

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian mengenai *data mining* algoritma apriori *association rule* bukanlah yang pertama kali dilakukan. Berikut beberapa penelitian terkait yang relevan dengan penelitian ini akan dibahas sebagai bahan rujukan.

1. Pengembangan Sistem Rekomendasi Penelusuran Buku dengan Penggalian *Association Rule* Menggunakan Algoritma Apriori (Studi Kasus Badan Perpustakaan dan Kearsipan Provinsi Jawa Timur).

Permasalahan yang terjadi di Badan Perpustakaan dan Kearsipan (BAPERSIP) Provinsi Jawa Timur adalah tingkat kesadaran masyarakat Surabaya akan minat membaca sangat kurang, hal ini bisa mempengaruhi visi dan misi BAPERSIP. Salah satu solusi yang dapat diandalkan di BAPERSIP adalah dengan adanya web intranet yang memudahkan pengguna dalam mencari koleksi buku. Namun menurut penulis solusi ini memiliki kelemahan yaitu hanya terdapat fitur untuk mencari buku, belum adanya fitur spesifik lain yang menjadi salah satu faktor pendukung peningkatan jumlah buku yang dipinjam pengguna. Menurut pendapat penulis salah satu faktor pendukung yang dapat diwujudkan adalah dengan sistem rekomendasi peminjaman buku. Sistem rekomendasi diberikan kepada anggota ketika anggota memilih dan melihat rincian dari detail buku. Sistem rekomendasi ini dikembangkan berdasar data historis peminjaman buku kemudian melihat pola asosiasi buku yang dipinjam. Pola asosiasi yang mendukung pengembangan sistem rekomendasi adalah dengan menggunakan algoritma apriori. Untuk data transaksi peminjaman buku didapatkan dari hasil join beberapa tabel yang ada kemudian menggabungkan data transaksi tiap anggota menjadi transaksi mingguan. Sehingga dihasilkan data transaksi sebanyak 45108 transaksi dan 15445 buku yang dipinjam. Dari hasil uji coba ditemukan bahwa semakin

besar *minimum support* dan *minimum confidence* semakin sedikit waktu yang dibutuhkan untuk menghasilkan rekomendasi serta semakin sedikit rekomendasi yang dibutuhkan [4].

2. Analisis Algoritma Apriori untuk Rekomendasi Penempatan Buku Pada Perpustakaan dan Arsip Daerah Salatiga.

Permasalahan yang terjadi pada perpustakaan dan arsip daerah Salatiga yaitu sistem penempatan buku belum tersusun dengan baik hal ini dikarenakan penempatan buku masih manual. Untuk mengatasi permasalahan tersebut, penulis membuat sebuah aplikasi *data mining* untuk membantu perpustakaan dalam mengatur penempatan buku. Pada penelitian ini penulis menggunakan algoritma apriori terhadap data transaksi peminjaman buku. Dengan algoritma apriori maka dapat menemukan buku-buku mana saja yang sering dipinjam sehingga dapat digunakan untuk rekomendasi penempatan buku. *Data set* awal yang digunakan yaitu data transaksi peminjaman buku pada 4 desember 2013 sampai 18 agustus 2014 sebesar 65536 data. Hasil dari uji coba sistem dengan memasukan *min support* dan *min confidence* masing-masing 0.1 dihasilkan 25 *association rule*. Permasalahan penempatan buku di perpustakaan arsip daerah Salatiga berhasil diselesaikan dengan teknik data mining algoritma apriori metode *association rule*. Dari hasil analisis kelayakan *software* menyatakan bahwa aplikasi tersebut layak diimplementasikan di perpustakaan arsip dan daerah Salatiga [5].

3. Penerapan Data Mining *Association Rule* Menggunakan Algoritma Apriori untuk Meningkatkan Penjualan dan Memberikan Rekomendasi Pemasaran Produk Speedy (Studi Kasus PT Telkom Cabang Wonogiri).

Penelitian ini difokuskan untuk menentukan pola pemasaran pada produk speedy di PT Telkom Cabang Wonogiri agar lebih efisien dan unggul. Berdasarkan hasil observasi yang dilakukan penulis bahwa ditiap kecamatan yang menjadi target pemasaran belum semuanya menggunakan produk speedy. Hal ini menuntut manajer penjualan di PT Telkom harus menentukan pola strategi yang efisien. Untuk mengatasi masalah tersebut penulis menganalisa pola data hasil penjualan speedy dengan menggunakan *data*

mining association rule menggunakan algoritma apriori. Data transaksi penjualan yang digunakan penulis sebanyak 12 transaksi. Atribut kecamatan ini digunakan penulis untuk mengasosiasikan kecamatan mana saja yang belum menggunakan produk speedy. Delapan kecamatan diantaranya Wonogiri, Selogiri, Jatisrono, Manyaran, Baturetno, Slogohimo, Wuryantoro, dan Puwantoro. Penelitian tersebut menghasilkan tiga *association rule* dengan nilai *minimum support* 50% dan *minimum confidence* 100% yang dapat dilihat pada tabel dibawah ini [6] :

Tabel 2.1 Hasil Penelitian

| Aturan | Support | Confidence |
|---|----------------|-------------------|
| Jika ada yang menggunakan produk di kecamatan Selogiri, maka ada juga yang menggunakan produk di kecamatan Wonogiri. | 50% | 100% |
| Jika ada yang menggunakan produk di kecamatan Jatisrono, maka ada juga yang menggunakan produk di kecamatan Wonogiri. | 50% | 100% |
| Jika ada yang menggunakan produk di kecamatan Baturetno, maka ada juga yang menggunakan produk di kecamatan Wonogiri | 66.67% | 100% |
| Jika ada yang menggunakan produk di kecamatan Wonogiri, maka ada juga yang menggunakan produk di kecamatan Baturetno | 66.67% | 66.67% |

Association rule yang dihasilkan dapat digunakan untuk mengetahui kecamatan mana yang belum menggunakan produk speedy dan dapat digunakan untuk meningkatkan penjualan produk.

Tabel 2.2 Penelitian Terkait

| No | Penulis dan Tahun | Permasalahan | Metode | Hasil Penelitian |
|----|---|--|-----------------------------------|--|
| 1 | Uma Mazida, 2015 | Penempatan buku belum tersusun dengan baik hal ini dikarenakan belum adanya suatu sistem yang membantu dalam rekomendasi penempatan buku | Algoritma apriori. | Dihasilkan suatu aplikasi yang dapat merekomendasikan penempatan buku berdasarkan data transaksi peminjaman. |
| 2 | Mochamad Yusuf Pratama 2014 | Produk speedy masih belum terpasarkan dengan baik, karena masih terdapat beberapa kecamatan yang belum menggunakan produk speedy | <i>Association Rule</i> , apriori | Dihasilkan pola penjualan yang dapat digunakan dalam menentukan pola strategi pemasaran di kecamatan yang masih belum menggunakan produk speedy. |
| 3 | Nugroho Wandi, Rully A. Hendrawan, Ahmad Mukhlason, | Minat membaca masyarakat di kota Surabaya kurang sehingga Badan Perpustakaan dan Kearsipan Provinsi Jawa Timur perlu | <i>Association Rule</i> , apriori | Semakin besar <i>minimum support</i> dan <i>minimum confidence</i> semakin sedikit waktu yang dibutuhkan untuk menghasilkan |

| No | Penulis dan Tahun | Permasalahan | Metode | Hasil Penelitian |
|----|-------------------|---|--------|--|
| | 2013 | melakukan perbaikan pada sistem perpsutakaan. | | rekomendasi serta semakin sedikit rekomendasi yang dibutuhkan. |

Dari ketiga penelitian terkait di atas algoritma apriori digunakan untuk menghasilkan *association rule*. Dimana *association rule* ini menjadi tujuan atas permasalahan dari ketiga penelitian terkait di atas. Parameter umum yang digunakan dalam penelitian terkait di atas yaitu parameter *support* dan *confidence*. Kedua parameter tersebut diberikan oleh pengguna sebagai masukan untuk menyeleksi *association rule* yang kuat.

2.2 Data Mining

Data mining yaitu sekumpulan proses yang memperkerjakan (*machine learning*) atau teknik pembelajaran yang digunakan untuk analisis dan untuk mengekstrak pengetahuan. *Data mining* bertugas melakukan pencarian suatu trend atau pola yang diinginkan di dalam suatu basis data yang besar berguna untuk membantu dalam pengambilan keputusan. Pola-pola tersebut dapat dikenali oleh suatu perangkat tertentu yang dapat memberikan suatu analisa data yang berwawasan dan bisa dipelajari lebih teliti. Dari sudut pandang komersil *data mining* diperlukan karena banyaknya volume data dalam data *warehouse* seperti data *web e-commerce*, penjualan di *departement store*, persaingan yang kompetitif untuk dapat menyediakan layanan yang lebih baik. Sedangkan dari sudut pandang keilmuan yaitu kecepatan data yang dihimpun dan disimpan seperti pada satelit, teleskop untuk memindai langit dan simulasi saintifik yang dapat membangkitkan data dalam ukuran terabytes. *Data mining* berawal mengambil ide dari AI, pengenalan pola statistik, dan *database system* [7].

2.3 Operasi Data Mining

Operasi *data mining* menurut sifatnya dibedakan menjadi dua, yaitu bersifat prediksi (*prediction driven*) untuk menjawab pertanyaan apa dan sesuatu yang bersifat remang-remang atau transparan. Operasi prediksi digunakan untuk validasi hipotesis, *querying*, dan pelaporan (misal : *spreadsheet* dan *pivot table*), analisis multidimensi (*dimensional summary*), OLAP (*Online Analytical Processing*) serta analisis statistik. Dan yang bersifat penemuan (*discovery driven*) bersifat transparan untuk menjawab pertanyaan “mengapa?”. Operasi penemuan digunakan untuk analisis data eksplorasi, pemodelan prediktif, segmentasi *database*, analisis keterkaitan (*link analysis*) dan deteksi deviasi [7].

2.4 Permasalahan Data Mining

Sistem *data mining* berdasar pada basis data yang menyediakan data mentah dan hal ini memunculkan permasalahan dalam basis data yang cenderung dinamis, tidak lengkap, ber-*noise*, dan besar. Permasalahan lain muncul sebagai akibat dari kecukupan dan relevansi dari informasi yang disimpan [7]. Basis data seringkali didesain untuk tujuan yang berbeda dari *data mining* dan kadang kala *property* atau atribut yang akan menyederhanakan pekerjaan pembelajaran tidak tersedia atau tidak dapat dimintai dari dunia nyata. Data yang tidak meyakinkan menyebabkan permasalahan karena jika ada atribut esensial bagi pengetahuan tentang domain aplikasi tidak ada dalam data tidak memungkinkan untuk menemukan pengetahuan yang dapat mengenai domain yang diberikan, sebagai contoh tidak dapat mendiagnosa malaria dari basis data pasien jika basis data tersebut tidak mengandung jumlah sel darah merah pasien [7]. Data yang hilang dapat dibenahi dengan sistem penemuan dalam berbagai cara, seperti :

1. Secara sederhana dengan mengabaikan nilai-nilai yang hilang.
2. Menghilangkan *record* yang berhubungan.
3. Menebak nilai yang hilang dari nilai-nilai yang diketahui.
4. Memperlakukan data yang hilang sebagai sebuah nilai khusus yang dimasukkan sebagai tambahan dalam domain atribut.
5. Atau menghitung rata-rata nilai yang hilang menggunakan teknik *bayesian*.

Data yang mengandung *noise* dalam pengertian menjadi tidak teliti merupakan karakteristik dari semua koleksi data dan secara khusus cocok untuk sebuah distribusi statistik dari semua koleksi data dan secara khusus cocok untuk sebuah distribusi statistik biasa seperti *gaussian* saat nilai-nilai yang salah merupakan kesalahan masukan data. Ketidakjelasan (*uncertainty*) menunjuk kepada kepelikan *error* dan tingkat *noise* dalam data. Presisi data merupakan salah satu pertimbangan penting dalam sistem penemuan. Basis data cenderung menjadi besar dan dinamis dalam hal isinya yang selalu berubah saat informasi ditambahkan, dimodifikasi, atau dihapus. Permasalahan hal ini dari sudut pandang *data mining* adalah bagaimana menjamin bahwa aturan-aturan tersebut *up to date* dan konsisten dengan informasi paling terkini juga sistem pembelajaran mempunyai *time sensitive* saat beberapa nilai data berubah terhadap waktu dan sistem penemuan dipengaruhi oleh ketepatan waktu dari data tersebut [7].

2.5 Teknik Data Mining

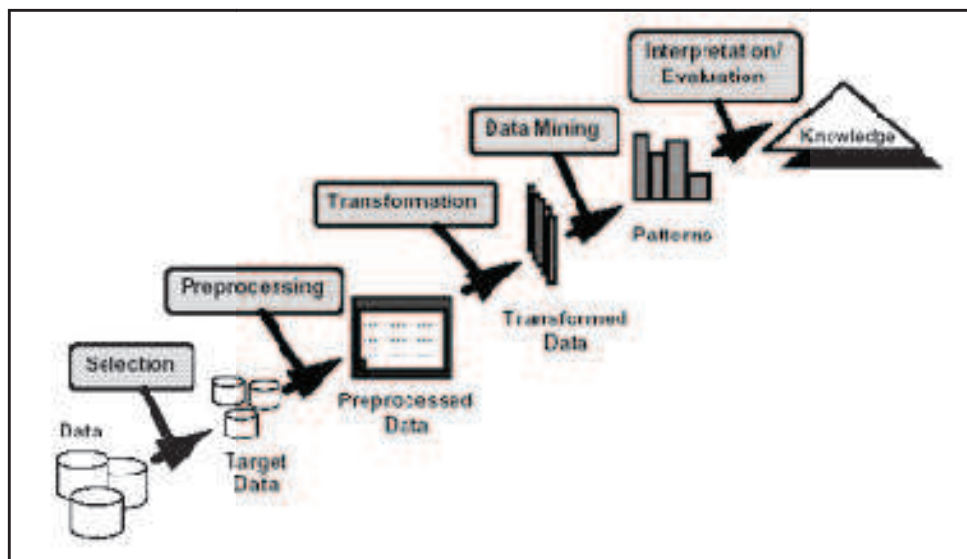
Beberapa teknik *data mining* dalam [7] adalah sebagai berikut :

1. Klasifikasi (*Classification*)
Klasifikasi atau *supervised learning* digunakan untuk menentukan sebuah *record* data baru ke beberapa klas yang telah terdefinisi sebelumnya.
2. Regresi
Teknik ini digunakan untuk memprediksi dengan suatu variabel kontinyu yang didasarkan pada variabel lain dengan asumsi dari model linier dan non linier. Penerapan teknik ini untuk jaringan saraf tiruan dan statistika.
3. Klasterisasi (*Clustering*)
Clustering atau *unsupervised learning* dengan membagi *data set* ke dalam *subset* sehingga elemen dari *subset* tersebut mempunyai *set-property* yang dibagikan bersama dan memiliki kemiripan yang tinggi dan rendah dalam suatu *subset*.
4. *Association Rule*
Asosiasi ini digunakan untuk mendeteksi sekumpulan atribut yang muncul secara bersamaan dalam frekuensi yang sering muncul. Contoh 50% yang

sering berbelanja disuatu supermarket yang membeli kopi dan gula, dan 40% yang membeli kedua-duanya.

2.6 Proses Data Mining

Tahapan proses *data mining* merupakan proses *Knowledge Discovery in Databases* (KDD).



Gambar 2.1 Tahap-Tahap *Knowledge Discovery in Databases*

Secara garis besari tahapan *Knowledge Discovery in Databases* dapat dijabarkan sebagai berikut [7] :

1. Menggali *knowledge* dan mengetahui tujuan yang hendak dicapai oleh pengguna terhadap domain aplikasi.
2. Menentukan target *data set* seperti pemilihan data dan fokus terhadap *sub set data*.
3. *Cleaning* dan *transformation* data seperti menghilangkan derau, dan nilai yang hilang.
4. Memilih algoritma yang akan digunakan seperti *association*, *classification*, *clustering*.
5. *Evaluation* atau *Interpretation* menghasilkan suatu pola yang baru dan menarik jika perlu dilakukan iterasi.

2.7 Algoritma Apriori

Algoritma apriori termasuk dalam *association rule*. Selain apriori diantaranya yaitu algoritma *hash based* dan *generalized rule*. Algoritma apriori merupakan algoritma yang digunakan untuk mencari pola frekuensi tinggi. Pola frekuensi tinggi yaitu pola-pola item yang disebut *minimum support* atau *threshold* dimana nilai ini mempunyai frekuensi di atas ambang batas tertentu. Pola frekuensi tinggi ini digunakan untuk membentuk *association rule*. Algoritma apriori terbagi dalam beberapa tahapan atau disebut sebagai iterasi. setiap iterasi menghasilkan pola frekuensi tinggi dan panjang yang sama. Pertama menghasilkan pola frekuensi tinggi dengan panjang yang satu. Pada iterasi pertama ini setiap *support* dihitung dengan *scanning database*. Setelah *item* yang mempunyai nilai *support* didapatkan kemudian *support* tersebut dipangkas dengan memasukan *minimum support*. *Minimum support* dipilih sebagai pola frekuensi tinggi atau disingkat 1-*itemset*. Sedangkan k-*itemset* berarti satu set yang terdiri dari k-*item*. Iterasi kedua dengan menghasilkan 2-*itemset* yang tiap *set* memiliki dua *item*. Pertama dibuat *candidate 2-itemsets*, setelah *support* dari *candidate 2-itemsets* didapat, *candidate 2-itemsets* dipangkas dengan *minimum support* yang telah dipilih. kemudian 2-*itemsets* yang telah dipangkas adalah pola frekuensi tinggi dengan panjang 2. Iterasi ke-k dibagi menjadi beberapa tahap [8] :

1. Membentuk *Candidate itemset*
Candidate k-itemset terbentuk dari pasangan (k-1)-*itemset* yang didapatkan pada iterasi sebelumnya. Ciri dari apriori yaitu adanya pemangkasan *candidate* terhadap k-*itemset* dimana *subset*-nya berisi k-q *item* dan tidak termasuk dalam pola frekuensi tinggi dengan panjang k-1.
2. Perhitungan *support* dari setiap *candidate k-itemset* didapatkan dengan *scanning database* untuk menghitung total transaksi yang memuat semua *item* didalam *candidate k-itemset* tersebut. Hal ini merupakan ciri dari algoritma apriori dimana memerlukan penghitungan dengan *scanning* seluruh *database* sebanyak k-*itemset* terpanjang.
3. Menetapkan pola frekuensi tertinggi

Pola frekuensi tinggi yang memuat *k-itemset* ditetapkan dari *candidate k-itemset* yang nilai *support* lebih besar daripada *minimum support*. Bila tidak mendapatkan pola frekuensi tinggi maka seluruh proses dihentikan. Namun apabila tidak, *k* ditambahkan satu dan kembali ke bagian satu. Tahapan asosiasi terbagi menjadi dua tahap [8]:

- a. Analisis pola frekuensi tertinggi.

Tahapan ini mencari pasang *item* yang memenuhi *minimum support*. Nilai *support* sebuah *item* diformulasikan sebagai berikut :

$$Support (A) = \frac{Total\ Transaksi\ Mengandung\ A}{\epsilon\ Transaksi} \quad (2.1)$$

Nilai *support* untuk 2 *item* diformulasikan sebagai berikut :

$$Support (A \cup B) = \frac{Total\ Transaksi\ Mengandung\ A\ dan\ B}{\epsilon\ Transaksi} \quad (2.2)$$

- b. Pembentukan Aturan Asosiasi

Pola frekuensi tinggi yang telah terbentuk, maka mencari *association rule* yang memenuhi minimum untuk nilai *confidence* dengan perhitungan $A \rightarrow B$.

$$Confidence\ P(B|A) = \frac{Total\ Transaksi\ Mengandung\ A\ dan\ B}{\epsilon\ A} \quad (2.3)$$

2.8 Parameter Correlation

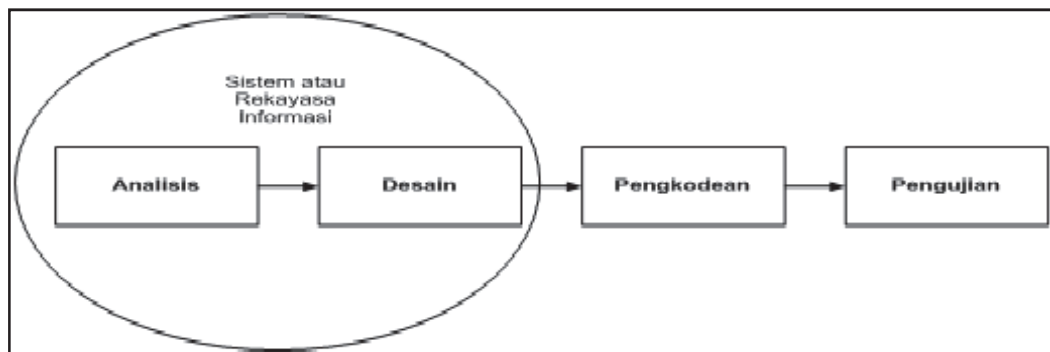
Setelah diketahui bahwa *association rule* terbagi dalam dua tahap yaitu analisa pola frekuensi tertinggi atau *frequent itemsets* dan pembentukan *association rule* yaitu dengan menghasilkan nilai *support* dan nilai *confidence*, maka *association rule* yang terbentuk akan dianalisis apakah *item A* akan dibeli bersamaan dengan *item B*. Oleh karena itu, untuk mengetahui hubungan tersebut maka dapat dilihat dari nilai *lift*. *Lift* yaitu parameter penting selain *support* dan *confidence* dalam *association rule*. Parameter ini yang disebut sebagai parameter *correlation*. Menurut J Han dan M Kamber yang dikutip oleh [9] dalam menentukan nilai *lift* menggunakan rumus sebagai berikut :

$$Lift (A,B) = \frac{P(A \cup B)}{P(A) \times P(B)} \quad (2.4)$$

Apabila nilai *lift* yang dihasilkan lebih dari satu maka *item A* dan *item B* benar-benar dibeli secara bersamaan maka *association rule* tersebut dapat dikatakan *valid*. Menurut J Han dan M Kamber apabila nilai *lift* kurang dari satu maka keterkaitan *item A* dan *item B* *negatively correlation* dan apabila nilai *lift* sama dengan satu maka keterkaitan *item A* dan *item B* *independent*.

2.9 Metode Pengembangan Sistem *Waterfall*

Metode *waterfall* ini sering disebut sebagai *sequence linier* yang menyediakan pendekatan alur hidup perangkat lunak atau sistem secara sekuensial. Pengembangan sistem metode *waterfall* menurut [10] dibagi menjadi empat tahap yaitu analisa, desain, pengkodean, pengujian, dan tambahan yaitu *support* atau pemeliharaan yaitu sebuah perangkat lunak yang sudah dibuat dapat berubah. Perubahan ini bisa muncul karena kesalahan yang tidak terdeteksi pada tahap *testing* atau sistem berada pada lingkungan yang baru. Berikut ilustrasi *waterfall*:



Gambar 2.2 Ilustrasi *Waterfall*

1. Analisis

Analisis yang dimaksudkan adalah analisis kebutuhan perangkat lunak, yaitu proses untuk melakukan pengumpulan kebutuhan perangkat lunak agar perangkat lunak mudah dipahami dan sesuai dengan kebutuhan *user*. Adanya spesifikasi kebutuhan bisa berupa kebutuhan fungsional dan kebutuhan nonfungsional.

2. Desain

Pada tahapan desain dapat dimulai dengan membuat rancangan program perangkat lunak yang di dalamnya terdapat struktur data, representasi perangkat lunak, dan prosedur pengkodean.

3. Pengkodean

Desain yang telah selesai dibuat, harus ditranslasikan atau ditransformasi ke dalam program. Hasilnya berupa program komputer yang sesuai dengan desain yang telah dibuat.

4. Pengujian

Pengujian perangkat lunak difokuskan pada segi logik dan fungsional. Pengujian dilakukan untuk meminimalisir *error* dan untuk memastikan bahwa *input* dan *output* sesuai yang diharapkan. Dalam penelitian ini penulis menggunakan pengujian metode black box yaitu menguji perangkat lunak dari spesifikasi fungsional tanpa menguji desain dan kode program.

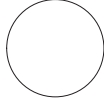



5. Pemeliharaan

Tahapan ini bisa terjadi apabila terdapat kesalahan yang tidak terdeteksi selama tahapan pengujian atau bisa terjadi karena perubahan perangkat lunak terhadap lingkungan baru.

2.10 *Data Flow Diagram*

Pertama kali DFD ini dikembangkan pada tahun 1979 oleh Trish Sarson dan Chris Gane yang mengembangkan sistem berbasis dekomposisi fungsional. Dekomposisi fungsional ini berupa DFD. Pada tahun 1980 Edward Yourdon dan Tom DeMarco memperkenalkan metode lain yaitu dengan mengubah persegi menjadi sudut pandang lengkung. DFD ini populer sebagai model analisis perangkat lunak. Berikut notasi-notasi DFD yang dikembangkan oleh Edward Yourdon dan Tom DeMarco [10]:


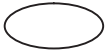
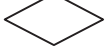


Tabel 2.3 Simbol-simbol Data Flow Document

| Simbol | Keterangan |
|---|----------------------------------|
|  | Proses atau fungsi atau prosedur |
|  | Basis data atau <i>file</i> |
|  | Entitas luar |
|  | Aliran data |

2.11 Entity Relationship Diagram

Model entity relationship terdiri dari komponen-komponen himpunan entitas dan himpunan relasi. Dimana atribut dalam relasi mempresentasikan seluruh fakta dari dunia nyata dan dapat digambarkan secara sistematis dengan *entity relationship diagram* (ERD). Berikut notasi-notasi untuk menggambarkan ERD [11] :

Tabel 2.4 Notasi ERD

| Notasi | Keterangan |
|---|------------------------|
|  | Himpunan entitas |
|  | Atribut |
|  | Himpunan relasi |
|  | <i>Link</i> |
|  | Himpunan entitas lemah |

Penggambaran relasi antara himpunan entitas dengan himpunan relasi dilengkapi dengan kardinalitas relasi. Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain. Terdapat

tiga jenis kardinalitas relasi yaitu relasi *one to one* (1 to 1), relasi *one to many* (1 to N), dan relasi *many to many* (N to N).

2.12 Java

Java disebut sebagai teknologi pemrograman yang dikembangkan oleh perusahaan Sun Microsystem. Teknologi java dipergunakan untuk membuat aplikasi berbasis jaringan, *database*, dekstop, *web* maupun grafis. Menurut [12] java merupakan pemrograman yang handal. Kehandalan ini terbukti dari slogannya yaitu “*write once run everywhere*” yang mana penulisan kode program dalam bahasa java dapat dijalankan pada berbagai sistem operasi dan berbagai *platform* tanpa tergantung pada arsitektur komputer yang digunakan. Meskipun demikian harus ada mesin penerjemah bahasa java yang disebut dengan *java virtual machine*. *Java virtual machine* ini dapat diterapkan pada berbagai sistem operasi dan *platform*. Hal ini membuat program java dapat dengan mudah dijalankan pada komputer bersistem operasi apapun tanpa harus mengubah kode program yang telah ditulis.

2.13 Netbeans

Netbeans merupakan salah satu IDE yang dikembangkan menggunakan bahasa pemrograman java. Netbeans sendiri ruang lingkupnya dapat terintegrasi dalam suatu *software* yang didalamnya terdapat pemrograman berbasis GUI, *compiler*, *interpreter*, maupun *text editor*. Untuk komponen GUI, pada netbeans telah tersedia. Komponen GUI membantu mempermudah dalam merancang dan membangun *form* dengan java. Komponen GUI tersebut yaitu GUI *builder*, *inspector windows*, *pallete windows*, *property windows*, *design area*, dan *source code* [12].

2.14 SPMF (*Sequence Pattern Mining Framework*)

SPMF adalah sebuah aplikasi perangkat lunak *open source* yang berfungsi untuk melakukan pengolahan *data mining* yang menyediakan *library data mining* dan ditulis dalam pemrograman java. SPMF itu sendiri didistribusikan dibawah lisensi

dari GPL v3 yang telah mengimplementasikan 115 algoritma untuk *association rule mining* dan *sequential pattern mining* [13].

2.15 MySQL

MySQL merupakan *database* yang mendukung berbagai macam pemrograman *script* untuk aplikasi *web* maupun aplikasi berbasis *desktop*. Sejarah MySQL pertama kali dikembangkan oleh perusahaan bernama MySQL AB yang merupakan perusahaan asal Swedia pada tahun 1994-1995. Pada mulanya sebelum menjadi MySQL AB perusahaan tersebut bernama TCX yang merupakan perusahaan yang bergerak dibidang konsultan *database* dan pengembang perangkat lunak. Perkembangan saat ini MySQL dibawai atau diakuisisi oleh perusahaan Oracle. Kepopuleran yang dimiliki MySQL yaitu menggunakan bahasa SQL sebagai *language basic* untuk mengakses *database* selain itu juga didukung dengan kinerja *query* yang cepat dan dapat digunakan oleh perusahaan skala menengah kecil serta bersifat *open source* [14].

2.16 Metode Penentuan Sampel Slovin

Metode penentuan sampel dengan Slovin, memberikan kesempatan kepada pemakai untuk menentukan nilai galat atau tingkat *error* yang bisa ditolelir dalam penentuan sampel. Berikut rumus Slovin [15] :

$$n = \frac{N}{1+N.e^2} \quad (2.5)$$

N = ukuran populasi, n = ukuran sampel, e = galat atau tingkat *error* (%).