dengan sangat baik.

Perpustakaan merupakan tempat yang menyediakan begitu banyak buku dan digunakan untuk tempat membaca bahkan meminjam buku. Namun dalam kenyataannya peminjam sering kali juga kebingungan dalam memilih buku, setiap mahasiswa diperbolehkan meminjam buku lebih dari satu dalam setiap transaksi peminjaman, namun tidak sedikit mahasiswa yang meminjam buku hanya satu, padahal ada kemungkinan buku yang dipinjam mempunyai pembahasan yang berlanjut di buku yang lainnya. Sehingga perlu dibuat suatu sistem aplikasi yang dapat melakukan analisis agar dapat merekomendasikan buku-buku untuk dipinjam dan untuk melakukan hal tersebut diperlukan suatu penerapan algoritma ke dalam sebuah sistem aplikasi.

Association rule mining adalah salah satu metode *data mining* yang dapat mengidentifikasi hubungan kesamaan antar item. Dalam metode ini dibutuhkan algoritma untuk mencari kandidat aturan asosiasi. Salah satu algoritma yang biasa dipakai adalah algoritma apriori. Kelebihan *Association rule* dengan apriori ini adalah lebih sederhana dan dapat menangani data yang besar. Sedangkan algoritma lainnya memiliki kelemahan dalam penggunaan memori saat jumlah data besar, tentunya berpengaruh terhadap banyaknya item yang diproses. Penting tidaknya aturan asosiasi dapat diketahui dengan 2 parameter, minimum *support* (prosentase kombinasi item dalam database) dan minimum *confidence* (kuatnya hubungan antar item dalam aturan asosiatif), keduanya ditentukan oleh user. Untuk menganalisis kebiasaan peminjam tersebut dapat digunakan analisis keranjang pasar (*market basket analysis*) yaitu suatu metode yang dapat digunakan untuk menganalisis perilaku pola belanja konsumen. Analisis ini bertujuan menemukan buku - buku yang sering dipinjam bersamaan dari data transaksi.

Berdasarkan masalah yang telah diuraikan di atas, maka peneliti ingin mengangkat judul “Penerapan Algoritma Apriori Untuk Pencarian Pola Peminjaman Buku Di Perpustakaan STMIK Palangkaraya” " sebagai bahasan pada penulisan judul tugas akhir ini dan dengan mengimplementasikan algoritma apriori ke dalam suatu sistem rekomendasi buku dirasa dapat diterapkan di sebuah sistem aplikasi berbasis JAVA sesuai dengan yang dibutuhkan.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang dikemukakan diatas, maka rumusan masalahnya adalah bagaimana mengimplementasikan *metode association rule data mining* menggunakan algoritma apriori untuk menemukan pola peminjaman buku ke dalam aplikasi ?

1.3 Batasan Masalah

Adapun yang menjadi pembatasan masalah adalah sebagai berikut guna menghindari meluasnya pembahasan diluar judul penulisan, yaitu

- a. Data yang digunakan adalah sampel data transaksi peminjaman buku, dan data buku di Perpustakaan STMIK Palangkaraya.
- b. Aplikasi digunakan untuk menrapkan algoritma apriori ke dalam aplikasi yang berbasis JAVA dalam melakukan analisis pola peminjaman buku.
- c. Bahasa pemrograman dan *database* yang digunakan adalah JAVA dan *database* MySQL.

1.4 Tujuan dan Manfaat

1. Tujuan

Adapun tujuan dari penelitian ini adalah mengimplementasikan perhitungan algoritma apriori menggunakan *association rule data mining* bagaimana bekerja dalam sebuah aplikasi yang berbasis JAVA untuk melakukan analisis, agar dapat menghasilkan *rule* atau aturan peminjan dari sampel data transaksi peminjaman buku dan memberikan rekomendasi buku dari hasil analisis tersebut.

2. Manfaat

a. Bagi Penulis

Manfaat yang diperoleh penulis adalah dapat mengimplementasikan ilmu yang telah dipelajari selama kuliah dan meningkatkan kemampuan serta menambahkan pengetahuan khususnya dalam mengimplementasikan metode *association rule data mining* menggunakan algoritma apriori untuk menemukan pola peminjaman buku.

b. Bagi STMIK Palangkaraya

Manfaat yang diberikan kepada kampus STMIK Palangkaraya adalah sebagai penambahan referensi serta dokumen akademik pustaka di perpustakaan STMIK Palangkaraya serta evaluasi bagi pengembangan peningkatan mutu pendidikan STMIK Palangkaraya di masa yang akan datang.

c. Bagi Pengguna

Manfaat yang didapatkan oleh pengguna adalah dapat lebih mengetahui buku apa saja yang sering di pinjam secara bersamaan, serta dapat membantu dalam mendapatkan rekomendasi buku dari hasil analisis aplikais buku yang cenderung lebih sering dipinjam.

1.5 Sistematika Penulisan

Adapun sistematika penulisan dalam penulisan tugas akhir ini terdiri dari beberapa bab dan masing masing bab membahas dan menguraikan pokok permasalahan yang berbeda sebagai gambaran di sini penulis menyertakan garis - garis besar yaitu.

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang, rumusan masalah, tujuan dan manfaat penelitian.

BAB II LANDASAN TEORI

Bab ini membahas mengenai teori teori *data mining* dan algoritma apriori serta teori-teori yang mendukung penelitian, yang menjadi dasar bagi pemecahan masalah pada proses penelitian dan perancangan.

BAB III ANALISIS DAN DESAIN SISTEM

Bab ini membahas tentang analisa dan perancangan sistem. Analisa perancangan sistem meliputi deskripsi aplikasi yang akan dibuat, analisa kelemahan sistem maupun kebutuhan sistem. Sedangkan perancangan sistem meliputi pembuatan diagram UML (*Unified Modelling Language*) yang terdiri atas *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan perancangan desain antar muka.

BAB IV IMPLEMENTASI

Bab ini berisi tentang implementasi dan pengujian sistem aplikasi yang akan dibuat menggunakan metode *association rule* dengan algoritma apriori, lalu penjelasan – penjelasan dari hasil implementasi dan pengujian yang dilakukan berdasarkan parameter – parameter pengujian yang meliputi *minimum support*, dan *minimum confidence*.

BAB V KESIMPULAN DAN SARAN

Bab ini adalah bab penutup, dimana berisi tentang kesimpulan serta analisa dari aplikasi yang telah dibuat, juga berisi tentang saran dari pembuat aplikasi, dimana nantinya bisa menjadi acuan untuk pengembangan.

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Kajian Penelitian Yang Relevan

Pada penelitian ini diperlukan dukungan hasil-hasil penelitian yang telah ada sebelumnya dimana penelitian tersebut berkaitan dengan penelitian yang serupa. Berikut ini hasil-hasil penelitian yang relevan dan perbandingan penelitian yang telah ada sebelumnya yang serupa dengan penelitian yang sedang dilakukan dan disajikan dalam bentuk Tabel 1

Tabel 1. Kajian Penelitian Yang Relevan

No	Penulis/Tahun	Topik Penelitian	Metode	Hasil	Perbedaan
1	M. Afdal, Muhammad Rosadi/2019	Penerapan Association Rule Mining Untuk Analisis Penempatan Tata Letak Buku Di Perpustakaan Menggunakan Algoritma Apriori	Apriori	Pada penelitian ini dianalisa dengan algoritma apriori dengan hasil analisa pada semua data dari tahun 2016-2018 didapatkan buku dengan kategori buku agama, teknologi, filsafat dan ilmu sosial paling sering dipinjam secara bersamaan.	Perbedaan terdapat pada implementasi algoritma apriori yang digunakan dimana dalam penelitian ini hanya menampilkan bagaimana tahap tahap dalam algoritma apriori namun tidak ada implementasinya ke kalam sebuah program aplikasi.
2	Srikanti, Yansi, Permana, Salisah/2018	Penerapan Algoritma Apriori Untuk Mencari Aturan Asosiasi Pada Data Peminjaman Buku Di Perpustakaan	Apriori	Hasil penelitian bahwa berhasil menemukan pola peminjaman, Oleh sebab itu disarankan pada pihak perpustakaan FST UIN SUSKA Riau untuk meletakkan buku-buku tersebut secara berdekatan.	Perbedaan terdapat pada implementasi algoritma apriori yang digunakan yang dimana dalam penelitian ini hanya menampilkan bagaimana tahap tahap dalam algoritma apriori namun tidak ada implementasinya ke dalam sebuah program aplikasi.

No	Penulis/Tahun	Topik Penelitian	Metode	Hasil	Perbedaan
3	Kuncoro,Widya Sari dan Wibawa /2017	Pemanfaatan Algoritma Apriori untuk Perancangan Ulang Tata Letak Barang di Toko Busana	Apriori	Dari hasil pengujian, sistem telah terbukti berhasil menerapkan algoritma Apriori untuk mendapatkan pola konsumen toko busana muslimah.	Implementasi algoritma apriori dalam penelitian ini menggunakan aplikasi yang berbasis web. sedangkan pada penelitian yang saya lakukan ini akan mengimplementasikannya ke dalam aplikasi yang berbasis java.
4	Sastie , Suparni , Pohan/2020	Analisa Algoritma Apriori Pada Pola Peminjaman Buku di Perpustakaan ITB Ahmad Dahlan	Apriori	Penerapan algoritma apriori dapat digunakan untuk mengetahui buku apa saja yang sering muncul di dalam proses peminjaman buku, memberikan rekomendasi buku-buku terkait dengan buku yg sering dipinjam.	Perbedaan dalam penelitian ini yaitu dimana dalam penerapan algoritma apriorinya menggunakan aplikasi Tanagra yang terapat pada exel yang digunakan dalam melakukan penelitian tersebut.
5	Budiyati, Hurniningsih , Lusita/2020	Implementasi Metode Algoritma Apriori Untuk Penempatan Buku Pada Rak Perpustakaan Stmik Jakarta Sti&K	Apriori	Dengan sistem yang dibuat menggunakan algoritma apriori ini dapat menghasilkan association rule dengan menghitung nilai support dan confidence berdasarkan data transaksi peminjaman buku berbasis web.	Perbedaan terdapat pada penerapan algoritma apriori yang dimanana pada penelitian ini sistem yang dibuat berbasis java.

2.2 Kajian Teori

2.2.1 Perpustakaan

Perpustakaan berasal dari kata Sansekerta, pustaka yang artinya kitab, buku. Dalam bahasa Inggris, istilah *library* berasal dari kata Latin *liber* atau *libri* yang artinya buku. Dari kata Latin tersebut, terbentuklah istilah *librarius* yang artinya tentang buku. Dalam bahasa Belanda *bibliotheek*, Jerman *bibliothek*, Perancis *bibliothèque*, Spanyol *bibliotheca*, dan Portugal *bibliotheca*. Semua istilah itu berasal dari bahasa Yunani *biblia* yang artinya tentang buku, kitab.

Perpustakaan berasal dari kata *liber = libri* yang artinya “pustaka” atau “kitab”. Menurut Sulisty Basuki, perpustakaan adalah ruangan, bagian dari sebuah gedung ataupun gedung itu sendiri yang digunakan untuk menyimpan buku atau terbitan lainnya yang biasanya disimpan menurut tata susunan tertentu untuk digunakan sebagai bahan bacaan bukan untuk dijual (Hartono, 2016:26).

Dari berbagai sumber tersebut, dapat disimpulkan bahwa perpustakaan ialah sebuah ruangan atau gedung yang digunakan untuk menyimpan koleksi buku atau koleksi lain bagi pengunjung untuk membaca, meminjam, ataupun merujuk, namun bukan untuk dijual. Pengertian ini menunjukkan bahwa koleksi perpustakaan digunakan untuk mendayagunakan koleksinya untuk kepentingan para pengunjung, bukan untuk kegiatan jual beli.

2.2.2 *Data Mining*

Data mining merupakan suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan didalam basis data. Data mining juga merupakan proses yang menggunakan matematika, teknik statistik, kecerdasan buatan, dan *machine learning* untuk mengidentifikasi dan mengekstraksi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data besar (Pane, 2013).

Data mining atau sering disebut sebagai *knowledge discovery in database* (KDD) adalah sebagai sebuah proses untuk menemukan hubungan, pola, dan trend baru yang bermakna dengan menyaring data yang sangat besar, yang tersimpan dalam penyimpanan, menggunakan teknik pengenalan pola seperti teknik statistik dan matematika (Kusrini, 2017).

Ada juga yang berpendapat *Data Mining* (DM) adalah inti dari proses *knowledge discovery of database*, melibatkan kesimpulan algoritma yang mencari data, mengembangkan model dan menemukan pola-pola yang sebelumnya tidak diketahui.

Dari definisi-definisi yang telah disampaikan, hal penting yang terkait dengan data mining adalah

1. *Data mining* merupakan suatu proses otomatis terhadap data yang sudah ada.

2. Data yang akan diproses berupa data yang sangat besar.
3. Tujuan *data mining* adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

2.2.3 *Association Rules*

Analisis asosiasi atau *association rule mining* merupakan teknik data mining yang digunakan untuk menemukan pola yang menggambarkan kekuatan hubungan fitur dalam data. Pola yang ditemukan biasanya merepresentasikan bentuk aturan implikasi atau subset fitur. Tujuannya adalah menemukan pola yang menarik dengan cara yang efisien (Irawan, 2017 : 73).

Menurut Amirudin et al, dalam Tampubolon, (2013 : 97), *association rule* adalah teknik data mining untuk menemukan aturan asosiasi antara kombinasi item. Contoh dari aturan asosiasi dari analisa pembelian di suatu pasar swalayan adalah dapat diketahui berapa besar kemungkinan seseorang membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut pemilik pasar swalayan dapat mengatur penempatan barangnya atau merancang kampanye pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu.

Menurut Susanto dan Suryadi dalam Tampubolon, (2013 : 97), aturan asosiasi akan menggunakan data latihan, sesuai dengan pengertian data mining, untuk menghasilkan pengetahuan. Pengetahuan untuk mengetahui item-item belanja yang sering

dibeli secara bersamaan dalam suatu waktu. Aturan asosiasi yang berbentuk “*if...then...*” atau “jika...maka...” merupakan pengetahuan yang dihasilkan dari fungsi Aturan Asosiasi.

Analisis asosiasi didefinisikan suatu proses untuk menemukan semua aturan asosiasi yang memenuhi syarat minimum untuk *support (minimum support)* dan syarat minimum untuk *confidence (minimum confidence)*. Metodologi dasar analisis asosiasi terbagi menjadi dua tahap :

1. Analisis Pola Frekuensi Tinggi Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support* sebuah item diperoleh dengan rumus berikut :

$$\text{Support } A = \frac{\text{Jumlah transaksi mengandung } A}{\text{Jumlah transaksi}} \times 100\% \quad (1)$$

Sementara itu, nilai *support* dari 2 item diperoleh dari rumus 2 berikut :

$$\text{Support } (A, B) = \frac{\Sigma \text{transaksi mengandung } A \text{ dan } B}{\Sigma \text{Jumlah transaksi}} \times 100\% \quad (2)$$

Keterangan :

- a. Support A,B adalah nilai penunjang (persentase) kombinasi dari dua item yaitu item A dan item B dalam *database*.
- b. Σ transaksi untuk A dan B adalah jumlah kemunculan kombinasi item A dan B dalam keseluruhan transaksi.
- c. Σ transaksi adalah jumlah total transaksi yang ada dalam *database*.

2. Pembentukan aturan asosiasi, setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat minimum untuk *confidence* dengan menghitung nilai confidence aturan asosiatif $A \rightarrow B$. Nilai *confidence* dari aturan $A \rightarrow B$ diperoleh dari rumus sebagai berikut:

$$\text{confidence } P(B|A) = \frac{\sum \text{transaksi mengandung } A \text{ dan } B}{\sum \text{transaksi mengandung } A} \times 100 \% \quad (3)$$

Keterangan :

- a. *Confidence* $P(A|B)$ adalah nilai kepastian kuatnya hubungan antar item A dan item B dalam aturan asosiatif atau berapa kali item A muncul bersamaan dengan item B.
- b. Σ transaksi untuk A dan B adalah jumlah kemunculan kombinasi item A dan B dalam keseluruhan transaksi.
- c. Σ transaksi untuk A adalah jumlah total transaksi item A dalam keseluruhan transaksi.

2.2.4 Algoritma Apriori

Algoritma apriori merupakan salah satu algoritma klasik *data mining*. Algoritma apriori digunakan agar komputer dapat mempelajari aturan asosiasi, mencari pola hubungan antar satu atau lebih item dalam suatu *dataset*.

Algoritma Apriori adalah salah satu algoritma pada bidang data mining untuk penggalian aturan asosiasi atau yang lebih

dikenal dengan istilah *association rule mining* (ARM). Algoritma Apriori merupakan pendekatan iteratif dimana k -itemset digunakan untuk mengeksplorasi $(k + 1)$ -itemset.

Algoritma apriori merupakan algoritma yang paling terkenal untuk menemukan pola frekuensi tinggi. algoritma apriori dibagi menjadi beberapa tahap yang disebut narasi (Buulolo, 2013) yaitu:

1. Pembentukan kandidat *itemset*, kandidat k -itemset dibentuk dari kombinasi $(k-1)$ -itemset yang didapat dari iterasi sebelumnya. Satu cara dari algoritma apriori adalah adanya pemangkasan kandidat k -itemset yang subset-nya yang berisi $k-1$ item tidak termasuk dalam pola frekuensi tinggi dengan panjang $k-1$.
2. Perhitungan support dari tiap kandidat k -itemset *Support* dari setiap kandidat k -itemset didapat dengan men-*scan database* untuk menghitung jumlah transaksi yang memuat semua item di dalam kandidat k -itemset tersebut. Ini adalah juga termasuk ciri dari algoritma apriori dimana diperlukan perhitungan dengan *scan* seluruh *database* sebanyak k -itemset terpanjang.
3. Tetapkan pola frekuensi tinggi Pola frekuensi tinggi yang memuat k -itemset ditetapkan dari kandidat k -itemset yang *supportnya* lebih besar dari *minimum support*.

4. Bila tidak didapat pola frekuensi tinggi baru maka seluruh proses dihentikan, bila tidak maka k ditambah satu dan kembali ke bagian 1.

Algoritma apriori adalah satu algoritma dasar yang diusulkan oleh Agrawal dan Srikant pada tahun 1994 untuk menemukan *frequent itemsets* pada aturan asosiasi Boolean. Ide utama pada algoritma apriori adalah : pertama, mencari *frequent itemset* (himpunan item-item yang memenuhi *minimum support*) dari basis data transaksi. Kedua, menghilangkan *itemset* dengan frekuensi yang rendah berdasarkan level *minimum support* yang telah ditentukan sebelumnya. Selanjutnya membangun aturan asosiasi dari itemset yang memenuhi nilai *minimum confidence* dalam basis data (Agrawal dan Srikant, 2016 : 122).

2.2.5 Basis Data (*Database*)

Basis Data adalah kumpulan data yang saling berhubungan secara logikal serta deskripsi dari data tersebut, yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Basis Data adalah sebuah penyimpanan data yang besar yang bisa digunakan oleh banyak pengguna dan departemen. Semua data terintegrasi dengan jumlah duplikasi yang minimum. Menurut Ladjamudin (2013), *Database* adalah sekumpulan data store (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk*, *optical disk*, *magnetic drum*, atau media penyimpanan sekunder lainnya.

Basis data adalah dua atau lebih simpanan data dengan elemen-elemen data penghubung, yang dapat diakses lebih dari satu cara. Basis data dinyatakan dengan tehnik-tehnik formal dan manajemen basis data (Abdillah, 2012: 1).

Dapat disimpulkan basis data adalah penyimpanan data yang terstruktur, terintegrasi dan saling berkaitan dengan elemen-elemen penghubungnya dan dapat di akses dengan berbagai cara, oleh karena itu basis data juga bisa didefinisikan sebagai kumpulan yang menggambarkan sendiri dari catatan yang terintegrasi dan penggambaran dari data dikenal sebagai sistem katalog (atau kamus data atau metadata). Definisi data disini dibedakan dari program aplikasi, yang umumnya sama dengan pendekatan pengembangan modern perangkat lunak, dimana definisi internal dan eksternal dari sebuah objek dipisahkan. Salah satu keuntungan dari pendekatan tersebut adalah abstraksi data dimana kita dapat mengubah definisi internal dari sebuah objek tanpa mempengaruhi pengguna dari objek jika definisi eksternal objek tersebut tidak berubah.

2.2.6 Database Management System (DBMS)

Database Management System adalah sistem perangkat lunak yang memungkinkan pengguna dapat mendefinisikan, membuat, merawat, dan mengatur akses ke basis data. DBMS dapat juga diartikan sebagai program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memodifikasi dan

memperoleh data/informasi dengan praktis dan efisien. Biasanya DBMS memungkinkan pengguna untuk mendefinisikan *Database* melalui sebuah *Data Definition Language* (DDL), menspesifikasikan tipe data, struktur dan batasan pada data yang disimpan pada *Database*. Kemudian juga memungkinkan *insert*, *update*, *delete*, dan mengambil data dari *database* melalui *Data Manipulation Language* (DML), mempunyai pusat penyimpanan untuk semua data dan deskripsi data memungkinkan DML untuk menyediakan fasilitas umum untuk data tersebut yang umumnya disebut bahasa query (Connolly & Begg, 2015).

2.2.7 MySQL

Menurut Arief (2012d:152) MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolahan datanya.

MySQL adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada. Seperti yang sudah disinggung di atas, MySQL masuk ke dalam jenis RDBMS (*Relational Database Management*

System). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. Contohnya di dalam MySQL sebuah *database* terdapat satu atau beberapa tabel.

SQL sendiri merupakan suatu bahasa yang dipakai di dalam pengambilan data pada *relational database* atau *database* yang terstruktur. Jadi MySQL adalah *database management system* yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server*.

2.2.8 Java

Java merupakan bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya di suatu perusahaan perangkat lunak yang bernama *Sun Microsystems*, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama “*Oak*”, namun pada tahun 1995 diganti namanya menjadi “Java”.

Menurut Suyanto (2015:2), menyatakan bahwa: Java diciptakan oleh suatu tim yang dipimpin oleh Patrick Naughton dan James Gosling dalam suatu proyek dari *sun microsystem* yang memiliki kode ` dengan tujuan untuk menghasilkan bahasa komputer sederhana yang dapat dijalankan di peralatan sederhana dengan tidak terikat pada arsitektur tertentu, mulanya disebut *oak*, tetapi karena *oak* sendiri merupakan nama dari bahasa pemrograman komputer yang sudah ada, maka *sun* mengubahnya

menjadi java.sun kemudian meluncurkan *browser* dari java yang disebut *hot java* yang mampu menjalankan applet. Setelah itu teknologi java diadopsi oleh Netscape yang memungkinkan program java dijalankan di *browser* netscape yang kemudian diikuti *Internet Explore*. Karena keunikan dan kelebihananya, teknologi java mulai menarik banyak vendor seperti IBM, Symantec, Inprise, dll. Sun merilis versi awal java secara resmi pada awal tahun 1996 yang kemudian terus berkembang hingga muncul jdk 1.1 kemudian jdk 1.2 yang mulai disebut sebagai versi java2 karena banyak mengandung peningkatan dan perbaikan. Perubahan utama adalah swing yang merupakan teknologi GUI (*Graphical User Interface*) yang mampu menghasilkan windows yang portable. Dan pada tahun 1998-1999 lahirlah teknologi J2EE (*Java 2 Enterprise Edition*).

2.2.9 Netbeans IDE

Netbeans merupakan sebuah aplikasi *Integrated Development Environment* (IDE) berbasis Java dari *Sun Microsystems* yang berjalan di atas swing. Swing sendiri adalah sebuah teknologi Java untuk pengembangan aplikasi dekstop yang dapat berjalan di berbagai macam platform seperti windows, linux, Mac OS X dan juga Solaris. Untuk lebih lengkapnya lagi simaklah Pengertian Netbeans.

Netbeans menurut Maya adalah salah satu aplikasi IDE (*Integrated Development Environment*) yang digunakan oleh *Developer Software computer* untuk menulis, mengcompile, mencari kesalahan, dan untuk menyebarkan program (Undaksana & Kusaeri, 2018).

2.2.10 XAMPP

Menurut MADCOMS (2016) “Xampp adalah sebuah paket kumpulan *software* yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain.” Xampp berfungsi untuk memudahkan instalasi lingkungan PHP, di mana biasanya lingkungan pengembangan web memerlukan PHP, Apache, MySQL dan PhpMyAdmin.

Xampp berfungsi membantu user untuk melakukan instalasi dan konfigurasi *web server*, yaitu *apache*, *Database Management System*(DBMS), MySQL, dan *web Programing*, yaitu PHP (Afandi & Yulianis, 2018).

2.2.11 Unified Modeling Language (UML)

UML adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada *system* (Mulyani, 2016).

UML adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi dan mendokumentasikan *artifac* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan

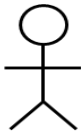
perangkat lunak, dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak seperti pada permodelan bisnis dan sistem *non* perangkat lunak lainnya.

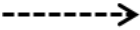
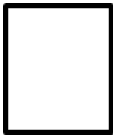
Untuk membuat suatu model, UML memiliki *diagram grafis* yang diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analisa atau rekayasa. *Diagram grafis* tersebut antara lain :

1) *Use Case Diagram*

Use Case Diagram menjelaskan manfaat sistem menurut pandangan orang yang berbeda di luar sistem (*actor*). *Diagram* ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar. Simbol-simbol data *entity relationship* dapat dilihat pada tabel 2

Tabel 2. Simbol- Simbol *Use Case Diagram*

No	Gambar/symbol	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>usecase</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).

No	Gambar/symbol	Nama	Keterangan
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>usecase</i> sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasikan bahwa <i>usecase</i> target memperluas perilaku dari <i>usecase</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor .
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-

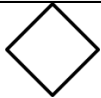
No	Gambar/symbol	Nama	Keterangan
			elemennya (sinergi).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2) Activity Diagram

Activity Diagram memodelkan alur kerja sebuah proses bisnis dan urutan aktivitas dalam suatu proses. *Diagram* ini sangat mirip dengan *flowchart* karena dapat memodelkan sebuah alur kerja satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). Seringkali bermanfaat bila membuat sebuah proses *activity diagram* terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. Simbol-simbol dari *activity diagram* dapat dilihat pada Tabel .3

Tabel 3. Simbol-Simbol *Activity Diagram*

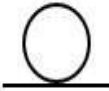
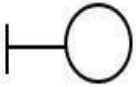
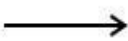
No	Relasi	Fungsi	Notasi
1	Activity	Memperlihatkan bagaimana masing-masing kelas antar muka saling berinteraksi satu sama lain	
2	Initial Mode	Bagaimana objek dibentuk atau diawali	

No	Relasi	Fungsi	Notasi
3	<i>Decision</i>	Pilihan untuk pengambilan keputusan	
4	<i>Activity Final Mode</i>	Bagaimana objek dibentuk atau diakhiri	

3) Sequence Diagram

Sequence Diagram adalah gabungan dari *class diagram* dan *object* yang memiliki suatu gambaran model statis, diagram ini menjelaskan bagaimana suatu operasi dilakukan, pesan apa yang akan dikirim, dan kapan pelaksanaannya. Diagram ini memiliki objek objek yang berkaitan dengan proses berjalannya operasi yang diurutkan dari kiri ke *simbol sequence diagram* :

Tabel 4. Simbol Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari form
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

BAB III

METODE PENELITIAN

3.1 Lokasi Penelitian

Dalam pembuatan penelitian ini penulis melakukan penelitian di perpustakaan STMIK Palangkaraya yang terletak di Jl. G. Obos No.114, Menteng, Kec. Jekan Raya, Kota Palangka Raya, Kalimantan Tengah 74874.

3.2 Teknik Pengumpulan Data

Dalam melakukan penelitian untuk menyusun proposal Tugas Akhir ini penulis menggunakan beberapa tahapan atau metode penelitian, yaitu :

a. Tahapan Pengumpulan Data

1. Metode Pengamatan (*Observation*)

Metode pengamatan atau observasi adalah suatu metode pengumpulan data dengan cara penulis mengamati langsung ke lapangan guna mendapatkan data dan informasi yang dibutuhkan penulis melakukan pengamatan langsung di perpustakaan STMIK Palangkaraya dan melakukan pencatatan secara manual guna mendapatkan data daftar transaksi peminjaman buku dan data buku.

2. Wawancara

Wawancara merupakan cara mengumpulkan data dengan cara tanya jawab yang dikerjakan secara langsung di tempat atau lokasi penelitian dan berlandaskan pada tujuan yang dilakukan

oleh penulis. Dalam penelitian ini pengumpulan data didapatkan dari keterangan lisan melalui tanya jawab dengan pihak-pihak perpustakaan yang dapat memberikan informasi ataupun data yang diperlukan.

3. Studi literatur

Studi literatur adalah mencari dan mempelajari referensi teori yang relevan dengan kasus atau permasalahan yang ditemukan. Referensi untuk penelitian ini dapat dicari dari buku, jurnal, artikel laporan penelitian, dan situs-situs di internet.

3.3 Analisis

Ditahap ini akan dijelaskan beberapa faktor yang diperlukan oleh sistem termasuk perangkat keras yang digunakan sampai diperlukannya perancangan perangkat lunak sehingga perangkat lunak tersebut sesuai dengan maksud dan tujuan dibangunnya sistem tersebut. Ada beberapa tahap yang akan dilakukan dalam analisis ini, berikut adalah tahap analisis system yang dilakukan oleh penulis :

3.3.1 Analisis Proses

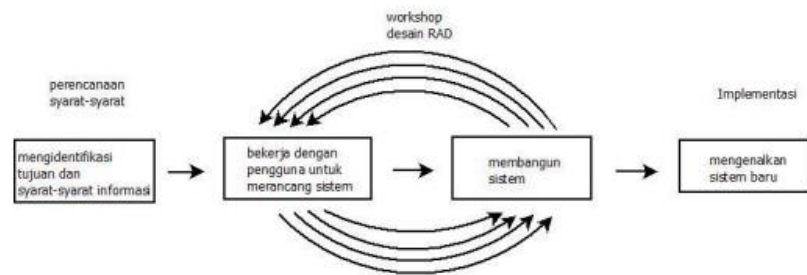
Disini penulis memilih metode RAD sebagai metode pengembangan sistem untuk kasus yang diteliti yaitu dengan alasan berikut :

1. Aplikasi yang dirancang dan dikembangkan merupakan aplikasi yang sederhana dan tidak memerlukan waktu yang lama.

2. Dengan menggunakan metode RAD, akan dicapai suatu sistem yang berhasil dibuat dalam periode waktu yang relatif pendek dengan catatan, kebutuhan sistem telah diperoleh dan dipahami dengan baik.
3. RAD juga dapat mengatasi permasalahan ketidak jelasan tentang kebutuhan pengguna *user requirement*, terhadap apa yang harus sistem lakukan nantinya. Ketidak jelasan ini karena sulitnya bagi pengguna untuk mengemukakan apa yang diinginkan kepada penulis dan lebih mudah jika langsung berinteraksi dengan aplikasi.

a. *Rapid Application Development (RAD)*

Rapid Application Development (RAD) adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik *inkremental* (bertingkat). *Rapid Application Development (RAD)* menekankan pada siklus pembangunan pendek, singkat, dan cepat. Waktu yang singkat adalah batasan yang penting untuk model ini. *Rapid Application Development (RAD)* menggunakan metode *iteratif* (berulang) dalam mengembangkan sistem dimana *working model* (model kerja) sistem dikonstruksikan diawal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) pengguna. Model kerja digunakan hanya sesekali saja sebagai basis desain dan implementasi sistem akhir.



Gambar 1. Rapid Application Development (RAD)
(Sukamto & Shalahuddin 2018)

Metodologi penelitian ini menggunakan pendekatan *Rapid Application Development* (RAD) yaitu Analisis Persyaratan, *Design Workshop* (Pemodelan) dan Implementasi (Konstruksi). Sehingga tahapan penelitian yang dilakukan mengikuti tahapan yang ada di dalam metodologi RAD.

1. Analisis Persyaratan

Tahapan Analisis Persyaratan bertujuan untuk mengidentifikasi kebutuhan, batasan dan objektifitas dari sistem yang akan dibangun dengan mengumpulkan data dari *stakeholder*. Tahapan ini dilakukan dengan melakukan wawancara kepada pihak petugas perpustakaan STMIK Palangkaraya mengenai kebutuhan sistem yang dibutuhkan. Setelah melakukan wawancara, maka didapatkan solusi mengenai sistem yang dibutuhkan.

2. Design Workshop

Pada tahapan Pemodelan bertujuan untuk merancang semua kegiatan dalam arsitektur sistem secara keseluruhan dan meningkatkan pemahaman atas masalah berdasarkan analisis-analisis yang dilakukan. Pada Tahap ini peneliti merancang semua kegiatan yang melibatkan design abstraksi sistem perangkat lunak secara keseluruhan yang bertujuan untuk meningkatkan pemahaman atas masalah berdasarkan analisis yang dilakukan.

3. Implementasinya

Tahapan implementasi yaitu mengimplentasikan sistem dan penerapan metode dalam pemrograman terhadap hasil kebutuhan sistem dan dapat dijelaskan dalam tahap implementasi *database* dan *coding* program. Tujuan dari tahapan ini yaitu mengkonstruksi sistem dan penerapan metode dalam pemrograman terhadap hasil kebutuhan sistem dan akan dijelaskan dalam tahapan implemetasi *database*.

pemrograman, dan antarmuka. Kemudian hasil yang didapatkan yaitu main database dan kode program.

3.3.2 Analisis Kelemahan Sistem

Namun untuk menganalisa kinerja sistem yang akan dibangun terdapat kelemahan sistem. Pada penelitian ini penulis

menggunakan metode analisis *PIECES*. Berikut beberapa aspek yang akan dibahas berdasarkan *PIECES*.

a. *Performance* (Kinerja)

Penjabaran aspek kinerja sistem lama ke sistem baru akan dijelaskan pada tabel dibawah ini.

Tabel 5. perbandingan kinerja

Sistem Lama	Sistem Baru
Kinerja sistem tergolong lama dan memakan waktu yang lama dikarenakan melakukan analisis secara manual	Kinerja sistem baru lebih cepat dikarenakan seluruh proses analisis dilakukan dengan otomatis dari sistem, sehingga petugas perpustakaan hanya perlu menginputkan nilai <i>support</i> dan <i>confidence</i>

b. *Information* (Informasi)

Penjabaran aspek informasi yang didapat oleh sistem lama dan sistem baru akan dijelaskan pada tabel dibawah ini.

Tabel 6. Perbandingan informasi

Sistem Lama	Sistem Baru
Pada saat petugas memasukan data buku yang dipinjam oleh peminjam maka data akan langsung disimpan dan nanti data tersebut dapat ditampilkan kembali	Kinerja sistem yang baru dapat mengecek data transaksi bahkan dapat melakukan analisis

c. *Economy* (Ekonomi)

Ditinjau dari segi aspek ekonomi yang akan mempengaruhi penggunaan biaya dan memberikan sebuah sistem yang ekonomis, sehingga sistem baru yang akan

dibangun dinilai lebih efektif dan murah. Dapat dilihat pada tabel dibawah ini adalah perbandingan sistem lama dan sistem baru.

Tabel 7. Perbandingan Ekonomi

Sistem Lama	Sistem Baru
Pada sistem yang lama memerlukan koneksi internet dalam mengelola data	dimana pada sistem yang baru tidak membutuhkan koneksi internet

d. *Control* (pengendalian)

Adapun untuk membandingkan sistem lama dan sistem baru berdasarkan control pada segi ketepatan waktu, kemudahan akses dan ketelitian data yang diproses. Pada tabel dibawah ini adalah perbandingan dari sistem.

Tabel 8. Perbandingan Pengendalian

Sistem Lama	Sistem Baru
Kontrol sistem terlalu rumit karena seluruh proses perhitungan dilakukan secara manual	Sedangkan pada sistem yang baru semua proses analisis maupun perhitungan dilakukan secara otomatis oleh sistem

e. *Efficiency* (Efisiensi)

Dilihat dari segi keefisiensi sistem yang digunakan sistem lama dan sistem baru secara optimal. Pada tabel 9 dibawah ini adalah perbandingan dari sistem lama dan sistem baru.

Tabel 9. Perbandingan Efisiensi

Sistem Lama	Sistem Baru
Proses analisis memakan waktu yang cukup lama karena dilakukan secara manual.	Sistem yang baru tidak perlu melakukan analisis secara manual karena dapat dilakukan oleh sistem secara otomatis.

f. *Service* (Layanan)

Dilihat dari segi layanan sistem yang digunakan sistem lama dan sistem baru secara optimal. Pada tabel 10 dibawah ini adalah perbandingan dari sistem lama dan sistem baru.

Tabel 10. Perbandingan Layanan

Sistem Lama	Sistem Baru
Pada sistem yang lama dari segi layanan pengguna perpustakaan hanya diberikan sebuah sistem yang tidak dapat melakukan analisis	Pada sistem yang baru memberikan layanan yang dapat lebih banyak mempermuda petugas perpustakaan dalam melakukan analisis secara otomatis dan mudah dipahami.

3.3.3 Analisis Kebutuhan

Agar sistem baru yang diusulkan untuk dibangun dapat digunakan dalam mengatasi masalah yang ada maka diperlukan beberapa analisis kebutuhan berikut :

3.3.3.1 Kebutuhan Perangkat Keras

Dalam kebutuhan perangkat keras (*hardware*) yang digunakan oleh penulis adalah sebagai berikut :

Tabel 11. Spesifikasi Perangkat Keras

No	Nama Perangkat	Spesifikasi
1	Type	HP Pavilion
2	Processor	AMD Ryzen 5 3550H
3	RAM	8 GB
4	SSD	512 GB
5	VGA	NVIDIA GeForce GTX 1650
6	Keyboard	Standar
7	Mouse	Standar

3.3.3.2 Kebutuhan Perangkat Lunak

Dalam kebutuhan perangkat lunak (*software*) yang digunakan oleh penulis adalah sebagai berikut :

Tabel 12. Perangkat Lunak

No	Nama Perangkat	Fungsi
1.	<i>Netbeans</i>	Sebagai perangkat lunak yang digunakan untuk membuat aplikasi JAVA dengan format <i>.jar</i>
2.	<i>Java Development Kit (JDK)</i>	yang akan memberikan akses ke fungsi standar, yang menggunakan sistem pemograman java sebagai implementasi program.
3	XAMPP	Perangkat lunak ini digunakan sebagai <i>server</i> yang berdiri sendiri (<i>localhost</i>)

3.3.3.3 Kebutuhan Informasi

Pada kebutuhan informasi, penulis mengumpulkan data/atau informasi secara langsung atau bertatap muka dengan pihak bagian perpustakaan STMIK Palangkaraya, dimana data data yang diperlukan pada penelitian ini adalah data data buku yang ada di perpustakaan dan data transaksi peminjaman buku oleh mahasiswa.

3.3.4 Analisis Kelayakan Sistem

Dokumen yang dihasilkan dari tahapan tahapan sebelumnya di kumpulkan menjadi suatu proposal pendahuluan penelitian. Untuk memastikan usulan tersebut bisa diteruskan menjadi penelitian yang menguntungkan maka proposal harus dievaluasi kelayakannya dari berbagai segi kelayakan, diantaranya :

3.3.4.1 Kelayakan Teknologi

Kelayakan teknologi yang diberikan aplikasi ini adalah sebagai perangkat lunak yang dapat memberikan kemudahan untuk membantu pihak perpustakaan di STMIK Palangkaraya dalam melakukan analisis.

3.3.4.2 Kelayakan Hukum

Untuk menentukan kelayakan hukum terhadap sistem yang dirancang, bisa dilihat pada poin poin berikut :

- 1) Informasi yang disajikan sudah mendapat persetujuan dari pihak perpustakaan STMIK Palangkaraya.

- 2) Pembuatan aplikasi memiliki beberapa referensi penerapan algoritma yang didapatkan dari internet yang bersifat *open source* yang dimana aplikasi itu juga dapat dikembangkan oleh orang banyak.

3.3.4.3 Kelayakan Operasional

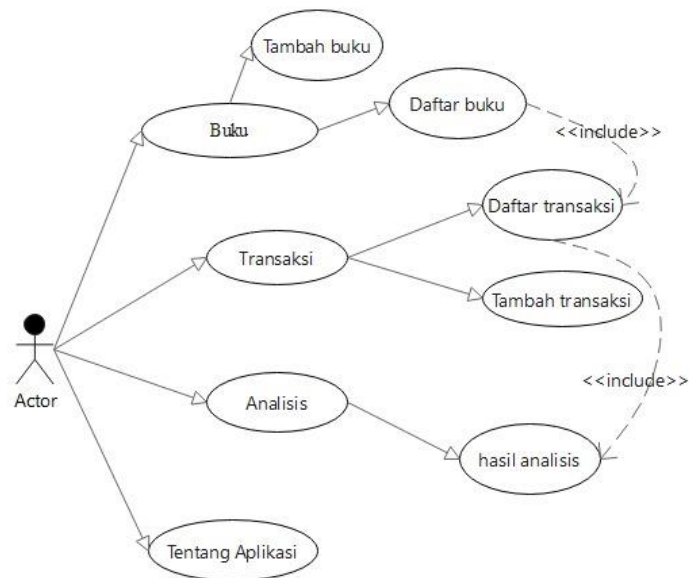
Mengenai kelayakan operasional aplikasi yang di bangun, dirancang dengan sederhana agar pengguna aplikasi dapat lebih mudah memahami pengoperasian saat menggunakan aplikasi.

3.4 Desain Sistem

3.4.2 Desain Proses

a. Use Case Diagram

Use case diagram disini merupakan gambaran dari *user* yang pada saat akan menggunakan sistem dan perilaku *user* terhadap sistem. Pada sistem ini hanya terdapat seorang user aplikasi yang dapat mengakses beberapa menu pilhan aplikasi yaitu menu buku, transaksi, analisis dan info aplikasi dan beberapa hak akses lainnya yang dapat dilihat lebih jelas pada Gambar 2.



Gambar 2. *Use Case Diagram* Aplikasi

Pada gambar 2 dijelaskan bahwa terdapat seorang *user* yang dapat melakukan hak akses untuk beberapa halaman menu yang memiliki fungsinya masing masing. Berikut ini penjelasan dari masing masing menu yang terdapat dari rancangan *use case diagram* diatas.

1. Menu buku memiliki dua pilihan sub menu yang yaitu sub menu untuk melihat daftar buku dan sub menu menambahkan buku.
2. Menu bar transaksi pada halaman menu transaksi juga terdapat dua sub menu yang dapat digunakan oleh *user* yaitu sub menu untuk menampilkan daftar transaksi dan juga sub menu untuk menambahkan transaksi peminjaman buku yang dimana saat melakukan proses

peminjaman dilakukan penginputan data mahasiswa yang meminjam buku dan judul buku.

3. Menu bar analisis pada menu ini merupakan menu untuk melakukan analisis menggunakan penerapan *associan rule* algoritma apriori yang dimana *user* harus menentukan atau menginputkan jumlah minimal *support* dan *confidence* yang diinginkan oleh *user* pada saat melakukan analisis, yang nantinya setelah selesai maka akan menampilkan hasil analisis.
4. Menu bar tentang aplikasi memiliki sub menu info yang merupakan menu dimana halaman menu ini akan ditampilkan profil penulis penelitian dan beberapa penjelasan singkat tentang aplikasi.

b. *Activity Diagram*

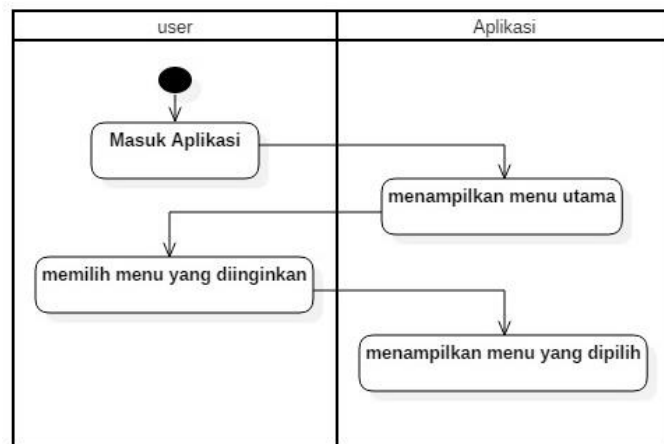
Activity diagram disini merupakan gambaran alur proses atau cara kerja sistem. Pada diagram ini digambarkan aktivitas aktivitas apa saja yang dikerjakan oleh sebuah sistem.

1) *Activity Diagram Menu Home*

Tabel 13 menunjukan pada saat pengguna membuka aplikasi maka akan ditampilkan menu utama aplikasi yang memiliki beberapa pilihan menu yang kemudian pada saat *user* memilih menu yang diinginkan maka

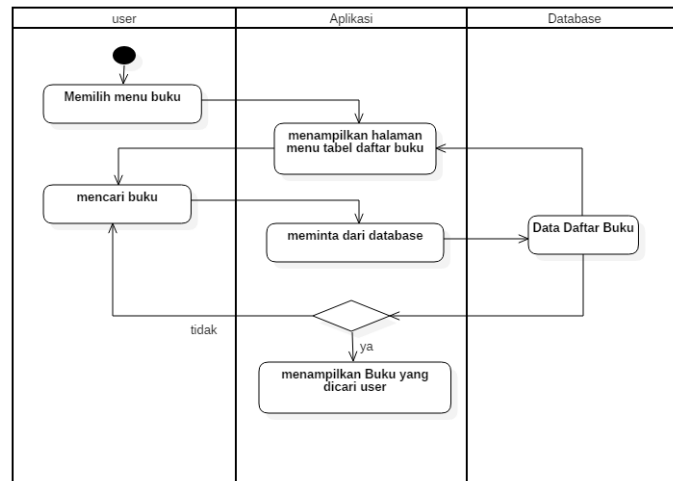
aplikasi akan menampilkan halaman menu yang diminta oleh *user*.

Tabel 13. *Activity Diagram Menu Home*



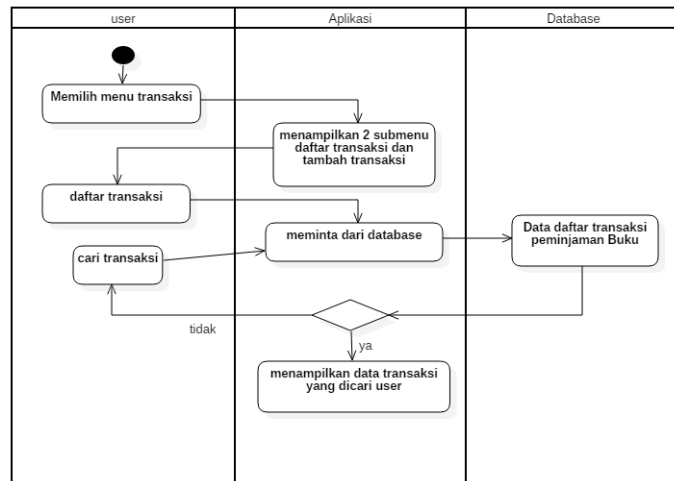
2) *Activity Diagram Menu Buku*

Tabel 14 *activity diagram* menu buku menjelaskan bahwa pada saat *user* memilih menu daftar buku maka sistem akan menampilkan daftar buku yang diminta dari *database*, yang dimana juga pada saat *user* ingin mencari sesuai judul yang diinginkan maka sistem akan meminta dari *database* sesuai dengan judul yang dicari maka akan ditampilkan ke aplikasi sedangkan jika tidak ada maka sistem akan menampilkan tabel kosong dan *user* dapat kembali mencari kembali buku yang diinginkan.

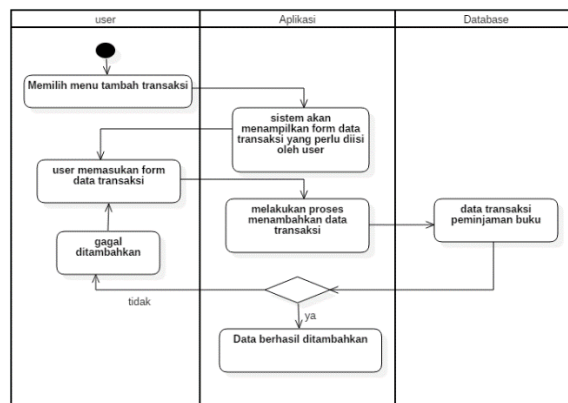
Tabel 14. *Activity Diagram Menu Buku*

3) *Activity Diagram Menu Transaksi*

Tabel 15 *activity diagram* menu transaksi akan memiliki dua pilihan pada saat *user* memilih menu daftar transaksi maka sistem akan meminta dari *database* lalu akan ditampilkan daftar data transaksi ke halaman halaman tabel daftar transaksi, saat *user* ingin mencari daftar transaksi maka sistem meminta dari *database* sesuai dengan yang diminta oleh *user* jadi data transaksi yang diminta terdapat pada *database* maka akan ditampilkan jika tidak maka *user* dapat mencari kembali data transaksi yang diinginkan.

Tabel 15. *Activity Diagram Menu Transaksi*4) *Activity Diagram Menu Tambah Transaksi*

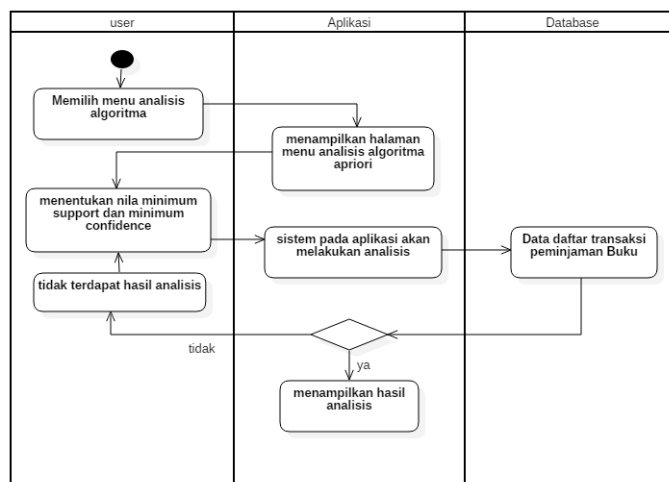
Tabel 16 *activity diagram* tambah transaksi saat *user* meminta untuk menambahkan transaksi maka sistem akan menampilkan form data transaksi yang perlu diisi lalu saat *user* selesai mengisi *form* tambah transaksi maka sistem akan melakukan proses menambahkan data transaksi kedalam *database* jika berhasil maka akan muncul pesan data berhasil ditambahkan jika tidak maka *user* dapat kembali mengisi *form* tambah data transaksi.

Tabel 16. *Activity Diagram Menu Tambah Transaksi*

5) Activity Diagram Algoritma Apriori

Tabel 1 *activity diagram* menu algoritma apriori saat *user* memilih menu analisis maka aplikasi akan menampilkan menu analisis algoritma apriori yang dimana selanjutnya *user* akan menginputkan nilai minimum *support* dan minimum *confidence* yang dimana selanjutnya aplikasi akan melakukan analisis dengan meminta dari data transaksi peminjaman buku yang selanjutnya jika selesai melakukan analisis dan *rule* ditemukan maka aplikasi akan menampilkan hasil analisis jika tidak maka *user* dapat melakukan kembali prosesnya dari menentukan nilai minimum *support* dan *confidence*.

Tabel 17. Activity Diagram Algoritma Apriori

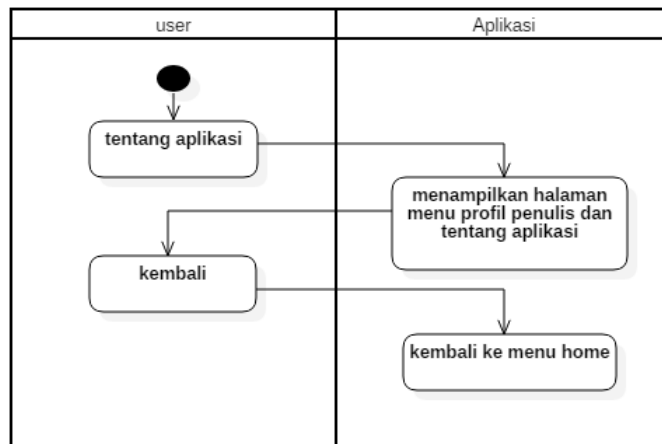


6) Activity Diagram Menu Tentang Aplikasi

Tabel 18 *activity diagram* menu tentang aplikasi menampilkan info tentang aplikasi dan juga profil penulis,

pada saat *user* menekan tombol kembali maka akan dialihkan ke halaman menu *home*.

Tabel 18. *Activity Diagram* Menu Tentang Aplikasi

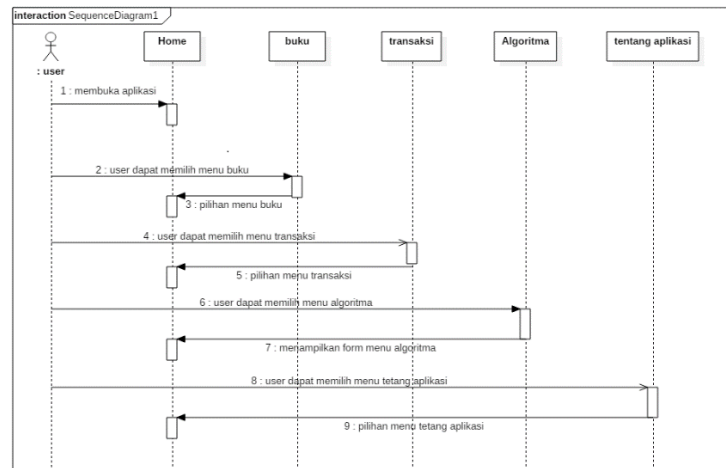


c. *Sequence Diagram*

Sequence Diagram merupakan salah satu *diagram interection* yang menjelaskan bagaimana suatu operasi itu dilakukan pesan yang akan dikirim dan kapan dilaksanakan. Berikut adalah *sequence diagram* dari aplikasi yang dibuat.

1) Tampilan *Sequence Diagram* menu home

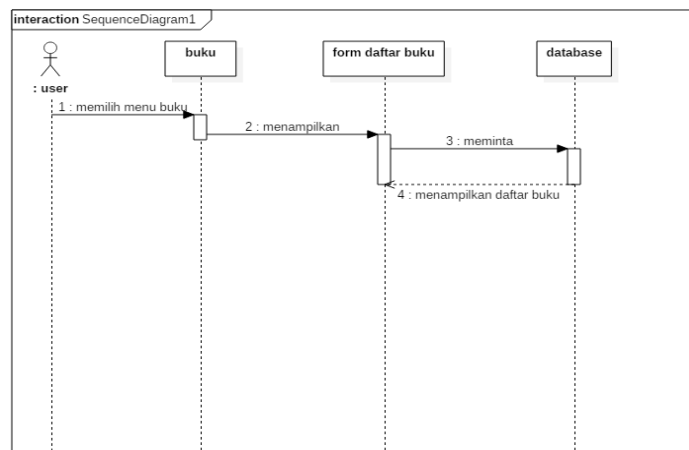
Pada tampilan menu *sequence diagram* menu home aplikasi yang akan menampilkan menu tampilan awal yang dimana pada menu *home user* dapat memilih beberapa menu bar yaitu menu buku, transaksi, algoritma, dan tentang aplikasi.



Gambar 3. *Sequence Diagram* menu home

2) Tampilan *Sequence Diagram* menu buku

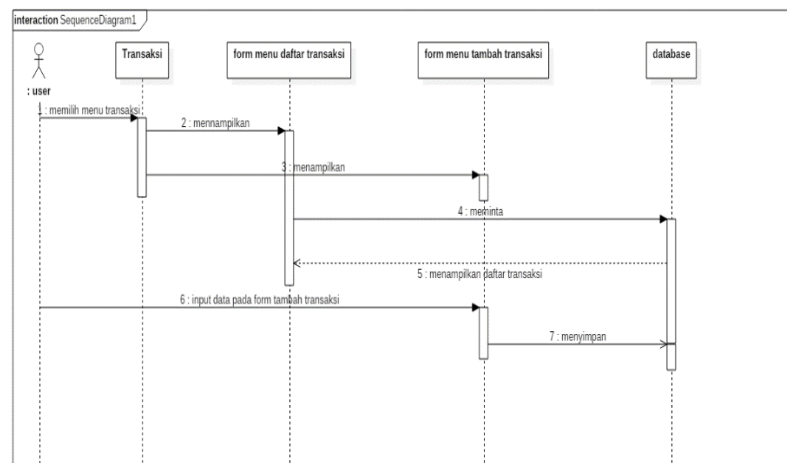
Gambar 4 menunjukkan aplikasi yang akan menampilkan menu buku saat *user* memilih menu bar buku maka akan menampilkan pilihan menu *form* daftar buku yang dimana pada saat masuk menu daftar buku maka sistem akan meminta ke *database* daftar buku dan akan menampilkan data daftar buku ke *form* daftar buku



Gambar 4. *Sequence Diagram* Menu

3) Tampilan *Sequence diagram* menu transaksi

Gambar 5 menunjukkan aplikasi yang akan menampilkan menu transaksi pada saat *user* memilih menu bar transaksi maka akan menampilkan pilihan menu *form* menu daftar transaksi dan *form* menu tambah transaksi yang dimana saat memilih ke menu daftar transaksi maka sistem akan meminta dari *database* maka data daftar transaksi akan ditampilkan ke *form* menu daftar transaksi, jika *user* memilih menu tambah transaksi maka *user* perlu memasukkan data transaksi yang diperlukan pada *form* tambah transaksi dan data tersebut akan disimpan kedalam *database*.

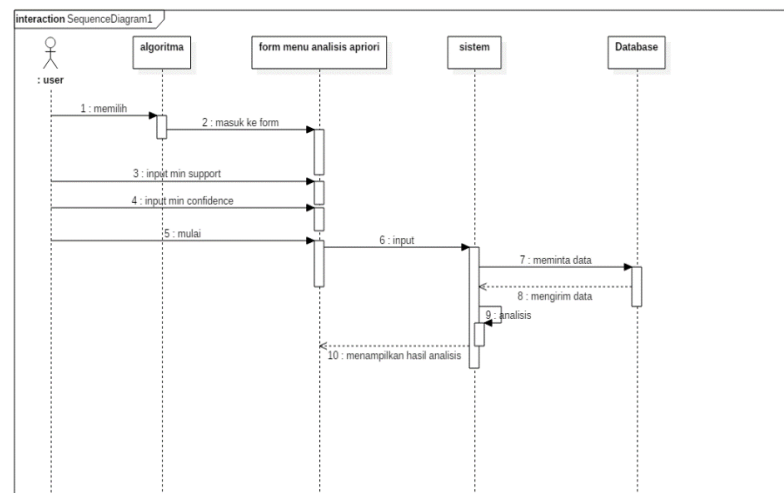


Gambar 5. *Sequence Diagram* Menu Transaksi

4) Tampilan *Sequence Diagram* Apriori

Gambar 6 menunjukkan aplikasi yang akan menampilkan menu Apriori pada saat *user* memilih

menu bar algoritma maka akan menampilkan *form* menu analisis apriori dan user harus menginputkan *min support* dan minimum *confidence* setelah diinputkan maka *user* dapat meminta untuk mulai melakukan analisis yang dimana data minimum *support* dan minimum *confidence* diinputkan kedalam sistem yang selanjutnya sistem akan meminta data transaksi dari *database* dan mengirim ke sistem yang dimana sistem akan melakukan analisis yang selanjutnya hasil analisis akan ditampilkan ke *form* menu analisis apriori.

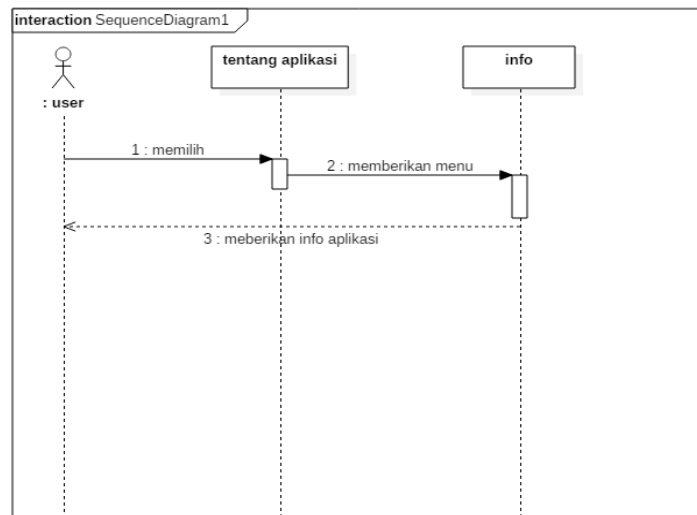


Gambar 6. *Sequence Diagram* Apriori

5) Tampilan *Sequence Diagram* menu Tentang Aplikasi

Gambar 7 menunjukan aplikasi yang akan menampilkan menu tentang aplikasi saat *user* memilih menu bar tentang aplikasi maka akan memberikan

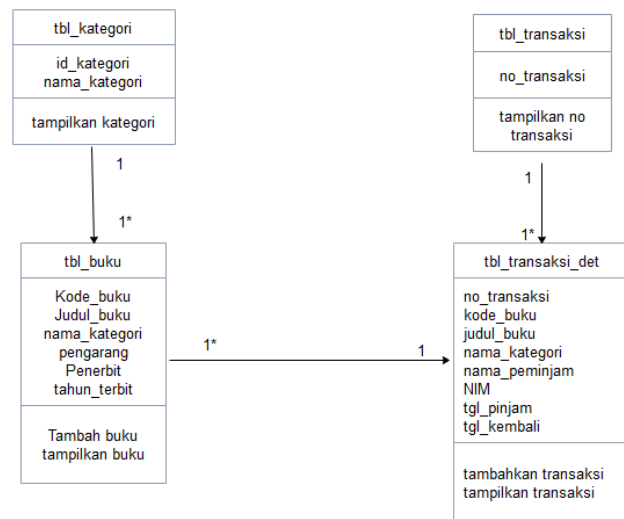
menu info yang dimana menampilkan info aplikasi kepada *user*.



Gambar 7. Sequence Diagram Menu Tentang aplikasi

d. *Class Diagram*

Class Diagram menggambarkan struktur *class* didalam aplikasi. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. Berikut ini merupakan *class diagram* dimana terdapat 4 buah *class diagram* yang saling berhubungan yang dimana menjelaskan *tbl_kategori* dan *tbl_buku* karena satu kategori dapat memiliki banyak buku maka dapat disebut *one to many* sedangkan *tbl_buku* dan *tbl_transaksi_det* *many to one* karena banyak buku dapat dipinjam dalam satu kali transaksi dan untuk *tbl_transaksi* dan *tbl_transaksi_det* *one to many* karena satu no transaksi diambil dari banyak transaksi yang bersamaan. Dapat dilihat pada Gambar 8. *Class Diagram* yaitu sebagai berikut:

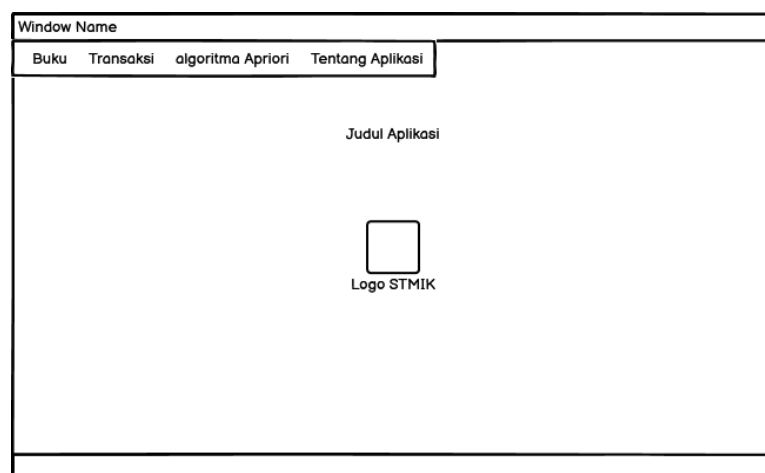


Gambar 8. Class Diagram

3.4.3 Desain Perangkat Lunak

a. Tampilan Menu Utama Aplikasi

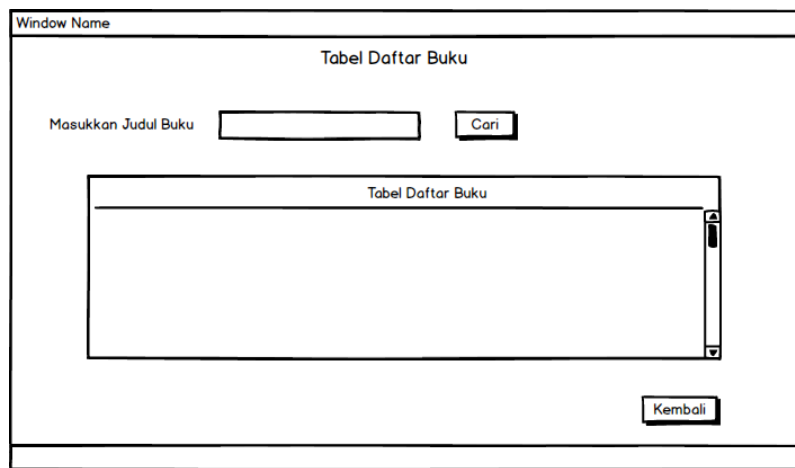
Pada saat membuka aplikasi maka pengguna akan ditampilkan menu awal dengan beberapa menu bar yang dapat dipilih oleh si pengguna di pojok kiri atas dan juga nantinya menampilkan logo STMIK Palangkaraya.



Gambar 9. Tampilan Menu Utama Aplikasi

b. Tampilan Halaman Daftar Buku

Gambar 9 menunjukkan aplikasi yang akan menampilkan menu daftar buku dan akan menampilkan tabel data data daftar buku yang terdapat di perpustakaan, dan juga pada menu halaman ini terdapat fitur yang dapat digunakan untuk mencari buku yang terdapat di perpustakaan STMIK Palangkaraya.



Gambar 10. Tampilan Menu Daftar Buku

c. Tampilan Halaman Tambah Transaksi Peminjaman Buku

Gambar 11 menunjukkan aplikasi yang menampilkan menu halaman tambah transaksi peminjaman ini pengguna aplikasi dapat melakukan prosen penginputan data buku apa saja yang dipinjam oleh mahasiswa sesuai dengan data diri siswa dan juga fitur pencarian judul buku yang ingin dicari agar mempermudah dalam mencari buku saat menginputkan data, yang nantinya akan dimasukan kedalam database jika di klik simpan.

Gambar 11. Menu Tambah Transaksi Peminjaman Buku

b. Tampilan Menu Daftar Transaksi

Gambar 12 menunjukkan tampilan halaman menu ini akan menampilkan tabel daftar semua transaksi peminjaman buku yang terdapat di perpustakaan STMIK Palangkaraya.

Gambar 12. Tampilan Menu Daftar Transaksi

c. Tampilan Menu Analisis

Gambar 13 aplikasi menunjukkan halaman menu analisis pengguna aplikasi terlebih dahulu harus menginputkan atau menentukan minimum *support* dan minimum *confidence* yang diinginkan pada saat mau melakukan analisis, yang dimana selanjutnya dapat dilakukan analisis saat mengklik

tombol analisis, yang setelah beberapa saat akan menampilkan hasil *rule* yang didapatkan dari minimum *support* dan *confidence*, yang akan menghasilkan pola peminjaman buku.

Gambar 13. Tampilan Menu Analisis

d. Tampilan Menu Tentang aplikasi

Gambar 14 pada halaman menu ini menampilkan tentang aplikasi berupa profil pribadi pengembang aplikasi dan juga tujuan dibuatnya aplikasi ini.

Gambar 14. Menu Tentang Aplikasi

3.4.4 Desain Basis Data

a. Desain fisik *database* buku

Tabel ini memuat data buku yang dimana berikut ini nama field, tipe dan keterangan yang dapat dilihat pada tabel 19 berikut :

Tabel 19. Sturktur Tabel Buku

Nama Field	Tipe	Keterangan
Kode_buku	Int(15)	<i>Primary key</i> id buku
Judul_buku	Varchar(30)	Judul pada buku
Nama_kategori	Varchar(50)	Nama judul pada buku
Pengarang	Varchar(50)	Nama penulis buku
Penerbit	Varchar(50)	Penerbit buku
Tahun_terbit	Varchar(50)	Tahun terbitnya buku

b. Desain Fisik *Database* Daftar Transaksi

Tabel ini memuat data transaksi peminjaman buku yang ada di *database* perpustakaan. Struktur tabel transaksi dapat dilihat pada tabel 20 berikut :

Tabel 20. Struktur Tabel Transaksi

Nama Field	Tipe	keterangan
No_transaksi	Int(15)	No transaksi transaksi
Kode_buku	Varchar(30)	<i>Forenky key</i> pada buku

Judul_buku	Varchar(50)	Nama judul pada buku
Nama Field	Tipe	Keterangan
Nama_kategori	Varchar(50)	Nama kategori buku
Nama_peminjam	Varchar(50)	Nama peminjam
NIM	Varchar(50)	NIM peminjam
Tgl_pinjam	Date(50)	Waktu tanggal pinjam
Tgl_kembali	Date(50)	Waktu tanggal kembali

c. Desain Fisik *Database* kategori

Tabel ini memuat data kategori buku yang ada di *database* perpustakaan. Struktur tabel kategori dapat dilihat pada tabel 21 berikut :

Tabel 21. Struktur Tabel Kategori

Nama Field	Tipe	Keterangan
Id_kategori	Int(15)	<i>Primary key</i> id_kategori
Nama_kategori	Varchar(100)	Nama kategori buku

d. Desain Fisik *Database* Tabel Nomor Transaksi

Tabel ini memuat semua nomor transaksi peminjaman buku di *database*. Struktur tabel nomor transaksi dapat dilihat pada tabel 22 berikut :

Tabel 22. Struktur Tabel Nomor Transaksi

Nama Field	Tipe	keterangan
No_transaksi	Int(15)	id_transaksi

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Implementasi

Pada Tahap Implementasi program meliputi uji coba sistem, manual program, untuk menggambarkan program yang dibuat dapat dimengerti dengan baik serta diketahui cara penggunaannya oleh pemakai.

Pengujian yang dilakukan terhadap program yaitu berupa setiap fungsi dan prosedur dalam program yang dijalankan satu persatu hingga dapat menampilkan hasil dan untuk meminimal kesalahan yang terdapat dalam program.

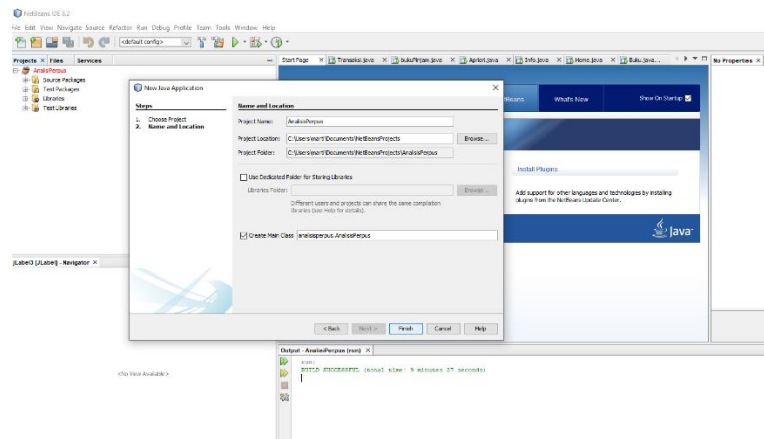
3.4.4.1 Manual Program

Pada Pembahasan ini, Penulis membahas keseluruhan dari proses pembuatan program aplikasi. Pada proses pembuatan program ada beberapa tahap disini penulis menjelaskan beberapa tahapan yang dirasa penting untuk dibahas dalam proses pembuatan program aplikasi, yaitu sebagai berikut.

1. Membuat *project*

Penulis membuka aplikasi netbeans lalu selanjutnya pilih file + new project (ctrl + shift + n) pilih *Java* => *Java Application* lalu *Next* masukan judul *project AnalisisPerpus*

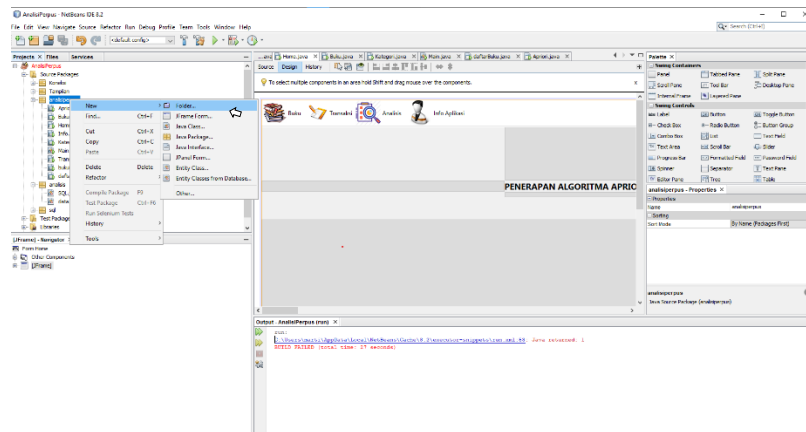
yang akan dibuat dan klik *finish project* pun berhasil dibuatkan. Dapat dilihat pada Gambar 15. Membuat *Project*



Gambar 15. Membuat *Project*

2. Membuat *Jframe*

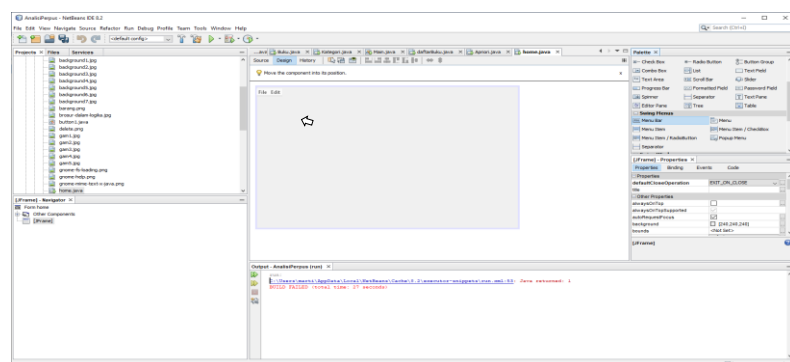
Untuk membuat atau mendesain tampilan aplikasi pertama-tama buatlah *jframe form* langkahnya klik kanan pada judul *project* yang telah dibuatkan lalu pilih *new* => *jframe form* lalu buatlah judul *jframe form* yang akan dibuatkan disini penulis membuatkan *jframe form* menu *home*. Dapat dilihat pada Gambar 16. Desain *Jframe Form* Menu *Home*



Gambar 16. Desain *Jframe Form Menu Home*

3. Membuat Desain Aplikasi

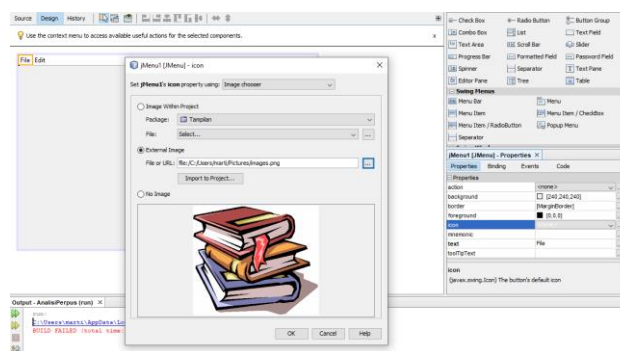
Untuk mendesain *interface* aplikasi dapat dilakukan dengan cara drag and drop pada bagian samping kanan terdapat menu palette lalu selanjutnya pilih rancangan menu yang ingin ditambahkan ke dalam aplikasi dengan cara klik tahan pada fitur menu yang diinginkan lalu lepaskan pada *jframe form* sebagai contoh disini penulis melakukan drag and drop untuk menu bar aplikasi. Dapat dilihat pada Gambar 17.



Gambar 17. Membuat Desain Aplikasi

5. Menambahkan *Icon* Pada Menu Bar

Untuk menambahkan icon pada menu bar dapat dilakukan dengan cara klik menu bar lalu pada samping kanan akan muncul *properties* menu bar yang dipilih dan selanjutnya klik icon pada *properties* menu bar dan akan muncul perintah untuk memilih gambar atau icon yang ingin ditambahkan dan user dapat memilih icon yang terdapat pada komputer user tersebut. Dapat dilihat pada Gambar 18. Menambahkan *Icon* Pada Menu Bar.

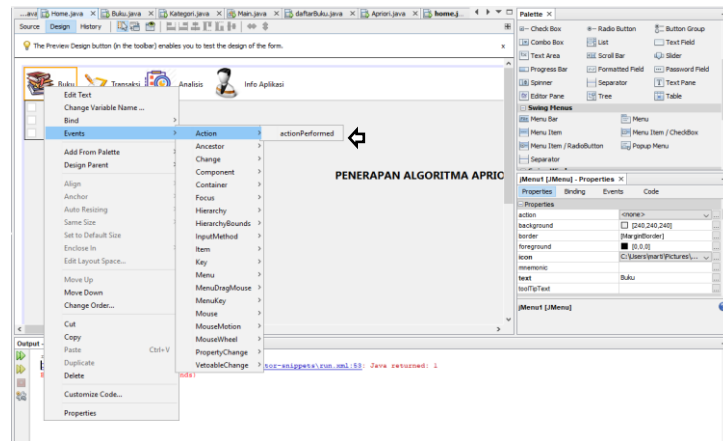


Gambar 18. Menambahkan *Icon* Pada Menu Bar

6. Menambahkan *Coding* Pada Fitur Menu

Setelah selesai mendesain aplikasi untuk menambahkan aksi pada fitur menu ataupun *button* yang telah dibuat caranya dapat dilakukan dengan cara klik kanan pada menu yang ini ditambahkan action pilih *event=>action=>actionPerformed* contohnya disini penulis menambahkan action pada menu bar buku setelah selesai maka akan diarahkan ke *source code* untuk bagian menu bar buku dan user pun dapat mengetikkan *source code*nya yang berisi perintah perintah dalam Bahasa

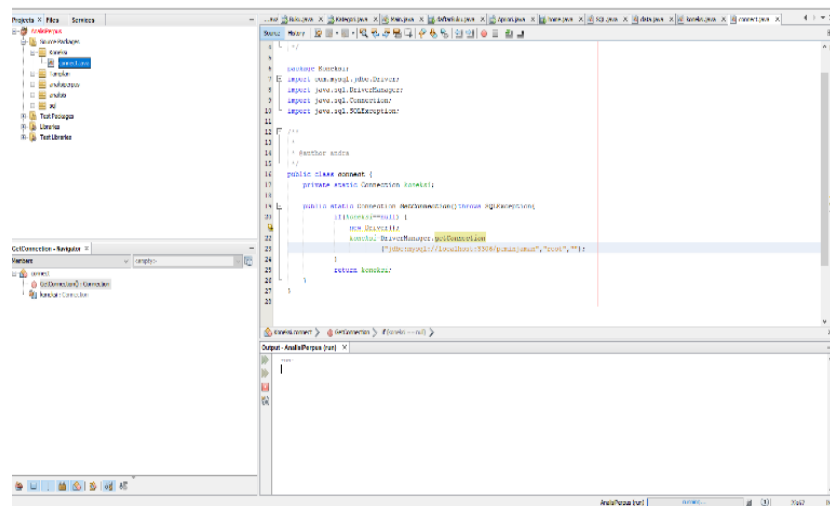
pemrograman. Dapat dilihat pada Gambar 19 Menambahkan Coding Pada Fitur Menu.



Gambar 19. Menambahkan Coding Pada Fitur Menu

7. Menghubungkan Database

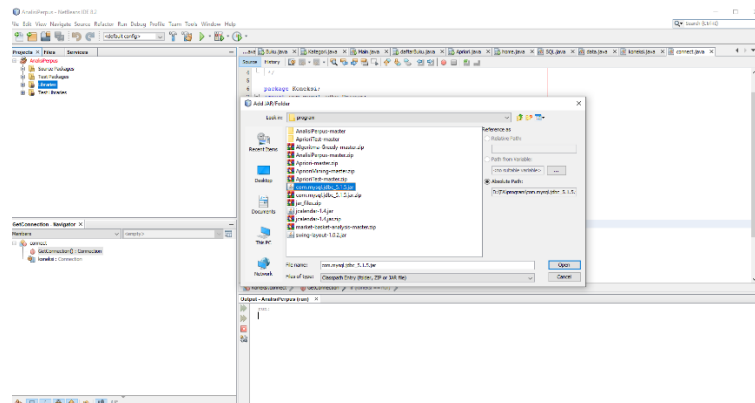
Untuk menghubungkan ke database buat lah *java pageages* dengan cara klik kanan pada *source pageages* pilih *new => java pageages* dan buatlah nama koneksi setelah selesai klik kanan pada *java pageages* koneksi pilih *new=>java class* disini penulis membuat namanya *connect.java* setelah selesai maka akan di arahkan *source page connect* dan disini buatlah *source code* untuk mengkoneksikan database. Dapat dilihat pada Gambar 20. *Connect Database*.



Gambar 20. Connect Database

8. Menambahkan File Jar

Agar aplikasi dapat dikonekan dengan *database* diperlukan file MySQL JDBC Driver yang berekstensi *jar* yang dapat didownload terlebih dahulu cara menambahkannya dapat dilakukan dengan cara klik kanan pada *libraries project* pilih *add JAR/folder* lalu pilihlah file *jar* yang telah didownload tadi dan open maka file MySQL JDBC Driver berhasil ditambahkan. Dapat dilihat pada Gambar 21 Menambahkan MySQL JDBC Driver.



Gambar 21. Menambahkan MySQL JDBC Driver

4.1.2 Pengujian

Pengujian Sistem yang telah dirancang ditunjuk untuk melihat dan menganalisa mengenai kesesuaian sistem terhadap rancangan yang telah dibuat

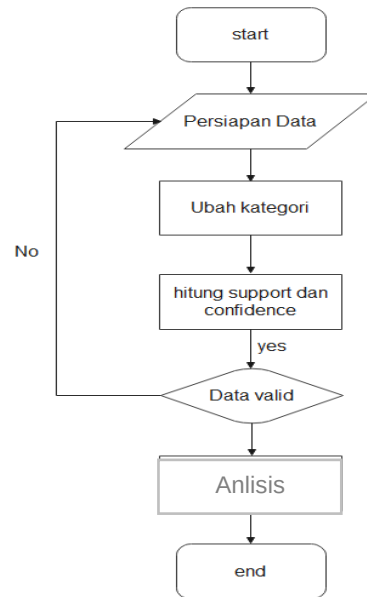
Adapun metode yang digunakan dalam uji coba program ini menggunakan metode pendekatan *black box testing* dan *white box testing*. Setelah melakukan uji coba program secara keseluruhan maka hasil diperoleh bahwa program dapat berjalan sesuai dengan yang diharapkan.

a. Pengujian *white box*

Pengujian *white box testing* lebih menitik beratkan pada struktur internal (*source code*) program, diaman setiap kode yang akan di *compile* atau di cek satu persatu untuk mengetahui apakah masih terjadi error atau tidak dan didapat bahwa *white box testing* menggunakan petunjuk untuk menghasilkan program yang diharapkan dan efisien.

1. Diagram alir (*flowchart*) Analisis

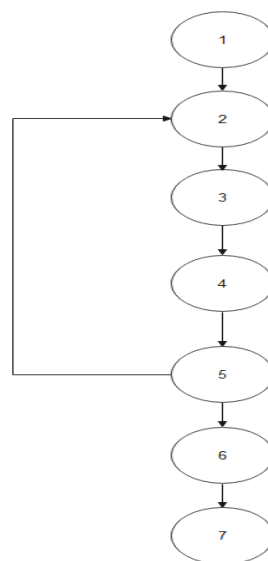
Berikut Gambar Diagram alir (*flowchart*) Analisis.



Gambar 22. Diagram Alir (*Flowchart*) Analisis

2. Grafik Alur (*Flowgraph*) Analisis.

Berikut Gambar Grafik Alir (*flowgraph*) Analisis.



Gambar 23. Grafik Alir (*Flowgraph*) Analisis

3. Listing Program Analysis

Tabel 16. Listing Program Analysis

Node	Source Code
1	<pre>public class Apriori extends javax.swing.JFrame { Vector<data> dt=new Vector<data>(); NumberFormat atur = NumberFormat.getInstance();</pre>
2	<pre>public Apriori(java.awt.Frame parent, boolean modal) { initComponents(); dataFromDataBaseToComboBox(); setLocationRelativeTo(this); TextField1.setText(String.valueOf(sql.bnyTransaksi())); //isidata() `atur.setMaximumFractionDigits(2); }</pre>
3	<pre>public void dataFromDataBaseToComboBox(){ try { Statement statement = (Statement)connect.GetConnection().createStatement(); ResultSet res = statement.executeQuery ("select * from tbl_kategori"); while(res.next()) { cbxKategori.addItem(res.getString("nama_kategori")); } res.last(); res.first(); res.close(); } catch (Exception e) { JOptionPane.showMessageDialog(rootPane, "Salah"); } }</pre>
4	<pre>void c2(){ jTextArea1.setText(""); int n=0; float bnyAB, bnyA, bnyB=0; float supp,conf=0; for(int a=0; a<dt.size();a++){ n++; for(int b=0+n; b<dt.size();b++){ bnyAB=sql.c2(dt.get(a).getKode_buku(), dt.get(b).getKode_buku()); bnyA=dt.get(a).getNilai();</pre>

	<pre> bnyB=dt.get(b).getNilai(); supp=bnyAB/Integer.parseInt(jTextField1.getText()*100; if(supp>=Float.parseFloat(jSpinner1.getValue().toString())){ conf=(bnyAB/bnyA)*100; </pre>
5	<pre> if(conf>= Float.parseFloat(jSpinner2.getValue().toString())) jTextArea1.setText(jTextArea1.getText()+" Jika meminjam buku \" "+dt.get(a).getJudul_buku()+" \" maka akan meminjam buku \" "+dt.get(b).getJudul_buku()+" \" dengan Supp "+atur.format(supp)+"% dan Conf "+atur.format(conf)+"% \n"); conf=(bnyAB/bnyB)*100; if(conf>= Float.parseFloat(jSpinner2.getValue().toString())) </pre>
6	<pre> private void button1ActionPerformed(java.awt.event.ActionEvent evt) { k = cbxKategori.getSelectedItem().toString(); if (k == "Semua Kategori") { sqlkat = "SELECT kode_buku, count(no_transaksi) , judul_buku "+"FROM tbl_transaksi_det GROUP BY kode_buku "+"HAVING count(no_transaksi) >=1 "; } else {sqlkat = "SELECT nama_kategori, kode_buku, count(no_transaksi) , judul_buku "+"FROM tbl_transaksi_det WHERE nama_kategori = '"+k+" GROUP BY kode_buku HAVING count(no_transaksi) >=1 "; }dt.removeAllElements(); try{int temp=Integer.parseInt(jTextField1.getText())*Integer.parseInt(j Spinner1.getValue().toString())/100; koneksi.konek(); ResultSet set = koneksi.stat.executeQuery(sqlkat); while(set.next()){ if(set.getInt("count(no_transaksi)")>=temp){ data data=new data(); data.setKode_buku(set.getString("kode_buku")); data.setJudul_buku(set.getString("judul_buku")); data.setNilai(set.getInt("count(no_transaksi)")); dt.add(data);} } set.close();koneksi.stat.close(); koneksi.con.close(); } </pre>
7	<pre> jTextArea1.setText(jTextArea1.getText()+" Jika meminjam buku \" "+dt.get(b).getJudul_buku()+" \" maka akan meminjam buku \" "+dt.get(a).getJudul_buku()+" \" dengan Supp "+atur.format(supp)+"% dan Conf "+atur.format(conf)+"% \n");} </pre>

4. Kompleksitas Siklomatik

Pengukuran kuantitatif Kompleksitas logis suatu program pada gambar

5.15 dapat diperoleh dengan perhitungan sebagai berikut :

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

Keterangan :

E = Jumlah *edge* pada *flowgraph*

N = Jumlah *node* pada *flowgraph*

R = Jumlah *region* pada *flowgraph*

Sehingga kompleksitas siklomatis yang diperoleh adalah :

a) *Basis Basis Path*

Path 1 : 1-2-3-4-5-6-7

Path 2 : 1-2-3-4-5-2-4-5-6-7

b) *Flowgraph* Mempunyai :

Region (R) : 2

Node (N) : 7

Edge(E) : 7

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

$$= 7 - 7 + 2$$

$$= 2$$

Jadi jumlah *cyclopmtic complexity* untuk *flowgraph* ini adalah 2

5. Pengujian Basis path

Tabel 23. Pengujian Basis Path.

Jalur	Input	Proses	keterangan
Jalur 1	Data berhasil dianalisis	1-2-3-4-5-6-7	Data transaksi berhasil dianalisis, system menampilkan hasil rule yang terbentuk.
Jalur 2	Data tidak menemuka hasil rule	1-2-3-4-5-2-4-5-6-7	Apabila data yang dianalisis tidak memenuhi hasil dari nilai support dan confidence yang ditentukan maka sistem tidak akan menampilkan rule.

b. Pengujian *black box*

Pengujian dilakukan terhadap aplikasi untuk memastikan bahwa aplikasi dapat berjalan dengan benar sesuai dengan kebutuhan dan tujuan yang diharapkan. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.

Tujuan dari metode *black box testing* ini adalah berfokus pada persyaratan fungsional perangkat lunak. Berikut pengujiannya:

Tabel 24. Uji *black box*

No	Fungsi yang diuji	input	Output yang diharapkan	Hasil	Output
1	Menampilkan menu utama	Mengklik aplikasi	Sistem menampilkan menu utama	Berhasil	Sistem berhasil menampilkan menu utama aplikasi
2	Menampilkan	Memilih	Sistem menampilkan	Berhasil	Sistem berhasil

No	Fungsi yang diuji	input	Output yang diharapkan	Hasil	Output
	menu bar	menu bar	sub menu		menampilkan sub menu yang diklik oleh user
3	Menampilkan menu daftar buku	Memilih menu bar buku dan klik sub menu buku	Sistem menampilkan seluruh data daftar buku	Berhasil	Sistem berhasil menampilkan seluruh data daftar buku
4	Menambahkan menu tambah buku	Memasukkan data buku yang akan ditambahkan	Sistem menampilkan form tambah data buku yang diperlukan untuk di inputkan oleh user	Berhasil	Sistem berhasil menampilkan form tambah data buku dan berhasil menyimpan data buku ke dalam database
5	Menampilkan menu transaksi	Memilih menu bar transaksi dan klik menu transaksi	Sistem menampilkan menu daftar transaksi dan semua data transaksi	Berhasil	Sistem berhasil menampilkan menu daftar transaksi dan semua data transaksi
6.	Menambahkan data transaksi	Memasukkan data transaksi yang ditambahkan	Sistem menampilkan form menu tambah transaksi	Berhasil	Sistem berhasil menampilkan form menu tambah transaksi dan menambahkan data transaksi peminjaman ke dalam <i>database</i>
7	Menampilkan hasil analisis menggunakan algoritma apriori	Memasukkan nilai <i>support</i> dan <i>confidence</i>	Sistem menampilkan hasil analisis	Berhasil	Sistem berhasil menampilkan hasil analisis dan ditampilkan pada menu algoritma apriori

No	Fungsi yang diuji	input	Output yang diharapkan	Hasil	Output
8	Menampilkan menu tentang saya	Memilih menu bar info aplikasi dan klik sub info	Menampilkan menu tentang aplikasi	Berhasil	Berhasil menampilkan menu profil penulis

c. Pengujian hasil perhitungan manual

Dari pengujian perbandingan perhitungan manual dengan hasil analisis menggunakan aplikasi mendapatkan mendapatkan hasil rule atau aturan asosiasi yang sama dari nilai yang memenuhi untuk *support* 2 % dan *confidence* 50% dari 48 sampel data transaksi peminjaman maka untuk menyimpulkan buku mana yang sering dipinjam adalah hasil analisis yang memiliki nilai *support* dan *confidence* paling tinggilah yang akan diambil. Dapat dilihat pada tabel perbandingan pengujiannya untuk nilai *support* dan *confidence* yang paling tinggi.

Table 25. Perbandingan hasil pengujian

rule	aplikasi		manual	
	support	confidence	support	confidence
Pemograman Microsoft visual c++ → matematika diskrit	4,17 %	100%	4,17 %	100%
Aljabar Linier Dasar →	4,17 %	100%	4,17 %	100%

rule	aplikasi		manual	
	support	confidence	support	confidence
Aljabar Linier Elementer				
Kalkulus dan Geometri Analisis Jilid 2 → Kalkulus Edisi Kesembilan Jilid 1	4,17 %	100%	4,17%	100%
Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL → Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android	4,17 %	100%	4,17 %	100%

d. Kelemahan Hasil Pengujian Sistem

Dari hasil pengujian sistem terdapat beberapa kelemahan yang terdapat pada aplikasi dan dari beberapa percobaan pengujian aplikasi diantaranya sebagai berikut hasil pengujiannya :

1. Aplikasi hanya mampu melakukan analisis dalam jumlah yang sedikit yaitu hanya sebanyak 48 data transaksi peminjaman.
2. Aplikasi membutuhkan waktu sekitar 3-5 menit untuk melakukan analisis dari 48 data transaksi.

3. Aplikasi tidak mampu melakukan analisis dengan data transaksi sebanyak 145 data transaksi peminjaman.
4. Aplikasi kadang kadang mampu melakukan analisis dengan jumlah 70 data transaksi peminjaman namun aplikasinya tidak dapat bekerja lagi atau not responding dan dipaksakan harus keluar aplikasi.
5. Semakin banyak data transaksi maka diperlukan juga waktu yang cukup lama untuk melakukan analisis
6. Jika menggunakan data transaksi 16-25 data transaksi aplikasi dapat melakukan analisis dalam waktu yang cukup singkat yaitu sekitar 1 menit.

4.2 Pembahasan

4.2.1 Pembahasan Listing Program

Dalam pembahasan listing program akan dijelaskan mengenai *source code* program yang ada beserta penjelasannya yang dimana disini penulis hanya membahas *source code* yang dirasa penting dan juga penerapan algoritma yang dimasukan kedalam program untuk aplikasi penerapan algoritma apriori.

- a. *Source code* menu tambah transaksi

Source code ini berfungsi untuk melakukan proses menambahkan data daftar transaksi buku ke dalam database yang dimana data tersebut nantinya akan digunakan untuk melakukan analisis. Dapat dilihat pada 15 gambar listing program menu tambah transaksi

```
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String tampilan = "yyyy-MM-dd";
    SimpleDateFormat format = new SimpleDateFormat(tampilan);
    int no = 0;
    String nama = txtNama.getText();
    String npm = txtNPM.getText();
    String tglpinjam =
String.valueOf(format.format(TglPinjam.getDate()));
    String tglkembali =
String.valueOf(format.format(TglKembali.getDate()));
    try {
        Statement statement = (Statement)
            connect.GetConnection().createStatement();
        statement.executeUpdate
            ("insert into tbl_transaksi values(Null);");

        ResultSet res = statement.executeQuery
            ("select * from tbl_transaksi ORDER BY no_transaksi
ASC");
        res.last();
        String hasil =res.getString("no_transaksi");
        no = Integer.parseInt(hasil);

        for(int x=0;x<TabelBukuPinjam.getRowCount();x++){
            String kode = TabelBukuPinjam.getValueAt(x,
0).toString();
            String buku = TabelBukuPinjam.getValueAt(x,
1).toString();
            String kat = TabelBukuPinjam.getValueAt(x,
2).toString();
            connect.GetConnection().createStatement();
            statement.executeUpdate
                ("insert into tbl_transaksi_det
values('"+no+"','"+kode+"','"+buku+"','"+kat+"','"+nama+"','"+npm+"',
','"+tglpinjam+"','"+tglkembali+"');");
        }
    }
}
```

```

        JOptionPane.showMessageDialog(null, "Data Berhasil
Disimpan");
    } catch (Exception t) {
        JOptionPane.showMessageDialog(null, "Data Gagal
Disimpan");
    }

```

Gambar 24. Listing Program Menu Tambah Transaksi

b. *Source code* menu tambah buku

Source code ini berfungsi melakukan proses penambahan data buku ke dalam database. Dapat dilihat pada Gambar 25 *Listing Program* menu tambah buku.

```

public class Buku extends javax.swing.JFrame {

    /** Creates new form Buku */
    public Buku() {
        initComponents();
        dataTable();
        dataFromDataBaseToComboBox();
    }
    @SuppressWarnings("unchecked")
    public String kode_lama;
    private void
    btnTambahActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String kode = txtKode.getText();
        String judul = txtJudul.getText();
        String kategori = (String) cbxKategori.getSelectedItem();
        String pengarang = txtPengarang.getText();
        String penerbit = txtPenerbit.getText();
        String tahun = txtTahun.getText();
        try {
            Statement statement = (Statement)
                connect.GetConnection().createStatement();
            statement.executeUpdate
                ("insert into tbl_buku values('" + kode + "', '" + judul + "',

```

```

""+kategori+", ""+pengarang+", ""+penerbit+", ""+tahun+"");");
    dataTable();
    reset();
    JOptionPane.showMessageDialog(null, "Data Berhasil
Disimpan");
    }catch(Exception t){
        JOptionPane.showMessageDialog(null, "Data Gagal
Disimpan");
    }
}

```

Gambar 25. Listing Program Menu Tambah Buku

c. *Source code* analisis algoritma apriori

Source code ini berfungsi untuk melakukan proses perhitungan menggunakan algoritma apriori yang dimasukan kedalam program. Dapat dilihat pada Gambar 17 *Listing Program Analisis Algoritma Apriori*.

```

void c2(){
    jTextArea1.setText("");
    int n=0;
    float bnyAB, bnyA, bnyB=0;
    float supp,conf=0;

    for(int a=0; a<dt.size();a++){
        n++;
        for(int b=0+n; b<dt.size();b++){
            bnyAB=sql.c2(dt.get(a).getKode_buku(),
dt.get(b).getKode_buku());
            bnyA=dt.get(a).getNilai();
            bnyB=dt.get(b).getNilai();
            supp=bnyAB/Integer.parseInt(jTextField1.getText())*100;

            if(supp>=Float.parseFloat(jSpinner1.getValue().toString())){
                conf=(bnyAB/bnyA)*100;
                if(conf>=
Float.parseFloat(jSpinner2.getValue().toString()))

```

```

        jTextArea1.setText(jTextArea1.getText()+" Jika
meminjam buku \" "+dt.get(a).getJudul_buku()+" \" maka akan
meminjam buku \" "+dt.get(b).getJudul_buku()+" \" dengan
Supp "+atur.format(supp)+"% dan Conf
"+atur.format(conf)+"% \n");
        conf=(bnyAB/bnyB)*100;
        if(conf>=
Float.parseFloat(jSpinner2.getValue().toString()))
        jTextArea1.setText(jTextArea1.getText()+" Jika
meminjam buku \" "+dt.get(b).getJudul_buku()+" \" maka akan
meminjam buku \" "+dt.get(a).getJudul_buku()+" \" dengan
Supp "+atur.format(supp)+"% dan Conf "+atur.format(conf)+"%
\n");
    }
}
}
}

void c3(){
    int n=0,m=0,o=0;
    String item1,item2,item3="";
    float bnyABC,bnyAB=0;
    float supp,conf=0;
    for(int a=0; a<dt.size();a++){ n++;
        for(int b=0+n ;b<dt.size();b++){
            for(int c=0+b+1; c<dt.size();c++){
                bnyABC=sql.c3(dt.get(a).getKode_buku(),
dt.get(b).getKode_buku(), dt.get(c).getKode_buku());

supp=bnyABC/Integer.parseInt(jTextField1.getText())*100;

if(supp>=Float.parseFloat(jSpinner1.getValue().toString())){
    bnyAB=sql.c2(dt.get(a).getKode_buku(),
dt.get(b).getKode_buku());
    conf=(bnyABC/bnyAB)*100;
    if(conf>=
Float.parseFloat(jSpinner2.getValue().toString()))
        jTextArea1.setText(jTextArea1.getText()+" Jika
meminjam buku \" "+dt.get(a).getJudul_buku()+" \" dan buku \"
"+dt.get(b).getJudul_buku()+" \" maka akan meminjam buku \"

```



```

"+dt.get(c).getJudul_buku()+" \" dengan Supp
"+atur.format(supp)+"% dan Conf "+atur.format(conf)+"% \n");

        bnyAB=sql.c2(dt.get(a).getKode_buku(),
dt.get(c).getKode_buku());
        conf=(bnyABC/bnyAB)*100;
        if(conf>=
Float.parseFloat(jSpinner2.getValue().toString()))
            jTextArea1.setText(jTextArea1.getText()+" Jika
meminjam buku \" "+dt.get(a).getJudul_buku()+" \" dan buku \"
"+dt.get(c).getJudul_buku()+" \" maka akan meminjam buku \"
"+dt.get(b).getJudul_buku()+" \" dengan Supp
"+atur.format(supp)+"% dan Conf "+atur.format(conf)+"% \n");
            bnyAB=sql.c2(dt.get(b).getKode_buku(),
dt.get(c).getKode_buku());
            conf=(bnyABC/bnyAB)*100;
            if(conf>=
Float.parseFloat(jSpinner2.getValue().toString()))
                jTextArea1.setText(jTextArea1.getText()+" Jika
meminjam buku \" "+dt.get(b).getJudul_buku()+" \" dan buku \"
"+dt.get(c).getJudul_buku()+" \" maka akan meminjam buku \"
"+dt.get(a).getJudul_buku()+" \" dengan Supp
"+atur.format(supp)+"% dan Conf "+atur.format(conf)+"% \n");
        }
    }
}
}
}
}
}

```

Gambar 26. *Listing Program Analisis Algoritma Apriori*

4.4.2 Pembahasan *Interface*

Pada aplikasi penerapan algoritma apriori ini seorang user dapat menggunakan semua hak akses yang terdapat pada sistem yang dimana beberapa hak akses tersebut dapat dilihat sebagai berikut ini :

- a. Halaman Menu Utama Aplikasi

Halaman menu utama aplikasi menampilkan tampilan beberapa menu bar yang dapat dipilih oleh user di pojok kiri dan juga menampilkan logo STMIK Palangkaraya yang berada pada tengah aplikasi dapat dilihat pada Gambar 27. Menu Utama Aplikasi berikut



Gambar 27. Menu Utama Aplikasi

b. Halaman Menu Daftar Buku

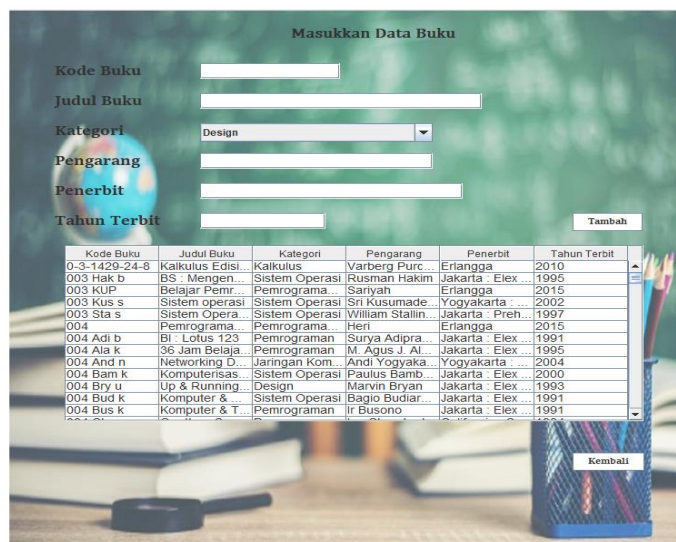
Pada halaman menu daftar buku akan menampilkan tabel data data daftar buku yang terdapat di perpustakaan, dan juga pada menu halaman ini terdapat fitur yang dapat digunakan untuk mencari buku yang terdapat di perpustakaan STMIK Palangkaraya. Dapat dilihat pada Gambar 28. Menu Daftar Buku berikut



Gambar 28. Menu Daftar Buku

c. Halaman Menu Tambah Daftar Buku

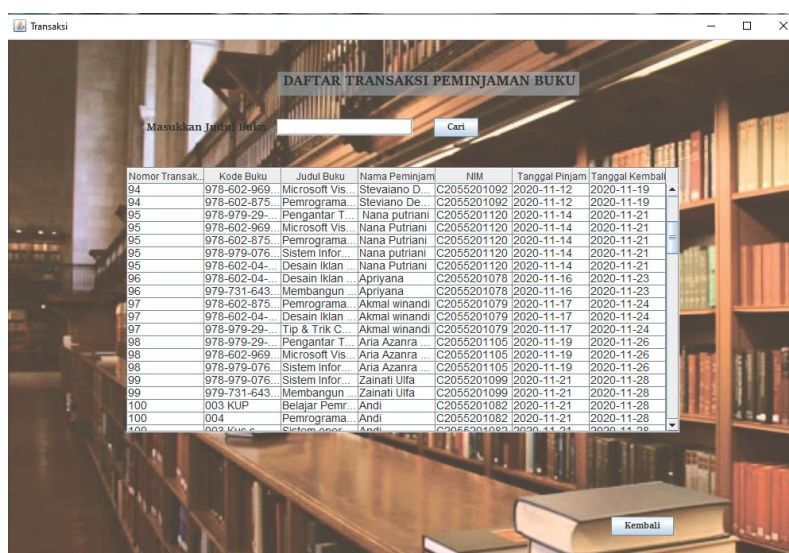
Pada halaman menu tambah daftar buku seorang user diharuskan mengisi form data buku yang akan ditambahkan selanjutnya jika telah selesai maka user dapat memilih fitur tambahkan maka data buku akan ditambahkan ke dalam database. Dapat dilihat pada Gambar 29. Menu Tambah Daftar Buku berikut



Gambar 29. Menu Tambah Daftar Buku

d. Halaman Menu Daftar Transaksi

Pada halaman menu ini akan menampilkan tabel daftar semua transaksi peminjaman buku yang terdapat di perpustakaan STMIK Palangkaraya. Dapat dilihat pada Gambar 30. Menu Daftar Transaksi berikut



Nomor Transak.	Kode Buku	Judul Buku	Nama Peminjam	NIM	Tanggal Pinjam	Tanggal Kembali
94	978-602-969...	Microsoft Vis...	Steviano D.	C2055201092	2020-11-12	2020-11-19
94	978-602-875...	Pemrograma...	Steviano De...	C2055201092	2020-11-12	2020-11-19
95	978-979-29...	Pengantar T...	Nana putriani	C2055201120	2020-11-14	2020-11-21
95	978-602-969...	Microsoft Vis...	Nana Putriani	C2055201120	2020-11-14	2020-11-21
95	978-602-875...	Pemrograma...	Nana Putriani	C2055201120	2020-11-14	2020-11-21
95	978-979-076...	Sistem Infor...	Nana putriani	C2055201120	2020-11-14	2020-11-21
95	978-602-04...	Desain iklan...	Nana Putriani	C2055201120	2020-11-14	2020-11-21
96	978-602-04...	Desain iklan...	Apriyana	C2055201078	2020-11-16	2020-11-23
96	979-731-643...	Membangun...	Apriyana	C2055201078	2020-11-16	2020-11-23
97	978-602-875...	Pemrograma...	Akmal winandi	C2055201079	2020-11-17	2020-11-24
97	978-602-04...	Desain iklan...	Akmal winandi	C2055201079	2020-11-17	2020-11-24
97	978-979-29...	Tip & Trik C...	Akmal winandi	C2055201079	2020-11-17	2020-11-24
98	978-979-29...	Pengantar T...	Aria Azanra	C2055201105	2020-11-19	2020-11-26
98	978-602-969...	Microsoft Vis...	Aria Azanra	C2055201105	2020-11-19	2020-11-26
98	978-979-076...	Sistem Infor...	Aria Azanra	C2055201105	2020-11-19	2020-11-26
99	978-979-076...	Sistem Infor...	Zainati Uifa	C2055201099	2020-11-21	2020-11-28
99	979-731-643...	Membangun...	Zainati Uifa	C2055201099	2020-11-21	2020-11-28
100	003 KUP	Belajar Pemr...	Andi	C2055201082	2020-11-21	2020-11-28
100	004	Pemrograma...	Andi	C2055201082	2020-11-21	2020-11-28
100	003 KUP	Sistem Infor...	Andi	C2055201082	2020-11-21	2020-11-28

Gambar 30. Menu Daftar Transaksi

e. Halaman Menu Tambah Daftar Transaksi Peminjaman

Pada menu halaman tambah transaksi peminjaman ini pengguna aplikasi dapat melakukan prosen penginputan data buku apa saja yang dipinjam oleh mahasiswa sesuai dengan data diri siswa dan juga fitur pencarian judul buku yang ingin dicari agar mempermudah dalam mencari buku saat menginputkan data, yang nantinya akan dimasukan kedalam database jika di klik simpan. Dapat dilihat pada Gambar 31. Menu Tambah Transaksi Peminjaman Buku.

Buku	Jml	Kategori	Peminjam	Tanggal
1. Kaidah & Kaidah	1	Kaidah	Kaidah	2015
2. Kaidah & Kaidah	1	Kaidah	Kaidah	2015
3. Kaidah & Kaidah	1	Kaidah	Kaidah	2015
4. Kaidah & Kaidah	1	Kaidah	Kaidah	2015
5. Kaidah & Kaidah	1	Kaidah	Kaidah	2015
6. Kaidah & Kaidah	1	Kaidah	Kaidah	2015
7. Kaidah & Kaidah	1	Kaidah	Kaidah	2015
8. Kaidah & Kaidah	1	Kaidah	Kaidah	2015
9. Kaidah & Kaidah	1	Kaidah	Kaidah	2015
10. Kaidah & Kaidah	1	Kaidah	Kaidah	2015

Gambar 31. Menu Tambah Transaksi Peminjaman

f. Halaman Menu Analisis

Pada halaman menu analisis pengguna aplikasi terlebih dahulu harus menginputkan atau menentukan minimum *support* dan minimum *confidence* yang diinginkan pada saat mau melakukan analisis, yang dimana selanjutnya dapat dilakukan analisis saat mengklik tombol analisis, yang setelah beberapa saat akan menampilkan hasil *rule* yang didapatkan dari minimum *support* dan *confidence*, yang akan menghasilkan pola peminjaman buku. Dapat dilihat pada Gambar 32. Menu Analisis

Minimum Support: 0.1%
 Nilai Minimum dari jumlah transaksi yang meminjam buku yang sama

Minimum Confidence: 0.1%
 Level minimum suatu tingkat kepercayaan antara suatu buku

Count: 70
 Keseluruhan banyak jumlah suatu transaksi

Bentuk Kategori: [Dropdown Menu]

Analisis Kembali

Gambar 32. Menu Analisis

g. Halaman Tentang Aplikasi

Pada halaman menu ini menampilkan tentang aplikasi berupa profil pribadi pengembang aplikasi dan juga tujuan dibuatnya aplikasi ini. Dapat dilihat pada Gambar 33. Tentang

Aplikasi



Gambar 33. Menu Tentang Aplikasi

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis, implementasi dan pengujian dari aplikasi ini maka penulis dapat menarik kesimpulan sebagai berikut:

- a. Dalam penelitian ini telah dihasilkan sebuah “Aplikasi Penerapan Algoritma Apriori Untuk Pencarian Pola Peminjaman Buku Di Perpustakaan STMIK Palangkaraya” yang dapat melakukan analisis pola peminjaman buku.
- b. Berdasarkan dari hasil pengujian aplikasi yang dibuat dapat mengimplementasikan penerapan algoritma apriori dari sampel 48 data transaksi peminjaman buku di perpustakaan STMIK Palangkaraya dan menghasilkan *rule* atau aturan pola peminjaman buku.
- c. Hasil pengujian analisis pada aplikasi, *rule* atau aturan pola peminjaman buku dengan nilai *support* 2% dan *confidence* 50% dari 48 sampel data transaksi menghasilkan “Jika meminjam buku " Pemograman Microsoft visual c++ " maka akan meminjam buku " matematika diskrit " dengan Supp 4,17 % dan Conf 100 %”.
- d. Hasil pengujian lainnya yang memiliki nilai *support* paling tinggi yaitu supp 4,17 % dan conf 100% berikut “Aljabar Linier Dasar → Aljabar Linier Elementer, Kalkulus dan Geometri Analisis Jilid 2 → Kalkulus Edisi Kesembilan Jilid 1, Aplikasi Pemrograman Web

Dinamis dengan PHP dan MySQL → Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android”.

- e. Dari hasil perbandingan dengan menggunakan perhitungan aplikasi dan perhitungan secara manual juga mendapatkan *rule* atau aturan maupun nilai *support* dan *confidence* yang sama.
- f. Penerapan algoritma apriori dalam bahasa pemrograman java yang diterapkan pada aplikasi memiliki kekurangan dalam mengolah data dengan jumlah yang cukup banyak, hal ini disebabkan oleh proses perulangan dalam mengolah data dan kemudian menyeleksi data satu per satu dari database sehingga membuat *coding* program yang diterapkan penulis dalam bahasa java tidak mampu untuk melakukan analisis dalam jumlah data yang banyak dan hasilnya aplikasi tidak dapat berjalan dengan baik.
- g. Dari hasil rule atau aturan yang terbentuk bisa disimpulkan dari nilai *support* dan *confidence* yang paling tinggi dari kombinasi antara dua item tersebut dapat digunakan untuk merekomendasikan buku berdasarkan buku mana yang sering dipinjam secara bersamaan

5.2 Saran

Dari hasil pembahasan dan kesimpulan, penulis memberikan beberapa saran yang mungkin berguna untuk pengembangan lebih lanjut yaitu:

- a. Memperbaharui bahasa pemrograman java yang digunakan penulis agar aplikasi ini dapat melakukan analisis dalam jumlah data transaksi yang banyak.

- b. Ditambahkan algoritma lainnya sebagai perbandingan perhitungan sehingga hasil yang didapat lebih akurat.
- c. Diterapkannya algoritma apriori kedalam bahasa pemrograman yang dapat lebih mempercepat waktu dalam melakukan analisis pada jumlah data yang lebih banyak dibandingkan dengan bahasa pemrograman Java.

DAFTAR PUSTAKA

- Abdal, M., & Rosadi, M. (2019). Penerapan Association Rule Mining Untuk Analisis Penempatan Tata letak Buku Di Perpustakaan Menggunakan Algoritma Apriori. *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, 5(1), 99-108.
- Budiyati, E., Hurniningsih, & Lusita, M. d. (2020). Implementasi Metode Algoritma Apriori Untuk Penempatan Buku Pada Rak. *Journal Of Information System, Informatics and Computing*, 4(1), 30-39.
- Buulolo, & Efori. (2013). Implementasi Algoritma Apriori Pada Sistem Persediaan Obat. *Jurnal ISSN: Pelita Informatika Budi Darma*, IV(1).
- Hartono. (2016). *Manajemen Perpustakaan Sekolah*. Yogyakarta: Ar-Ruzz Media.
- Kusrini, & Luthfi, E. T. (2017). *Algoritma Data Mining*. Yogyakarta: Andi.
- Ladjamudin, A.-B. B. (2013). *Analisi Sistem Informasi*. Yogyakarta: Andi.
- Manurung, O., & Hasugian, P. S. (2019). Analisa Algoritma Apriori Untuk Peminjaman Buku pada Perpustakaan SMA 1 Silima Pungga-Pungga Parongil. *Jurnal REMIK Riset dan E-Jurnal Manajemen Informatika Komputer*, 4(1).
- Marisa, F., & Wijaya, I. D. (2018). Penerapan Metode Data Mining Market Basket analysis terhadap Data Penjualan Produk Pada Toko Oase Menggunakan Algoritma Apriori. *Jurnal Informatika Merdeka Pasuruan*, 3(2), 17-22.
- Mulyani. (2016). *Metode Analisis dan Perancangan Sistem*. Bandung: Abdi Sitematika.
- Madcoms. (2016). *Sukses Membangun Toko Online dengan PHP & MySQL*. Yogyakarta: Andi.
- Pane, D. K. (2013). Implementasi Data Mining pada Penjualan Produk Elektronik Dengan Algoritma Apriori (Studi Kasus : Kreditplus). *Pelita Informatika Budi Darma*, 4(3).
- Pracoyo, S., & Seniwat, E. (2018). Algoritma Apriori Untuk Penempatan Buku Di Perpustakaan SMK MA'ARIF 1 Wates. *Jurnal Sistem Informasi Universitas AMIKOM Yogyakarta*, 1(2), 1-6.

- Sastie, D. E., Suparni, & Pohan, A. b. (2020). Analisa Algoritma Apriori Pada Pola Peminjaman Buku di Perpustakaan ITB Ahmad Dahlan. *Jurnal Media Informatika Budidarma*, 4(1), 136-143.
- Sholik, M., & Salam, A. (2018). Implementasi Algoritma Apriori untuk Mencari Asosiasi Barang yang Dijual di E-commerce OrderMas. *Jurnal Techno*, 17(2), 158-170.
- Srikanti, E., Yansi, R. F., Norhavina, Permana, I., & Salisah, f. N. (2018). Penerapan Algoritma Apriori Untuk Mencari Aturan Asosiasi Pada Data peminjaman Buku Di perpustakaan. *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, 4(1), 77-80.
- Suyanto. (2015). Pemograman Java Pengenalan Java. *Jurnal Komputer*.
- Sukamto, R., & Shalahudin, M. (2018) *Rekayasa Perangkat Lunak*. Bandung: Informatika Bandung
- Tampubolon, K., Saragih, H., & Reza, B. (2013). Data Mining Algoritma Apriori Pada Sistem Persediaan Alat Alat Kesehatan. *Jurnal Informasi dan Teknologi ilmiah*, 1(1), 93-106.
- Tommy, L., Kirana, C., & Lindawati, V. (2019). Recommender System dengan Kombinasi Apriori dan Content-Based Filtering Pada Aplikasi Pada Pemesanan produk. *Jurnal TEKNOINFO*, 13(2), 84-95.



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA**

Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangkaraya
email : humas@stmikpk.ac.id - website : www.stmikpk.ac.id

SURAT TUGAS
No.4/STMIK-3.C.2/AU/1/2021

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan nama-nama tersebut di bawah ini :

1. Nama : Hotmian Sitohang, M.Kom.
NIK : 198503282008002
Sebagai Pembimbing I Dalam Pembuatan Program
2. Nama : Elok Falqotul Himmah, S.Si., M.Sc.
NIK : 198503092009003
Sebagai Pembimbing II Dalam Penulisan Tugas Akhir

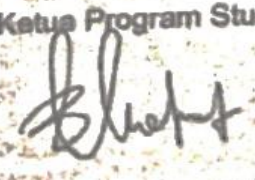
Untuk membimbing Tugas Akhir mahasiswa :

Nama : Martin
NIM : C1755201079
Program Studi : TEKNIK INFORMATIKA (55201)
Tanggal Daftar : 28 September 2020
Judul Tugas Akhir : Penerapan Algoritma Apriori Untuk Pencarian Pola Peminjaman Buku di Perpustakaan STMIK Palangkaraya

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 15 Januari 2021

Ketua Program Studi,


Hotmian Sitohang, M.Kom.
NIK. 198503282008002

Tembusan :

1. Pembimbing I dan II
2. Mahasiswa yang bersangkutan
3. Arsip



SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA

Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangkaraya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

KARTU KEGIATAN KONSULTASI
TUGAS AKHIR

Nama Mahasiswa : Martin
NIM : C1755201079
Tanggal Persetujuan Judul :
Judul Tugas Akhir : Penerapan Algoritma Apriori Untuk pencarian Pola Pemunggaran Buku di perpustakaan STMIK palangkaraya

No.	Tanggal Konsultasi		Uraian	Tanda Tangan
	Terima	Kembali		
1	04/02/21		x. Lengkapi Latar Belakang sesuai judul dan masalah	
			x. metode disesuaikan dgn perangkat	
			x. Sediakan gambar & tabel di jelaskan	
2			latihan BAB II dan III.	
			x. Sediakan metode perangkat lunak	
3.			x. Lengkapi semua sesuai pedoman	
4.			x. Masukkan daftar isi sesuai isi proposal dan lampiran	
			x. Acc Seminar	
5.			Perbaiki penulisan sesuai contoh.	
			Gantikan referensi maks. 10K terakhir, referensi & jurnal ilmiah	
			acc seminar	
6			Perbaiki UML	
7			Bab IV buatlah langkah 22	
			cara pembuatan aplikasi	
8			Lengkapi Lembar wawancara	
			dan uji blok box dan dokumentasi	
9			perbaiki program	
10			Lengkapi perbandingan hitungan data manual dan dengan program	



SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

STMIK PALANGKARAYA

Jl. G. Obos No. 114 ~ Telp. 0536-3224593 ~ Fax. 0536-3225515 Palangka Raya

Email: humas@stmikplk.ac.id ~ Website: www.stmikplk.ac.id

Nomor : 228/STMIK-L.21A.K/II/2021
Lampiran : -
Perihal : Permohonan Izin Penelitian dan Pengumpulan Data untuk Tugas Akhir

Kepada
Yth. (Kepala Perpustakaan STMIK Palangkaraya)
Di -
Palangkaraya

Dengan hormat,

Sehubungan dengan penyusunan Tugas Akhir mahasiswa sebagai persyaratan kelulusan Program Studi Teknik Informatika (S1) pada STMIK Palangkaraya, maka dengan ini kami sampaikan permohonan izin penelitian dan pengumpulan data bagi mahasiswa kami berikut:

Nama : MARTIN
NIM : C1755201079
Prodi (Jenjang) : Teknik Informatika (S1)
Thn. Akad. (Semester) : 2020/2021 (8)
Lama Penelitian : 18 Maret 2021 s.d 18 April 2021
Tempat Penelitian : STMIK Palangkaraya

Dengan judul Tugas Akhir:

**PENERAPAN ALGORITMA APRIORI UNTUK PENCARIAN POLA
PEMINJAMAN BUKU DI PERPUSTAKAAN
STMIK PALANGKARAYA**

Adapun ketentuan dan aturan pemberian informasi dan data yang diperlukan dalam penelitian tersebut menyesuaikan dengan ketentuan/peraturan pada instansi Bapak/Ibu.

Demikian permohonan ini disampaikan, atas perhatian dan kerja samanya diucapkan terima kasih.

Palangka Raya, 18 Maret 2021
Ketua,

Suparno, M.Kom.
NIK 196901041995105



Lampiran 4. Lembar Wawancara

Wawancara bersama narasumber Iin Wahyu Lestari (Penjaga perpustakaan STMIK Palangkaraya)

N : Narasumber

P : Penulis

P : “Selamat pagi bu, sebelumnya saya mohon izin untuk melakukan penelitian disini, dan ini surat tugas saya. Apakah bisa?”

N : “o iyaa bisa. Jadi tentang apa yaa ?”

P : “Saya, mau melakukan penelitian tentang penerapan algoritma apriori analisis pola peminjaman buku jadi saya butuh sampel data transaksi peminjaman buku yang ada di perpustakaan ini bu.

P : “Apa data daftar transaksinya ada bu” ?

N : “banyak kalo untuk data peminjaman buku bisa diliat di smart tapi data nya seperti ini, ”

P : “ooo coba saya liat bu, oh iya bu saya mau minta data peminjaman bukunya untuk file peminjamannya bisa di cetak dalam bentuk pdf ?

N : “kalo cetak sih bisa tapi yang tercetak cuman data peminjaman yang dipilih aja jadi engga bisa dicetak dari semua data transaksi.”

P : “jadi cuman harus satu satu data transaksi peminjaman yang dicetak jdi lama ya bu ?

N : “yaa lama, kalo mau kamu fotoin aja”

P : “iyaa bu saya fotoin aja, bu untuk data daftar buku yang ada di perpustakaan
ada gak bu ? ”

N : “kalo daftar buku semuanya ada di excel”

P :”saya minta file data daftar buku excelnya boleh kah bu ?”

N : “Bisa”

P : “iyaa bu makasih banyak ya”

Lampiran 5. Dokumentasi wawancara

Gambar 1. wawancara bersama kepala UPT Perpustakaan bersama



Gambar 2. melihat data peminjaman buku di perpustakaan





**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
(STMIK) PALANGKARAYA**

Jl. G. Obos No.114 Telp.0536-3224593, 3225515 Fax.0536-3225515 Palangka Raya
email : humas@stmikplk.ac.id - website : www.stmikplk.ac.id

SURAT TUGAS PENGUJI TUGAS AKHIR

No. 150/STMIK-3.C.2/AK/VII/2021

Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Palangkaraya menugaskan kepada nama-nama berikut :

1. Nama : Sulistyowati, S.Kom., M.Cs.
NIK : 198212162007002
Sebagai Ketua
2. Nama : Herkules, S.Kom., M.Cs.
NIK : 198510042010106
Sebagai Sekretaris
3. Nama : Sam'ani, S.T, M.Kom
NIK : 197703252005105
Sebagai Anggota
4. Nama : Hotmian Sitohang, M.Kom.
NIK : 198503282008002
Sebagai Anggota
5. Nama : Elok Faiqotul Himmah, S.Si., M.Sc.
NIK : 198503092009003
Sebagai Anggota

Tim Penguji Tugas Akhir Mahasiswa :

- Nama : Martin
NIM : C1755201079
Hari/ Tanggal Ujian : Rabu, 14 Juli 2021
Waktu : 08.00 WIB
Judul Tugas Akhir : Penerapan Algoritma Apriori Untuk Pencarian Pola Peminjaman Buku di Perpustakaan STMIK Palangkaraya

Demikian surat ini dibuat agar dapat dipergunakan sebagaimana mestinya dan dilaksanakan dengan penuh tanggung jawab.

Palangka Raya, 03 Juli 2021
Ketua Program Studi,

Lili Rusdiana, M.Kom.
NIK. 198707282011007

Tembusan :

1. Dosen Penguji
2. Mahasiswa yang Bersangkutan
3. Arsip



SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

STMIK PALANGKARAYA

Jl. G. Obos No. 114 ~ Telp. 0536-3224593 ~ Fax. 0536-3225515 Palangka Raya

Posel: humas@stmikplk.ac.id ~ Laman: www.stmikplk.ac.id

1.

BERITA ACARA SIDANG TUGAS AKHIR

1. Hari/Tanggal sidang : Rabu, 14 Juli 2021
2. Waktu (Jam) : 08:00 sampai 09:00 WIB
3. Nama Mahasiswa : MARTIN
4. Nomor Induk Mahasiswa : C1755201079
5. Program Studi : Teknik Informatika (S1)
6. Tahun Angkatan : 2017
7. Judul Tugas Akhir : PENERAPAN ALGORITMA APRIORI UNTUK PENCARIAN POLA PEMINJAMAN BUKU DI PERPUSTAKAAN STMIK PALANGKARAYA

8. Dosen Penguji	Nama	Nilai	Tanda Tangan
1	SULISTYOWATI		
2	HERKULES		
3	SAM'ANI		
4	HOTMIAN SITOANG		
5	ELOK FAIQOTUL HIMMAH		

9. Hasil Ujian : LULUS NILAI = 82,58

10. Catatan Penting :
1. Lama Perbaikan : hari
 2. Jika lebih dari 1 (satu) bulan dikenakan sanksi berupa denda sebesar Rp. 600.000,- (Enam ratus ribu rupiah) per bulan dari tanggal ujian
 3. Jika lebih dari 3 (tiga) bulan dari tanggal ujian maka hasil ujian dibatal-kan dan wajib mengajukan judul dan pembimbing baru



Mengetahui
Ketua Program Studi Teknik Informatika (S1)

LILIR RUSDIANA
NIK: 196707282011007

Palangka Raya, 16 Juli 2021
Ketua Penguji

(.....)
NIK :

Tembusan :

1. Arsip Prodi Teknik Informatika
 2. Mahasiswa yang bersangkutan
- Dibawa saat konsultasi perbaikan dengan dosen penguji

1. Source Code Analisis Algoritma Apriori

```
void c2(){
    jTextArea1.setText("");
    int n=0;
    float bnyAB, bnyA, bnyB=0;
    float supp,conf=0;

    for(int a=0; a<dt.size();a++){
        n++;
        for(int b=0+n;
b<dt.size();b++){

            bnyAB=sql.c2(dt.get(a).getKode_b
uku(), dt.get(b).getKode_buku());
            bnyA=dt.get(a).getNilai();

            bnyB=dt.get(b).getNilai();
            supp=bnyAB/Integer.parseInt(jT
extField1.getText())*100;

            if(supp>=Float.parseFloat(jSpinner
1.getValue().toString())){

                conf=(bnyAB/bnyA)*100;
                if(conf>=
Float.parseFloat(jSpinner2.getValu
e().toString()))

                    jTextArea1.setText(jTextArea1.
getText()+" Jika meminjam
buku \"
\"+dt.get(a).getJudul_buku()+\" \"
maka akan meminjam buku \"
\"+dt.get(b).getJudul_buku()+\" \"
dengan Supp
\"+atur.format(supp)+\"% dan
Conf \"+atur.format(conf)+\"%
\\n\");

                conf=(bnyAB/bnyB)*100;
                if(conf>=
Float.parseFloat(jSpinner2.getValu
e().toString()))

                    jTextArea1.setText(jTextArea1.
getText()+" Jika meminjam
buku \"
\"+dt.get(b).getJudul_buku()+\" \"
maka akan meminjam buku \"
\"+dt.get(a).getJudul_buku()+\" \"
dengan Supp
\"+atur.format(supp)+\"% dan
Conf \"+atur.format(conf)+\"%
\\n\");
            }
        }
    }
}

void c3(){
    int n=0,m=0,o=0;
    String item1,item2,item3="";
    float bnyABC,bnyAB=0;
    float supp,conf=0;
    for(int a=0; a<dt.size();a++){
        n++;
        for(int b=0+n
;b<dt.size();b++){
            for(int c=0+b+1;
c<dt.size();c++){

                bnyABC=sql.c3(dt.get(a).getKode_
buku(), dt.get(b).getKode_buku(),
dt.get(c).getKode_buku());

                supp=bnyABC/Integer.parseInt(jTe
extField1.getText())*100;

                if(supp>=Float.parseFloat(jSpinner
1.getValue().toString())){

                    bnyAB=sql.c2(dt.get(a).getKode_b
uku(), dt.get(b).getKode_buku());

                    conf=(bnyABC/bnyAB)*100;
                    if(conf>=
Float.parseFloat(jSpinner2.getValu
e().toString()))
```

```

jTextArea1.setText(jTextArea1.getText()+" Jika meminjam buku \"
"+dt.get(a).getJudul_buku()+" \"
dan buku \"
"+dt.get(b).getJudul_buku()+" \"
maka akan meminjam buku \"
"+dt.get(c).getJudul_buku()+" \"
dengan Supp
"+atur.format(supp)+"% dan Conf
"+atur.format(conf)+"% \n");

```

```

bnyAB=sql.c2(dt.get(a).getKode_b
uku(), dt.get(c).getKode_buku());

```

```

conf=(bnyABC/bnyAB)*100;
        if(conf>=
Float.parseFloat(jSpinner2.getValu
e().toString()))

```

```

jTextArea1.setText(jTextArea1.getText()+" Jika meminjam buku \"
"+dt.get(a).getJudul_buku()+" \"
dan buku \"
"+dt.get(c).getJudul_buku()+" \"
maka akan meminjam buku \"
"+dt.get(b).getJudul_buku()+" \"

```

dengan Supp

```

"+atur.format(supp)+"% dan Conf
"+atur.format(conf)+"% \n");

```

```

bnyAB=sql.c2(dt.get(b).getKode_b
uku(), dt.get(c).getKode_buku());

```

```

conf=(bnyABC/bnyAB)*100;
        if(conf>=
Float.parseFloat(jSpinner2.getValu
e().toString()))

```

```

jTextArea1.setText(jTextArea1.getText()+" Jika meminjam buku \"
"+dt.get(b).getJudul_buku()+" \"
dan buku \"
"+dt.get(c).getJudul_buku()+" \"
maka akan meminjam buku \"
"+dt.get(a).getJudul_buku()+" \"
dengan Supp
"+atur.format(supp)+"% dan Conf
"+atur.format(conf)+"% \n");
        }
        }
        }
        }
        }

```

2. Source Code Menu Tambah Buku

```

public class Buku extends
javax.swing.JFrame {

    /** Creates new form Buku */
    public Buku() {
        initComponents();
        dataTable();

dataFromDataBaseToComboBox();
    }

    @SuppressWarnings("unchecked")
    public String kode_lama;

```

```

        private void
btnTambahActionPerformed(java.awt
t.event.ActionEvent evt) {
        // TODO add your handling
code here:
        String kode =
txtKode.getText();
        String judul =
txtJudul.getText();
        String kategori = (String)
cbxKategori.getSelectedItem();
        String pengarang =
txtPengarang.getText();
        String penerbit =
txtPenerbit.getText();

```

```

        String tahun =
txtTahun.getText();
        try {
            Statement statement =
(Statement)

connect.GetConnection().createState
ment();
            statement.executeUpdate
                ("insert into tbl_buku
values(" +kode+", " +judul+",
"+kategori+",

```

```

        ""+pengarang+", ""+penerbit+", ""+tah
un+"");");
            dataTable();
            reset();

JOptionPane.showMessageDialog(nu
ll, "Data Berhasil Disimpan");
        }catch(Exception t){

JOptionPane.showMessageDialog(nu
ll, "Data Gagal Disimpan");
        }
    }
}

```


PENGUJIAN BLACK BOX
PENERAPAN ALGORITMA APRIORI UNTUK PENCARIAN POLA
PEMINJAMAN BUKU DI PERPUSTAKAAN
STMIK PALANGKARAYA

Nama Dosen : Deden Andriawan, M.Kom.

Gunakan tanda (√) untuk mengisi uji black box berikut

No	Fungsi yang diuji	input	Output yang diharapkan	Hasil	Output
1	Menampilkan menu utama	Mengklik aplikasi	Sistem menampilkan menu utama	Sesuai	Sistem berhasil menampilkan menu utama aplikasi
2	Menampilkan menu bar	Memilih menu bar	Sistem menampilkan sub menu	Sesuai	Sistem berhasil menampilkan sub menu yang diklik oleh user
3	Menampilkan menu daftar buku	Memilih menu bar buku dan klik sub menu buku	Sistem menampilkan seluruh data daftar buku	Sesuai	Sistem berhasil menampilkan seluruh data daftar buku
4	Menambahkan menu tambah buku	Memasukan data buku yang akan ditambahkan	Sistem menampilkan form tambah data buku yang diperlukan untuk di inputkan oleh user	Sesuai	Sistem berhasil menampilkan form tambah data buku dan berhasil menyimpan data buku ke dalam database
5	Menampilkan menu transaksi	Memilih menu bar transaksi dan klik menu transaksi	Sistem menampilkan menu daftar transaksi dan semua data transaksi	Sesuai	Sistem berhasil menampilkan menu daftar transaksi dan semua data transaksi
6.	Menambahkan data transaksi	Memasukan data transaksi yang ditambahkan	Sistem menampilkan form menu tambah transaksi	Sesuai	Sistem berhasil menampilkan form menu tambah transaksi dan menambahkan

No	Fungsi yang diuji	input	Output yang diharapkan	Hasil	Output
				Sesuai	data transaksi peminjaman ke dalam database
7	Menampilkan hasil analisis menggunakan algoritma apriori	Memasukan nilai <i>support</i> dan <i>confidence</i>	Sistem menampilkan hasil analisis	Sesuai	Sistem berhasil menampilkan hasil analisis dan ditampilkan pada menu algoritma apriori
8	Menampilkan menu tentang saya	Memilih menu bar info aplikasi dan klik sub info	Menampilkan menu tentang aplikasi	Sesuai	Berhasil menampilkan menu profil penulis

Palangkaraya 8 July 2021



Deden Andriawan, M.Kom.