

LAMPIRAN F

DAFTAR RIWAYAT HIDUP

Nama : Istiana
Tempat Lahir : Pekanbaru
Tanggal Lahir : 18 Agustus 1987
Jenis Kelamin : Perempuan
Agama : Islam
Alamat : Jl. Tapah No.30, Pekanbaru-Riau 28282
Anak : Ke-6 (dua) dari 6 (dua) bersaudara
Nama Orang Tua
Ayah : Zainuddin
Ibu : Nurhayati
Alamat Orang Tua : Jl. Tapah No.30, Pekanbaru-Riau 28282
Email : yellow_zha@yahoo.com
Jenjang Pendidikan :
1. Tahun 1994-2000 : Sekolah Dasar (SD) Negeri 015 Marpoyan Damai Pekanbaru
2. Tahun 2000-2003 : Diniyyah Menengah pertama (DMP) Perguruan diniyyah putri Padang panjang- Sumatra Barat
3. Tahun 2003-2006 : Madrasah Aliyah Negeri 2 Model Pekanbaru
4. Tahun 2006-2011 : Universitas Islam Negeri (UIN) Sultan Syarif Kasim Riau Pekanbaru
Fakultas Sains dan Teknologi
Jurusan Teknik Informatika

RANCANG BANGUN APLIKASI
DETEKSI KECURANGAN ABSENSI KARYAWAN
MENGGUNAKAN *DATA MINING ASSOCIATION RULE*

ISTIANA
10651004301

Tanggal Sidang : 7 Maret 2011

Periode Wisuda : Juni 2011

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. Soebrantas No.155 Pekanbaru

ABSTRAK

Data mining merupakan teknik untuk menemukan pola yang tersembunyi dari sekumpulan data yang tersimpan dalam sebuah database atau tempat penyimpanan data lainnya. Teknik pencarian pola dalam data mining yaitu menggunakan metode *Association rule*. Teknik ini bertujuan mencari item yang sering muncul secara bersamaan atau berdekatan. Salah satu contoh penerapan *association rule* yaitu dalam penemuan pola kebiasaan absensi karyawan untuk mendeteksi kecurangan absensi. Penelitian ini bertujuan untuk mengaplikasikan penerapan struktur penyimpanan data pohon *FP-Tree (Frequent Pattern Tree)* yang digunakan bersamaan dengan algoritma *FP-Growth* untuk menemukan *frekuent itemset* (data yang paling sering muncul) dari sebuah database absensi. Dari hasil pengujian struktur penyimpanan data *FP-Tree* dan algoritma *FP-Growth* telah berhasil dimanfaatkan untuk menggali pola kebiasaan absensi dengan hasil perhitungan frekuensi kemunculan karyawan. Akan tetapi *rule* yang dihasilkan dalam pencarian itemset sangat tergantung dari batasan nilai frekuensi. Hal ini menyebabkan jika batasan frekuensi ditingkatkan memungkinkan itemset yang tidak memenuhi batas nilai tidak terdeteksi ataupun sebaliknya.

Kata kunci : *Absensi, Association Rule, Data Mining, FP-Growth, FP-Tree*

APLICATION DESIGN AND DEVELOPMENT
ATTENDANCE CHEATING DETECTION EMPLOYESS
USING DATA MINING ASSOCIATION RULE

ISTIANA
10651004301

Date of Final Exam : March 7th 2011
Graduate Ceremony Period : Juny 2011

Informastic Technology Engineering Departement
Faculty of Sciences and Technology
State Islamic University of Sultan Syarif Kasim Riau
Soebrantas street No.155 Pekanbaru

ABSTRACT

Data mining is a technique to discover the hidden pattern from a collection of data stored in a database or other data storage. Technique finding pattern in data mining is using association rule. One example the implementation of association rule that is in the finding pattern attendance to detect cheating attendance. The research destination to applying the implementation of the data storage structure in the FP-Tree (Frequent Pattern Tree) are used together with FP-Growth algorithm (Frequent Pattern Growth) to find the frequency itemset (data which often appears) from a database attendance. From the testing data storage, structur FP-Tree algorithm FP-Growth has been successfully advantage the explore pattern attendance with the calculation result frequency of occurrence of employee. However, the rules resulting in the itemset search highly depends on the limit value of frequency. This caused if restrictions improved frequency allow itemset who do not meet the limit values are not detected or vice versa.

Keywords : Association Rule, Attendance, Data Mining, FP-Growth, FP-Tree

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN	v
PERSEMBAHAN.....	vi
ABSTRAK.....	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvii
DAFTAR LAMPIRAN.....	xix
BAB I PENDAHULUAN	
1.1 Latar Belakang	I-1
1.2 Rumusan Masalah	I-2
1.3 Batasan Masalah	I-2
1.4 Tujuan Penelitian	I-3
1.5 Sistematika Penulisan	I-3
1.6 Rencana Kerja	1-5
BAB II LANDASAN TEORI	
2.1 <i>Knowledge Discovery in Databases (KDD)</i>	II-1
2.2 Konsep <i>Data Mining</i>	II-5
2.3 Fungsi <i>Data Mining</i>	II-7
2.3.1	Fungsi
Prediktif.....	II-7
2.3.2 Fungsi Deskripsi	II-2
2.4 <i>Association rule mining</i>	II-10
2.4.1 Ukuran <i>Association rule mining</i>	II-10

2.4.2	Klasifikasi <i>Association rule mining</i>	II-12
2.5	<i>Discretization Histogram Analysis</i>	II-14
2.6	Pembangunan <i>Frequent Patern Tree (FP-Tree)</i>	II-15
2.7	Algoritma <i>Frequent Patern growth (FP-Growth)</i>	II-16
2.8	<i>Devide and Conquer</i>	II-19
BAB III METODOLOGI PENELITIAN		
3.1	Perumusan Masalah	III-2
3.2	Studi Literatur	III-2
3.3 Analisa	
	Sistem	III-2
3.3.1	Analisa Subsistem Model (<i>Model Association Rule</i>)	III-2
3.3.2	Analisa Subsistem Dialog	III-4
3.3.3	Analisa Subsistem Data	III-4
3.4	Perancangan	III-4
3.4.1	Perancangan Basis Data	III-4
3.4.2	Perancangan Struktur Menu	III-4
3.4.3	Perancangan Antar Muka (<i>Interface</i>)	III-4
3.5	Implementasi	III-5
3.5	Pengujian	III-5
3.6	Kesimpulan Akhir	III-5
BAB IV ANALISA DAN PERANCANGAN		
4.1	Analisa Sistem	IV-1
4.2	Menghitung frekuensi kemunculan itemset kecurangan absensi pegawai menggunakan algoritma <i>FP-Growth Dalam Knowledge Discovery in Database (KDD)</i>	IV-2
4.2.1	<i>Data Selection</i>	IV-2
4.2.2	<i>Pre-processing</i>	IV-4
4.2.3	<i>Transformation</i>	IV-7
4.2.4	Data mining	IV-11
4.2.4.1	Tahap pembangunan <i>FP-tree</i>	IV-11
4.2.4.2	Penerapan algoritma <i>FP-Growth</i>	IV-28

4.2.5 Interpretation / Evaluation.....	IV-32
4.3 Perancangan Sistem	IV-32
4.3.1 Flowchart Sistem	IV-33
4.3.2 Diagram Konteks (<i>Context Diagram</i>).....	IV-34
4.3.3 Data Flow Diagram (DFD)	IV-35
4.3.3.1 DFD Level 2 Proses 1 Login	IV-37
4.3.3.2 DFD Level 2 Proses 2karyawan	IV-38
4.3.3.3 DFD Level 2 Proses 3 Absensi	IV-39
4.3.3.4 DFD Level2 Proses4 Deteksi Kecurangan Absensi	IV-40
4.3.4 Entity Relationship Diagram (ERD).....	IV-42
4.3.5 Perancangan Tabel Basis Data.....	IV-43
4.3.5.1 Rancangan Tabel Data Login.....	IV-43
4.3.5.2 Rancangan Tabel Data Karyawan.....	IV-43
4.3.5.3 Rancangan Tabel Data Absensi	IV-44
4.3.5.4 Rancangan Tabel Kelompok.....	IV-44
4.3.5.5 Rancangan Tabel Frekuensi Item.....	IV-45
4.3.5.6 Rancangan Tabel <i>FP-Tree</i>	IV-45
4.3.5.7 Rancang Tabel <i>FPGrowth</i>	IV-46
4.4 Perancangan Subsistem Dialog.....	IV-46
4.4.1 Struktur Menu	IV-47
4.4.2 User Interface	IV-47
4.4.2.1 Rancangan menu <i>Login</i>	IV-47
4.4.2.2 Rancangan menu utama	IV-48
4.4.2.3 Rancangan Form Data <i>Login</i>	IV-48
4.4.2.4 Rancangan Form Karyawan.....	IV-49
4.4.2.5 Rancangan form absensi	IV-49
4.4.2.6 Rancangan Form Deteksi Kecurangan Absensi...	IV-50
4.4.2.7 Rancangan Form Laporan Deteksi Kecurangan Absensi.....	IV-52

BAB V IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi.....	V-1
5.1.1 Batasan Implementasi	V-1
5.2 Implementasi Sistem.....	V-2
5.2.1 Menu <i>Login</i>	V-2
5.2.2 Menu Utama.....	V-3
5.2.3 Form Data Karyawan.....	V-4
5.2.4 Fom Data Absensi.....	V-5
5.2.5 Form Deteksi Kecurangan Absensi.....	V-5
5.2.6 Form Laporan Kecurangan Absensi	V-9
5.2.7 Form Data Login.....	V-11
5.3 Pengujian.....	V-11
5.3.1 Pengujian Sistem.....	V-12
5.3.1.1 Pengujian Menu Login.....	V-12
5.3.1.2 Pengujian Menu Utama.....	V-13
5.3.1.3 Pengujian Menu Data Karyawan	V-14
5.3.1.4 Pengujian Menu Data Absensi.....	V-16
5.3.1.5 Pengujian Menu deteksi kecurangan absensi.....	V-18
5.3.1.6 Pengujian Form Data Login.....	V-19
5.3.2 Pengujian Peubah Paramete Data	V-19
5.4 Kesimpulan Pengujian	V-21

BAB VI PENUTUP

6.1 Kesimpulan	VI-1
6.2 Saran	VI-1

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

BAB I

PENDAHULUAN

1.1 Latar Belakang

Data mining merupakan proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Tujuan dari data mining adalah menemukan hubungan-hubungan atau pola-pola yang mungkin memberikan indikasi yang bermanfaat. Salah satu teknik data mining yaitu *Association rule mining*.

Association rule mining adalah suatu prosedur untuk mencari hubungan antar *item* dalam suatu *data set* yang ditentukan. *Association rule mining* menemukan hubungan atau pola dari suatu kumpulan data berdasarkan perhitungan frekuensi kemunculan data. Salah satu contoh penerapan *association rule* yaitu dalam penemuan pola kebiasaan absensi karyawan untuk mendeteksi kecurangan absensi karyawan.

Deteksi kecurangan absensi telah pernah di teliti oleh Gregorius Setia Budhi dengan menggali pola kebiasaan absensi (*Checkclock*) pegawai sebuah perusahaan dengan menganalisa *database* absensi elektronik secara otomatis, guna mendeteksi dan menemukan informasi tentang karyawan – karyawan yang seringkali melakukan absensi dalam waktu yang sangat berdekatan. Asumsi dari pemikiran ini adalah bila pada data absensi ditemukan pola / *pattern* dengan frekuensi cukup tinggi tentang dua atau lebih karyawan yang absen dalam waktu yang hampir bersamaan dalam kurun waktu tertentu, besar kemungkinan dalam realitasnya, karyawan yang melakukan absen hanya satu orang sedangkan lainnya "titip absen" kepada karyawan tersebut. Proses pencarian dilakukan dengan menggunakan algoritma *Pincer Search*. Algoritma ini digunakan untuk menentukan himpunan data yang sering muncul (*Frequent itemset*) dalam kumpulan sebuah data. Dalam hasil pengujiannya, algoritma *pincers search* masih mengalami permasalahan dalam mempercepat proses generasi *frequent itemset*. Dikarenakan algoritma *pincers search* memerlukan langkah *candidate generation*,

yaitu dengan melakukan *scanning database* yang berulang-ulang dalam pencarian frekuensi itemset. (Setia Budhi, 2010).

Dalam tugas akhir ini, penulis akan membangun kembali sistem deteksi kecurangan absensi karyawan dengan penerapan algoritma yang berbeda pada metode *Association Rule*. Algoritma yang digunakan yaitu *Frequent item Growth (FP-Growth)*. Algoritma ini dipilih dikarenakan dapat mengoptimalkan penentuan pencarian *frequent itemset* yang lebih ringkas. (Samuel, 2008). Di harapkan dengan algoritma ini akan dapat menutupi kekurangan pada penelitian sebelumnya.

1.2 Rumusan Masalah

Dari latar belakang yang telah diuraikan sebelumnya, maka permasalahan yang akan dibahas dalam penelitian ini adalah, "Bagaimana Rancang bangun Aplikasi Data Mining *Association Rule* menggunakan algoritma *FP-Growth* pada data absensi elektronik untuk mendeteksi kecurangan absensi Karyawan".

1.3 Batasan Masalah

Batasan yang digunakan dalam penelitian ini agar pembahasan yang dilakukan tidak meluas yaitu :

1. Algoritma yang digunakan adalah Algoritma *FP-Growth*
2. Data absensi yang dilakukan pada mesin absen *Checkclock*
3. Data sampel yang digunakan untuk mendeteksi kecurangan absensi (*Checkclock*) adalah data simulasi absensi kedatangan.
4. Dari acuan jurnal sebelumnya, proses *binning/* pengelompokan diset dari kecepatan rata-rata dua orang karyawan melakukan absensi secara bergantian yaitu dengan frekuensi jarak absensi 10 detik.
5. Perbandingan hanya dilakukan antara dua orang karyawan yang melakukan absensi berdekatan. Sehingga itemset dalam pengelompokan data hanya terdiri dari dua item (karyawan).

1.4 Tujuan Penelitian

Tujuan penelitian tugas akhir ini adalah :

1. Rancang bangun Aplikasi Data Mining *Association Rule* pada Data Absensi Elektronik untuk Mendeteksi Kecurangan Absensi Karyawan.
2. Pengujian Algoritma *FP-Growth* dan struktur penyimpanan data *FP-Tree*. Yaitu pengujian pada proses pencarian frekuensi itemset yang diperoleh dari sekumpulan data, kecepatan, jumlah transaksi dan *rule* .
3. Untuk mengetahui dampak perubahan parameter-parameter dari *rule* yang dihasilkan.

1.5 Sistematika Penulisan

Laporan Tugas Akhir ini terdiri dari enam bab yang disusun dengan sistematika penulisan sebagai berikut :

BAB I. PENDAHULUAN

Bab ini berisi tentang Latar Belakang pelaksanaan penelitian, Rumusan Masalah yang dihadapi, Batasan yang digunakan, Tujuan Tugas Akhir yang hendak dicapai melalui penelitian ini serta Sistematika Penulisan.

BAB II. LANDASAN TEORI

Bab ini membahas teori-teori yang berhubungan dengan pembahasan penelitian yang diangkat, yang terdiri dari pembahasan mengenai konsep sistem, metode *Association Rule* yang digunakan untuk mendeteksi kecurangan absensi karyawan.

BAB III. METODOLOGI PENELITIAN

Pada bab ini dijelaskan mengenai tahapan dalam pelaksanaan penelitian tugas akhir. Tahapan penelitian tugas akhir dimulai dari identifikasi permasalahan hingga diperoleh kesimpulan dari penelitian dan saran yang dapat dipergunakan untuk penelitian selanjutnya.

BAB IV. ANALISA DAN PERANCANGAN

Bab ini berisi tentang analisa data yang telah diolah dengan menggunakan metode *Association Rule* yang akan digunakan untuk Mendeteksi Kecurangan Absensi (*Checkclock*) adalah Data Absensi Karyawan.

BAB V. IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi penjelasan mengenai implementasi yang terdiri dari: batasan implementasi, lingkungan implementasi, analisa hasil, pengujian sistem dan kesimpulan pengujian.

BAB VI. PENUTUP

Bab ini berisi kesimpulan yang dihasilkan dari pembahasan tentang Rancang bangun Aplikasi Data Mining *Association Rule* pada Data Absensi Elektronik untuk Mendeteksi Kecurangan Absensi (*Checkclock*) Karyawan dan beberapa saran sebagai hasil akhir dari penelitian yang telah dilakukan.

1.6 Rencana Kerja

Jadwal pengerjaan tugas akhir ini dijelaskan pada tabel 1.1 di bawah ini.

Tabel 1.1 Jadwal Rencana Pengerjaan Tugas Akhir

No	Kegiatan	Waktu Pelaksanaan							
		Juli	Agus	Sep	Okt	Nov	Des	Jan	Feb
1	Studi Literatur	■							
2	Pengajuan judul		■						
3	Analisa dan perancangan			■	■	■			
4	Implementasi dan pengujian					■	■		
5	Laporan Tugas Akhir			■	■	■	■	■	■
6	Seminar Hasil dan Sidang								■

BAB II

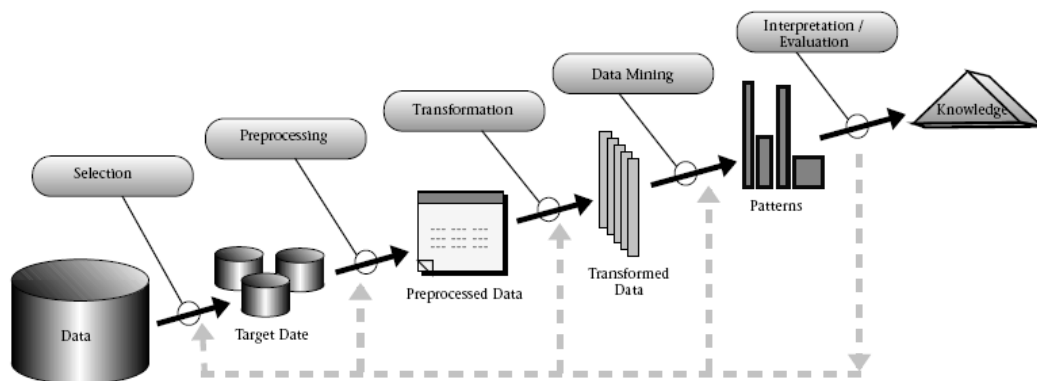
LANDASAN TEORI

2.1 *Knowledge Discovery in Databases (KDD)*

Knowledge discovery in databases (KDD) adalah proses untuk menemukan *interesting knowledge* dari sejumlah besar data yang disimpan baik di dalam *databases*, *data warehouses* atau tempat penyimpanan informasi lainnya (Gunawan, 2006).

Knowledge discovery in databases (KDD) berhubungan dengan teknik integrasi dan penemuan ilmiah, interpretasi dan visualisasi dari pola-pola sejumlah kumpulan data. *Knowledge discovery in databases (KDD)* adalah keseluruhan proses *non-trivial* untuk mencari dan mengidentifikasi pola (pattern) dalam data, dimana pola yang ditemukan bersifat sah, baru, dapat bermanfaat dan dapat dimengerti. (Gunawan, 2006).

Proses dari KDD dapat dilihat pada gambar di bawah ini Fayyad(1996):



Gambar 2.1 Tahap *Knowledge Discovery in Databases (KDD)*

Sebagai suatu rangkaian proses, KDD dapat dibagi menjadi beberapa tahap yang diilustrasikan di Gambar 2.1. Tahap-tahap tersebut bersifat interaktif di mana pemakai terlibat langsung atau dengan perantaraan *knowledge base*.

Proses *knowledge discovery in databases* (KDD) dibagi dalam beberapa tahap (Tessy, 2010) :

1. Selection

Melakukan reduksi terhadap data yang ada di dalam *database*. Proses reduksi diperlukan untuk mendapatkan hasil yang lebih akurat dan mengurangi waktu komputasi terutama untuk masalah dengan skala besar (*large scale problem*). Beberapa cara seleksi, antara lain:

- a. *Sampling*, adalah seleksi subset representatif dari populasi data yang besar.
- b. *Denoising*, adalah proses menghilangkan *noise* dari data yang akan ditransformasikan.
- c. *Feature extraction*, adalah proses membuka spesifikasi data yang signifikan dalam konteks tertentu.

2. Preprocessing

Pre-processing data adalah tahapan sebelum suatu data diproses, dapat berupa pembersihan data, transformasi data, atau yang lainnya.

- a. *Data cleaning* (Pembersihan data)

Proses ini digunakan untuk membuang data yang tidak konsisten dan bersifat *noise* dari data yang terdapat di berbagai basis data yang mungkin berbeda format maupun *platform* yang kemudian diintegrasikan dalam satu *database data warehouse*.

Pada umumnya data yang diperoleh, baik dari *database* suatu perusahaan maupun hasil eksperimen, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak valid atau juga hanya sekedar salah ketik. Selain itu, ada juga atribut-atribut data yang tidak relevan dengan hipotesa data mining yang kita miliki. Data-data yang tidak relevan itu juga lebih baik dibuang karena keberadaannya bisa mengurangi mutu atau akurasi dari hasil data mining nantinya. “*Garbage*

in garbage out” (hanya sampah yang akan dihasilkan bila yang dimasukkan juga sampah) merupakan istilah yang sering dipakai untuk menggambarkan tahap ini. Pembersihan data juga akan mempengaruhi performansi dari sistem data mining karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.

Tahap data *cleaning* :

1. Menangani *missing values* (nilai-nilai yang hilang)
2. *Smoothing* (menghaluskan) data yang *noise*. *Noise* adalah kesalahan yang terjadi secara random atau karena variasi yang terjadi dalam pengukuran variabel. Penangan *noise* pada data dapat diatasi dengan *Binning*, *Clustering*, dan *Regression*.
3. Menjadikan data lebih konsisten.

b. Intergrasi Data

Integritas data merupakan suatu kondisi dimana semua nilai-nilai yang dimasukkan atau disimpan didalam *database* telah sesuai dengan aturan yang ditetapkan untuk data tersebut. Jika data yang dimasukkan belum sesuai dengan ketentuan maka bisa dikatakan *database* belum terintegritasi.

Batasan dalam penentuan integritas data :

1. *Primary key constraints* (batasan *Primary key*)

Menentukan *Primary key* pada entitas, atribut yang menjadi *Primary key* tidak bisa memiliki nilai *null*, bila ada dua atribut yang menjadi *Primary key* maka hanya satu atribut yang benar-benar dianggap sebagai *Primary key*.

2. *Foreign key constraints* (batasan *Foreign key*)

Relasi antara entitas untuk menghindari penghapusan data yang memiliki relasi dengan entitas lain.

3. *Unique constraints*

Atribut yang bersifat unik, tidak bisa memiliki nilai yang sama

4. *Check constraints*

Membatasi nilai yang bisa dimasukkan pada suatu atribut dengan menguji pada suatu kondisi tertentu.

5. *Not null*

Menentukan spesifikasi dari suatu atribut, apakah atribut tersebut bisa memiliki nilai yang kosong.

6. *Default*

Mendefinisikan apa yang harus dilakukan bila data baru tidak memiliki data sama sekali pada suatu entitas yang akan disisipkan.

3. Transformasi data

Beberapa teknik data mining membutuhkan format data yang khusus sebelum bisa diaplikasikan. Sebagai contoh beberapa teknik standar seperti analisis asosiasi dan klustering hanya bisa menerima input data kategorikal. Karenanya data berupa angka numerik yang berlanjut perlu dibagi-bagi menjadi beberapa interval. Proses ini sering disebut *binning*. Disini juga dilakukan pemilihan data yang diperlukan oleh teknik data mining yang dipakai. Transformasi dan pemilihan data ini juga menentukan kualitas dari hasil data mining nantinya karena ada beberapa karakteristik dari teknik-teknik data mining tertentu yang tergantung pada tahapan ini. Transformasi dan pemilihan data ini untuk menentukan kualitas dari hasil data mining, sehingga data diubah menjadi bentuk yang sesuai untuk di miningkan.

Transformasi data diperlukan sebagai tahap *pre-procecing*, dimana data yang diolah siap untuk ditambang. Beberapa cara transformasi, antara lain (Santosa, 2007):

- a. *Centering*, mengurangi setiap data dengan rata-rata dari setiap atribut yang ada.
- b. *Normalisation*, membagi setiap data yang *dicentering* dengan standar deviasi dari atribut bersangkutan.
- c. *Scaling*, mengubah data sehingga berada dalam skala tertentu diproses lebih lanjut.

4. Aplikasi Teknik Data Mining

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

5. Evaluasi pola yang ditemukan

Pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti. Tahap ini merupakan bagian dari proses KDD yang mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

6. Presentasi Pengetahuan

Presentasi pola yang ditemukan untuk menghasilkan aksi tahap terakhir dari proses data mining adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisa yang didapat.

2.2 Konsep Data Mining

Secara sederhana data mining adalah ekstraksi informasi atau pola yang penting atau menarik dari data yang ada di *database*. *Data mining* merupakan bagian dari proses *Knowledge Discovery in Databases* (KDD).

Banyak pihak-pihak lain yang telah mendefinisikan *data mining*. Berikut beberapa definisi *data mining*:

1. *Data Mining* adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database* yang prosesnya menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning*, untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* (Turban, dkk. 2005).
2. *Data Mining* merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, *database*, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari *database* yang besar (Larose, 2005).
3. *Data mining* adalah ekstraksi informasi atau pola yang penting untuk menarik dari data yang ada di *database* yang benar sehingga menjadi informasi yang sangat berharga (Sucahyo, 2004)
4. *Data Mining adalah* merupakan proses penemuan yang efisien sebuah pola terbaik yang dapat menghasilkan sesuatu yang bernilai dari suatu koleksi data yang sangat besar (www.theartling.com, 2002)

Berdasarkan beberapa pengertian diatas dapat ditarik kesimpulan bahwa *data mining* merupakan proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Tujuan dari data mining adalah menemukan hubungan-hubungan atau pola-pola yang mungkin memberikan indikasi yang bermanfaat.

Data Mining merupakan konsep baru dalam penemuan nilai tambah dari sebuah data. Dengan data mining di dapatkan informasi-informasi yang selama ini tersembunyi dengan penemuan pola-pola dalam sebuah data. Bahkan dengan menggunakan *Data Mining* dapat memprediksi perilaku dan tren yang terjadi dalam perusahaannya atau memprediksi pola kebiasaan pelanggan sebagai konsumennya sehingga bisa membuat para pengusaha menjadi lebih proaktif dan dapat mengambil keputusan dengan benar.

2.3 Fungsi Data Mining

Fungsi *data mining* dapat diklasifikasikan menjadi dua kategori, yaitu: deskriptif dan prediktif. Yaitu (Tessy, 2010) :

2.3.1 Fungsi Prediktif

Model data mining prediktif Menggunakan beberapa variabel dari *database* untuk memprediksi suatu nilai yang akan datang atau untuk meramal masa depan. Pemodelan Prediksi memungkinkan user untuk mengirimkan record dengan beberapa field kosong, dan sistem akan menebak nilai yang kosong tersebut dengan pola-pola sebelumnya yang ditemukan dari basis data. Model prediktif diantaranya klasifikasi data, *decision tree*, *analisis time series*, *regresi*, prediksi, dan jaringan syaraf tiruan.

2.3.2 Fungsi Deskripsi

Melakukan ekstraksi data untuk penemuan informasi data yang tidak dapat dilakukan secara manual. Hal pertama yang dilakukan yaitu pencarian norm dari data tersebut, kemudian mendeteksi item-item deviasi dari data yang biasa (usual) dengan *frequensi item* yang telah diberikan.

Mendapatkan pola penafsiran (*human-interpretable patterns*) untuk menjelaskan data. Pola dan trend data sering dideskripsikan. Deskripsi tersebut sangat membantu dalam menjelaskan pola dan trend yang terjadi. Model *data mining* harus setransparan mungkin, dimana hasilnya dapat mendeskripsikan pola dengan jelas. *Clustering data*, *summarization*, *association rule*, *sequence discovery* merupakan bagian dari metode deskripsi.

Melalui tugas *descriptive mining*, dapat dilakukan penggolongan data dalam *database*, sedangkan melalui tugas *predictive mining*, yang ada dapat digunakan untuk membuat suatu prediksi.

Fungsi data mining dan jenis-jenis pola yang dapat ditemukan adalah sebagai berikut (Han, Kamber, 2001) :

1. *Concept/Class Description: Characterization and Discrimination*

Data dapat diasosiasikan dengan *class* atau *concept*. *Concept/class description* dapat diperoleh melalui *data characterization*, *data discrimination*, atau kedua-duanya. *Data characterization* adalah ringkasan dari karakter atau ciri umum dari *target class*. Sedangkan *data discrimination* adalah perbandingan ciri-ciri umum dari *target class* dari data *object* dengan ciri-ciri umum dari *object* dari satu atau serangkaian *class* yang kontras.

2. *Association Analysis*

Association analysis adalah penemuan *association rule* yang menunjukkan pola-pola yang sering muncul dalam data. Terdapat nilai *support* dan *confidence* yang dapat menunjukkan seberapa besar suatu *rule* dapat dipercaya. *Support* adalah ukuran dimana seberapa besar tingkat dominasi suatu *item* atau *itemset* terhadap keseluruhan transaksi. Cara perhitungannya adalah dengan rumus:

$$support(A \rightarrow B) = p(A \ Y \ B) \dots \dots \dots (2.1)$$

Sedangkan *confidence* adalah ukuran yang menunjukkan hubungan antara dua *item* secara *conditional*. Cara perhitungannya adalah dengan rumus:

$$confidence(A \rightarrow B) = p \frac{(A \ Y \ B)}{p(A)} \dots \dots \dots (2.1)$$

Untuk dapat lebih memahami apa yang dimaksud dengan *support* dan *confidence*, dapat dilihat contoh di bawah ini:

membeli(T, komputer) → membeli (T, Software)
 [support 1 % confidence 50 %]

Arti dari *rule* di atas adalah jika pada sebuah transaksi, T, membeli “*computer*”, ada peluang sebesar 50% bahwa pada transaksi tersebut juga membeli “*software*”, dan pada keseluruhan transaksi terdapat peluang 1% keduanya sama-sama dibeli.

3. *Classification and Prediction*

Classification adalah proses menemukan model yang mendeskripsikan *class* dan *concept* dari data. *Classification* juga dapat digunakan untuk memprediksi *class label* dari *data object*. Pada banyak aplikasi, *user* lebih menginginkan memprediksi *missing* atau *unavailable data value* daripada *class label*. Hal ini biasanya terjadi pada kasus dimana *value* yang akan diprediksi adalah berupa data numerik.

4. *Cluster Analysis*

Berbeda dengan *classification* dan *prediction*, *cluster analysis* dilakukan tanpa mengetahui *class label*. *Cluster* dari *object* dibentuk jika *object* di dalam suatu *cluster* memiliki kemiripan yang tinggi dengannya, dan memiliki ketidakmiripan dengan *object* di *cluster* lainnya.

5. *Outlier Analysis*

Sebuah *database* dapat mengandung data *object* yang tidak sesuai atau menyimpang dari model data. *Data object* ini disebut *outlier*. Banyak metode data mining yang menghilangkan *outlier* ini. Padahal, pada beberapa aplikasi seperti *fraud detection*, kejadian yang jarang terjadi justru lebih menarik untuk dianalisa daripada kejadian yang sering terjadi. Analisa dari *outlier* data disebut sebagai *outlier mining*.

6. *Evolution Analysis*

Data evolution analysis mendeskripsikan model yang beraturan atau *trend* untuk *object* yang sifatnya terus menerus berubah.

2.4 ***Association Rule Mining***

Association rule mining merupakan salah satu model data mining deskriptif. *Association rule* adalah suatu prosedur untuk mencari hubungan antar *item* dalam suatu *data set* yang ditentukan.

Association rule meliputi dua tahap (Han,Kamber, 2001) :

1. Mencari kombinasi yang paling sering terjadi dari suatu *itemset*.
2. Mendefinisikan *condition* dan *result* untuk *conditional association rule*.

2.4.1 Ukuran *Association Rule Mining*

Dalam menentukan suatu *association rule*, terdapat ukuran yang menyatakan bahwa suatu informasi atau *knowledge* dianggap menarik (*interestingness measure*). Ukuran ini didapatkan dari hasil pengolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran kepercayaan (*interestingness measure*) yang digunakan dalam menentukan suatu *association rule*, yaitu (Han, kamber, 2001) :

1. *Support*

Suatu ukuran yang menunjukkan seberapa besar tingkat dominasi suatu *item* atau *itemset* dari keseluruhan transaksi. Ukuran ini menentukan apakah suatu *item* atau *itemset* layak untuk dicari *confidence*-nya (misalnya, dari keseluruhan transaksi yang ada, seberapa besar tingkat dominasi yang menunjukkan bahwa *item* A dan B dibeli bersamaan).

$$support(A \rightarrow B) = \frac{|A \cup B|}{|D|} \dots \dots \dots (2.3)$$

2. *Confidence*

Suatu ukuran yang menunjukkan hubungan antar dua *item* secara *conditional* (misalnya, seberapa sering *item* B dibeli jika orang membeli *item* A). Perhitungan *confidence* menggunakan rumus :

$$confidence(A \cup B) = \frac{support\ count(A \cup B)}{support\ count(A)} \dots \dots \dots (2.4)$$

3. *Correlation*

Association rules dibentuk dengan menggunakan ukuran *support-confidence*. Ukuran *support-confidence* akan menjadi membingungkan jika menyatakan bahwa *rule* $A \Rightarrow B$ adalah *interesting*, sedangkan kemunculan A tidak mempengaruhi kemunculan B. *Correlation* merupakan alternatif lain dalam

menemukan *interesting relationship* antara *itemset* data berdasarkan hubungan atau korelasinya. Perhitungan *correlation* dapat dilakukan menggunakan Rumus *confidence*. Jika nilai yang dihasilkan oleh CRA,B kurang dari satu (CRA,B<1), maka kemunculan A tidak terlalu berhubungan dengan kemunculan B. Jika nilai yang dihasilkan lebih besar dari satu (CRA,B>1), maka A dan B berhubungan, artinya kemunculan yang satu akan mempengaruhi kemunculan yang lainnya. Jika nilai yang dihasilkan sama dengan satu (CRA,B =1), maka A dan B saling berdiri sendiri dan tidak ada hubungan diantara keduanya.

$$\begin{aligned} \text{Correlation (A}\Rightarrow\text{B)} &= \text{Correlation (B }\Rightarrow\text{ A)} \\ &= \frac{\text{suppot A}\cup\text{B}}{\text{support (A)}*\text{support (B)}} \dots \dots \dots (2.5) \end{aligned}$$

Untuk menghitung probabilitas dari suatu kejadian digunakan rumus:

$$P(A) = \frac{\text{Banyak anggota kejadian}}{\text{Banyak anggota ruang sampel}}$$

Dimana, Rumus ini hanya untuk kejadian diskrit dengan anggota ruang sampel yang *uniform*.

1. Ruang sampel adalah kumpulan dari semua kemungkinan hasil eksperimen.
2. Kejadian adalah bagian dari ruang sampel

Support dan *confidence* nantinya berguna dalam menentukan *interesting association rules*, yaitu untuk dibandingkan dengan batasan (*threshold*) yang ditentukan oleh *user*. Batasan tersebut umumnya terdiri dari *min_support* dan *min_confidence*. Bila memenuhi kedua batasan maka sebuah *rule* dapat disebut *interesting rule*.

minimum support count adalah nilai minimum transaksi yang terlibat dalam setiap pembelian *itemset* (group variasi produk). Sedangkan *confidence* adalah besar nilai keyakinan atau kepastian bahwa suatu *itemset* lain akan turut pada saat bersamaan dalam suatu *itemset* tertentu.

Aturan asosiasi menggambarkan item atau kejadian dalam data berukuran besar yang berisi data transaksi. Dengan kemajuan teknologi, data penjualan dapat disimpan dalam jumlah besar yang disebut dengan "basket data." Aturan asosiasi yang didefinisikan pada basket data, digunakan untuk keperluan promosi, desain katalog, segmentasi customer dan target pemasaran. Secara tradisional, aturan asosiasi digunakan untuk menemukan trend bisnis dengan menganalisa transaksi customer.

Akan tetapi karena konsep *association rule* merupakan konsep yang mencari frekuensi suatu pola yang muncul secara bersamaan, *Association rule* tidak hanya di terapkan untuk *market basket analysis* hal ini juga dapat diterapkan dalam data absensi karyawan yakni mencari pola kebiasaan absensi karyawan.

2.4.2 Klasifikasi *Association Rule Mining*

Association rule dapat diklasifikasikan menjadi beberapa bagian berdasarkan :

1. Berdasarkan tipe nilai yang dapat ditangani *rule*

Ada dua tipe nilai yang dapat ditangani oleh *Rule*, yaitu:

- a. ***Boolean Association Rule***: Jika suatu rule hanya menangani ada tidaknya hubungan antar item.

Contoh: **computer** \Rightarrow **financial_management_software**

atau dapat ditulis: **buys(X, "computer")** \Rightarrow **buys(X, "financial_management_software")**

- b. ***Quantitative Association Rule***: bila rule tersebut dapat menunjukkan hubungan antar item atau atribut secara kuantitatif.

Contoh: **age(X, "30..39")** \Rightarrow **income(X, "42K..48K")** \Rightarrow **buys(X, high resolution TV)**

Pada contoh ini atribut *age* dan *income* telah mengalami diskretisasi.

2. Berdasarkan Dimensi Data yang Terdapat pada *Rule*

Adapun dimensi data yang terdapat pada *Rule*, yaitu:

- a. ***Single Dimensional Association Rule***: bila item atau atribut dalam rule hanya melibatkan satu dimensi saja.

Contoh: $buys(X, \text{"IBM desktop computer"}) \Rightarrow buys(X, \text{"Sony B/W Printer"})$

Single Dimensional Association Rule disebut juga ***Intradimension Association Rule***, karena hanya terdiri dari satu buah predikat (*buys*) dengan beberapa pengulangan (predikat digunakan lebih dari satu kali dalam sebuah rule).

- b. ***Multidimensional Association Rule***

Metode ini memungkinkan penggalian informasi ditinjau dari beberapa atribut atau dimensi, dibandingkan *single-dimensional* umumnya (Han, Kamber, 2001).

Contoh: $age(X, \text{"20...29"}) \wedge occupation(X, \text{"student"}) \Rightarrow buys(X, \text{"laptop"})$.

Kategori Multidimensional Association Rule:

1. ***Interdimension Association Rule (no repeated predicates)*** : jika tanpa predikat yang diulang, contoh: $umur(x, \text{"19-25"}) \wedge pekerjaan(x, \text{"siswa"}) \Rightarrow beli(x, \text{"Coca Cola"})$

Artinya: Jika seseorang dengan umur antara 19-25 tahun dan berprofesi sebagai siswa maka akan membeli Coca Cola.

Rule ini melibatkan tiga atribut yaitu umur, pekerjaan, dan produk.

2. ***Hybrid-dimension Association Rule (repeated predicates)***: jika terdapat satu atau lebih predikat yang diulang, contoh: $umur(x, \text{"19-25"}) \wedge beli(x, \text{"popcorn"}) \Rightarrow beli(x, \text{"Coca Cola"})$

Artinya: Jika seseorang dengan umur antara 19-25 tahun dan membeli popcorn maka akan membeli Coca Cola. *Rule* ini melibatkan dua atribut yaitu umur dan produk.

Contoh:

Umur (x,"20-25") \wedge penyakit(x,"bronkitis") \Rightarrow penyakit(x,"kanker paru-paru")

Artinya jika seseorang dengan umur 20-25 dan menderita penyakit bronkitis maka juga menderita penyakit kanker paru-paru. *Rule* ini melibatkan dua atribut yaitu umur dan penyakit.

2.5 *Discretization Histogram Analysis*

Histogram menggunakan metode *binning* guna memperkirakan distribusi data dan juga untuk mereduksi jumlah data. Pada histogram, data dibagi – bagi menjadi subset – sub set data yang dinamakan bucket. Sebuah bucket bisa berupa sebuah value tertentu atau bisa juga merepresentasikan continuous ranges dari sekelompok data. (Setia Budhi, 2010).

MaxDiff Histogram adalah salah satu metode *Discretization Histogram*, Pada metode ini batasan tiap *bucket* ditentukan dengan cara memperhitungkan selisih dari sepasang data yang bersebelahan. Isi tiap *bucket* ditentukan dengan rumus $\beta - 1$. Selama selisih dua pasang data masih lebih kecil atau sama dengan $\beta - 1$, maka sekelompok data tersebut berada pada *bucket* yang sama, bila tidak, saatnya membuat *bucket* baru (Setia Budhi, 2010).

2.6 *Pembangunan Frequent Pattern tree (FP-TREE)*

Algoritma FP-Growth merepresentasikan transaksi dengan menggunakan struktur data *FP-Tree*. *FP-tree* merupakan struktur penyimpanan data yang dimampatkan (data yang disimpan secara ditimpakan). *FP-tree* dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam *FP-tree*. Karena dalam setiap transaksi yang dipetakan, mungkin ada transaksi yang memiliki item yang sama, maka lintasannya memungkinkan untuk saling

menimpa. Semakin banyak data transaksi yang memiliki item yang sama, maka proses pemampatan dengan struktur data *FP-tree* semakin efektif. Kelebihan dari *FP-tree* adalah hanya memerlukan dua kali tahapan data transaksi yang terbukti sangat efisien. (Samuel, 2008)

Proses penyusunan *FP-Tree* dari mulai representasi awal transaksi, pengurutan dengan hanya mempertahankan *frequent 1-itemset*, dan penyimpanannya di *FP-Tree*. Setelah *FP-Tree* terbentuk, langkah selanjutnya adalah memperoleh *frequent itemset* tanpa melakukan *candidates generation*.

Misal $I = \{a_1, a_2, \dots, a_n\}$ adalah kumpulan dari item. Dan basis data transaksi $DB = \{T_1, T_2, \dots, T_n\}$, di mana T_i ($i \in [1..n]$) adalah sekumpulan transaksi yang mengandung item di I . Sedangkan *support* adalah penghitungan (*counter*) frekuensi kemunculan transaksi yang mengandung suatu pola. Suatu pola dikatakan sering muncul (*frequent pattern*) apabila *support* dari pola tersebut tidak kurang dari suatu konstanta ζ (batas ambang minimum *support*) yang telah didefinisikan sebelumnya. Permasalahan mencari pola frequent dengan batas ambang minimum *support count* ζ inilah yang dicoba untuk dipecahkan oleh FP-Growth dengan bantuan Struktur *FP-tree*.

Adapun *FP-tree* adalah sebuah pohon dengan definisi sebagai berikut (Samuel, 2008) :

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

2.7 Algoritma *Frequent Pattern Growth (FP-GROWTH)*

Setelah tahap pembangunan *FP-tree* dari sekumpulan data transaksi, akan diterapkan algoritma *FP-growth* untuk mencari *frequent itemset* yang signifikan.

Informasi yang disimpan sebuah *node FP-Tree* berupa *Item*, *Index parent*, *Support*, dan *Next (Pointer)*. Ketika selesai membuat *FP-Tree*, kita tidak begitu saja bisa mendapatkan *frequent itemset* yang terdapat dalam *dataset*. Suatu kombinasi *itemset* bisa saja berada di beberapa *path* yang berbeda. Untuk mendapatkan suatu *pattern* dalam *FP-Growth* langkah yang lebih mudah adalah mencari arah dari ujung suatu *path*, kemudian kita mencari mulai dari *header* untuk *item* di ujung tersebut, barulah kemudian dibuat berdasarkan tiap *node* berisi *item* tersebut dicari arah *path node* ke atas. Hal ini tentu lebih cepat dari pada *up-down* karena *pointer* langsung yang dimiliki tiap *node* adalah *pointer* ke *parent*. *Path-path* yang dieksplorasi hanyalah *path-path* yang memiliki *node* yang sedang dicari. Jadi dalam struktur *FP-Tree* ada *link* dari suatu *item* ke *path-path* yang memiliki *item* tersebut, sehingga ketika dibutuhkan pencarian *pattern-pattern* untuk suatu *item* tertentu, hanya mencari dari *path-path* tersebut saja.

Algoritma *FP-growth* dibagi menjadi tiga langkah utama, yaitu (Soelaiman, 2006) :

1. Tahap Pembangkitan *Conditional Pattern Base*

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

2. Tahap Pembangkitan *Conditional FP-tree*

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

3 Tahap Pencarian *frequent itemset*

Apabila Conditional FP-tree merupakan lintasan tunggal (*single path*), maka didapatkan *frequent itemset* dengan melakukan kombinasi item untuk setiap *conditional FP-tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan *FP-growth* secara rekursif.

Ketiga tahap tersebut merupakan langkah yang akan dilakukan untuk mendapat *frequent itemset*, yang dapat dilihat pada algoritma berikut : (Soelaiman, 2006)

```
Input: Output: R sekumpulan lengkap pola  
t frequent  
growth(Tree,null)  
Method: FP-Procedure: FP-growth(Tree, $\alpha$ ) {  
01:ee mengandung single path P; if Tr02:then untuk tiap  
kombinasi (dinotasikan  
 $\beta$ ) dari node-node dalam path P do  
03:bangkitkan pola  $\beta$   $\alpha$  dengan support =  
minimum support dari node-node  
dalam  $\beta$ ;  
04:else tuk tiap  $a_i$  dalam header dari  
unTree do {  
05:bangkitkan pola  
06:bangun  $\beta = a_i \alpha$  dengan  
support =  $a_i$ .support  
07:if ee  $\beta = \emptyset$   
Tr08:then panggil FP-growth(Tree, $\beta$ ) } }
```

Gambar 2.3 Algoritma *FP-Growth*

3.7 *Divide dan Conquer*

Divide and conquer adalah varian dari beberapa strategi pemrograman *topdown*, tetapi keistimewaannya adalah membuat sub-sub problem dari problem yang besar, oleh karena itu strategi ini ditunjukkan secara berulang-ulang (*recursively*), didalam menerapkan algoritma yang sama dalam sub-sub problem seperti yang diterapkan pada masalah aslinya (original problem). Sebagaimana prinsip dasar algoritma perulangan dibutuhkan sebuah kondisi untuk mengakhiri perulangan tersebut. Biasanya untuk mengecek apakah problem sudah cukup kecil untuk diselesaikan dengan metode secara langsung. Mungkin dari segi ilustrasi kita, bahwa proses-proses pada komputer paralel tentunya memiliki proses/problem/job yang cukup kompleks sehingga harus dipecah-pecah menjadi sub-sub problem. Selain dibutuhkan sebuah “kondisi”, juga diperlukan “*fase divide*” untuk membagi/memecah problem menjadi sub-sub problem yang lebih kecil, dan “*fase combine*” untuk menggabungkan kembali solusi dari sub-sub problem kedalam solusi dari problem awalnya. (Prihastomo, 2008).

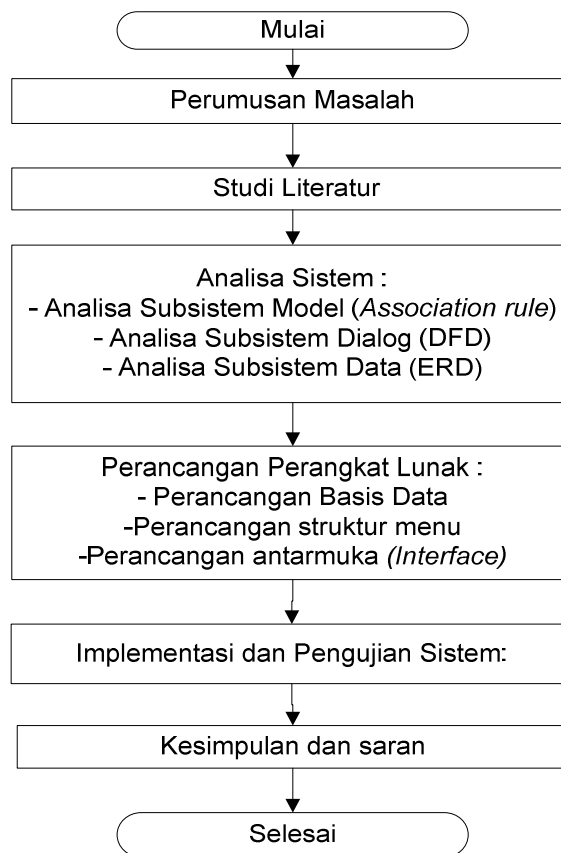
Algoritma *FP-growth* menemukan *frequent itemset* yang berakhiran *suffix* (pola akhir) tertentu dengan menggunakan metode *divide and conquer* untuk memecah problem menjadi subproblem yang lebih kecil (Soelaiman, 2006). Misalnya jika kita ingin menemukan semua frekuensi itemset yang berakhiran item k, hal yang pertama dilakukan adalah pembangkitan *conditional pattern base* pada sub *database* yang berisikan *prefix path* (himpunan item terurut yang mengawali k-itemsets), dan *suffix pattern (k-itemsets)*. Menghitung *support count* dari k-itemset, jika memenuhi maka item k termasuk frequent. Karena item k *frequent*, maka perlu di pecah ke subproblem selanjutnya yaitu menemukan item k yang *frequent* dengan penggabungan item lainnya. Dari pemecahan problem yang ada, maka ditemukan himpunan itemset yang berakhiran k.

BAB III

METODOLOGI PENELITIAN

Metodologi penelitian mempunyai peranan sangat penting sekali dalam penelitian tugas akhir, karena pada metodologi penelitian ini menggambarkan langkah-langkah secara sistematis yang dilakukan dalam memecahkan permasalahan yang diangkat. Deskripsi dilengkapi dengan penyajian diagram alur pelaksanaan penelitian untuk memudahkan dalam memahami tahapan penelitian.

Berikut ini adalah metodologi yang digunakan dalam penelitian tugas akhir yang berjudul ” Rancang Bangun Aplikasi Deteksi Kecurangan Absensi Karyawan Menggunakan Metode Data Mining *Association Rules*”. Untuk lebih jelasnya tentang metodologi penelitian ini dapat di lihat pada Gambar 3.1. *Flowchart* metodologi penelitian berikut :



Gambar 3.1. Flowchart Metodologi Penelitian

3.1 Perumusan Masalah

Merumuskan masalah tentang pola kebiasaan absensi karyawan untuk mendeteksi kecurangan absensi karyawan yang akan dioperasikan dengan menggunakan algoritma *fP-Growth* dan struktur penyimpanan data *fP-Tree* yang terdapat dalam proses KDD..

3.2 Studi Literatur

Literatur yang diperoleh peneliti antara lain:

1. Teori mengenai Data mining *Association Rule*
2. Konsep *FP-tree* dan Algoritma *FP-Growth*
3. Teori pendukung.

3.3 Analisa Sistem

Analisa dilakukan terhadap sistem untuk mendeteksi kehadiran karyawan berdasarkan ketepatan waktunya. Hal ini dilakukan untuk mengetahui apakah sistem tersebut sudah dapat memenuhi kebutuhan dalam hal tingkat akurasi data-data kehadiran berdasarkan ketepatan kedatangan.

Analisa dalam pembuatan sistem ini menggunakan metode data mining *Association Rules* serta penggunaan *Data Flow Diagram* untuk menganalisa kebutuhan sistem. nantinya akan di inputkan ke dalam sistem yang akan dibangun dengan menggunakan metode data mining *Association Rules*. Adapun analisa sistem yang akan digunakan meliputi:

3.3.1 Analisa subsistem model (*Model Association Rule*)

Membuat analisa terhadap model *Association rule* yang diterapkan dalam kasus pola kebiasaan absensi untuk mendeteksi kecurangan absensi karyawan. *Association rule* merupakan tahapan dari KDD. Analisa ini menjelaskan tahapan proses yang terjadi dalam penentuan frekuensi kemunculan karyawan dalam data absensi karyawan. Adapun tahapan tersebut yaitu Seleksi data, pembersihan data, tranformasi data, aplikasi teknik data mining, evaluasi pola yang ditemukan, dan presentasi pengetahuan.

Tahapan KDD diantaranya :

a. Seleksi data

Pada tahap ini data mentah data absensi dapat dipilih oleh *user* untuk diproses pada bulan tertentu.

b. Pembersihan data.

Hal ini dilakukan jika data absensi karyawan kosong/ tidak melakukan absensi. Dalam hal ini data absensi bernilai *default*.

c. Transformasi data

Setelah itu, dengan didukung oleh data master karyawan, data absensi yang telah dipilih dirubah menjadi data transaksi dengan cara dikelompok - kelompokkan menggunakan metode *discretization MaxDiff Histogram* berdasarkan waktu absensi karyawan yang berdekatan. Untuk pengelompokan ini *user* perlu memasukkan nilai β (*beta*) dalam satuan detik Nilai β (*beta*) ini harus lebih kecil daripada waktu terpendek 2 orang karyawan melakukan absensi / *checklock* secara bergantian. Sehingga terbentuk kelompok-kelompok data.

d. Association Rules Processing

Tahap ini adalah tahapan utama pada perangkat lunak. Ada 2 macam proses pada tahap ini, proses pertama adalah menggali *frequent itemset* dari *compact transaction data* menggunakan algoritma *FP-Growth* dan penerapan konsep *FP-Tree*, untuk proses ini user perlu memasukkan nilai *minimum support* untuk frekuensi kemunculan karyawan dengan karyawan yang lain . Proses kedua adalah merubah *frequent itemset* yang telah digali menjadi *association rule*. Untuk membatasi jumlah *rule* yang dihasilkan, user perlu menentukan nilai *support count* dari dua orang karyawan yang melakukan absensi secara berdekatan.

e. Evaluasi pola yang ditemukan

Pada tahap ini analisa dilakukan untuk mengetahui hasil deteksi sistem. Analisa dilakukan dengan mengubah- ubah nilai parameter dari batas frekuensi kemunculan item dan itemset. Adapun keluaran sistem berupa *Association rule*

kemunculan dua orang karyawan yang sama saat melakukan absensi sesuai nilai *support count* yang ditentukan.

f. Presentasi pengetahuan

Informasi yang didapat berdasarkan perubahan parameter-parameter dari *rule* yang dihasilkan.

3.3.2 Analisa subsistem dialog

Menganalisa struktur menu sistem dengan bantuan pemodelan *Data Flow Diagram* (DFD).

3.3.3. Analisa subsistem data

Pada tahap ini dilakukan analisa terhadap data-data yang diperlukan agar sistem dapat berjalan sesuai harapan yang dimodelkan ke dalam ERD (*Entity Relationship Diagram*).

3.4 Perancangan Perangkat Lunak

Setelah melakukan analisa, maka kemudian dilanjutkan dengan perancangan sistem berdasarkan analisa permasalahan yang telah dilakukan sebelumnya.

3.4.1 Perancangan Basis Data

Setelah menganalisa sistem yang akan dibuat, maka tahap selanjutnya adalah analisa dan perancangan basis data yang dilakukan untuk melengkapi komponen sistem.

3.4.2 Perancangan Struktur Menu

Rancangan struktur menu diperlukan untuk memberikan gambaran terhadap menu-menu atau *fitur* pada sistem yang akan dibangun.

3.4.3 Perancangan Antar Muka (*Interface*)

Untuk mempermudah komunikasi antara system dengan pengguna, maka perlu dirancang antar muka (*interface*). Dalam perancangan *interface* hal terpenting yang ditekankan adalah bagaimana menciptakan tampilan yang baik dan mudah dimengerti oleh pengguna.

3.5 Implementasi Dan Pengujian Sistem

3.5.1 Implementasi Sistem

Pada proses implementasi ini akan dilakukan pembuatan modul-modul yang telah dirancang dalam tahap perancangan ke dalam bahasa pemrograman. Implementasi sistem akan dilakukan dengan spesifikasi sebagai berikut :

Operating System : *Windows XP Professional*

Memory : 512 MB

Bahasa Pemrograman : *Visual Basic 6*

DBMS : *MySQL*

Report Engine : *Crystal Reports v10*

3.5.2 Pengujian Sistem

Tahapan pengujian dilakukan untuk mengevaluasi *input, output, interface, database* serta laporan yang dihasilkan sistem. Metode yang dipakai diproses pengujian adalah pengujian *Black Box* dan pengujian berdasarkan dengan nilai parameter yang diubah-ubah.

3.6 Kesimpulan Akhir

Dalam tahap ini dilakukan analisa akhir terhadap hasil pengujian yang telah dilakukan, untuk mengetahui apakah implementasi dari Aplikasi Data Mining *Association Rule* pada Data Absensi Elektronik untuk Mendeteksi Kecurangan Absensi Karyawan dengan penerapan algoritma *FP-Growth* dapat berjalan. Pada tahap ini juga ditarik kesimpulan-kesimpulan dalam rancang bangun program tersebut serta saran untuk perbaikan pengembangan sistem ini.

BAB IV

ANALISA DAN PERANCANGAN

Pada pembuatan sebuah sistem berbasis komputer, analisa memegang peran yang sangat penting dalam membuat rincian sistem, analisa perangkat lunak merupakan langkah pemahaman persoalan sebelum mengambil tindakan atau keputusan penyelesaian hasil utama. Sedangkan tahap perancangan sistem adalah membuat rincian sistem dari hasil analisis menjadi bentuk perancangan agar dimengerti oleh pengguna.

4.1 Analisa Sistem

Pada penelitian sebelumnya telah diterapkan algoritma *pincers search* dalam pencarian frekuensi kemunculan itemset (karyawan absensi berdekatan). Algoritma ini mengkombinasikan dua pendekatan yaitu *bottom-up* dan *top-down*. Pada awalnya pencarian menggunakan algoritma *pincer search* samahalnya dengan algoritma apriori yaitu dengan mencari himpunan kandidat dengan panjang $(k+1)$ dari sekumpulan pola *frequent* dengan panjang k , lalu mencocokkan jumlah kemunculan pola tersebut dengan informasi yang terdapat dalam *database*. Pada algoritma *pincers search* tahapan ini disebut *bottom-up*. Disamping untuk menghitung jumlah himpunan kandidat pada tahapan *bottom-up* algoritma ini juga melakukan pencarian *top-down* yang berfungsi untuk memotong atau memisahkan itemset yang *frequent* dan yang *infrequent*, tentunya disesuaikan dengan pencarian *bottom-up*.

Metode pencarian secara dua arah seperti ini dapat mempercepat pencarian frekuensi kemunculan item, akan tetapi dikarenakan *pincers search* memerlukan langkah *candidate generation* sama halnya dengan apriori, yaitu melakukan *scanning database* yang berulang-ulang jika itemset yang menjadi kandidat tidak *frequent* (*infrequent itemset*).

Sistem yang akan dibangun adalah suatu sistem yang menerapkan konsep *Association Rule*, samahalnya dengan konsep penelitian sebelumnya akan tetapi

dengan penerapan algoritma yang berbeda yaitu algoritma *FP-Growth*, Selain itu juga memanfaatkan konsep *FP-Tree* dalam struktur penyimpanan datanya. Cara penyimpanan data yang ditimpakan dengan akumulasi data memudahkan sistem dalam pembacaan data (*scanning database*). Data hanya memerlukan dua kali proses *scanning database* untuk menentukan *frequent itemset*. Dengan menggunakan *FP-Tree*, algoritma *FP-Growth* dapat langsung mengekstrak *frequent itemset* dari *FP-tree* yang telah terbentuk.

4.2 Menghitung frekuensi kemunculan itemset kecurangan absensi karyawan menggunakan algoritma *FP-Growth* Dalam *Knowledge Discovery in Database (KDD)*

4.2.1 *Data Selection*

Pemilihan himpunan data, atau memfokuskan pada subset variabel atau sampel data, dimana penemuan (*discovery*) akan dilakukan. Pemilihan (*seleksi*) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai dari data mentah absensi yang dipilih oleh *user* untuk diproses pada jangka waktu tertentu. Selain itu data absensi juga didukung oleh data master pegawai yang memerlukan identitas seperti nama dan NIK. Data absensi pegawai merupakan data yang akan digunakan untuk proses *data mining*. Variabel inputan untuk data absensi berupa nomor absensi, nomor induk karyawan, tanggal dan waktu absensi.

Berikut adalah tabel simulasi data absensi karyawan dimana terdiri dari 15 karyawan dari data absensi selama satu bulan.

Tabel 4.1 Absensi

Tanggal	Nama Karyawan														
	K 1	k 2	K 3	K 4	K 5	K 6	K 7	K 8	K 9	K 10	K 11	K 12	K 13	K 14	K 15
1 mei 2010	6:21:01	6:54:14	8:31:05	8:11:24	7:03:56	8:03:59	7:13:09	7:06:25	8:36:10	6:52:05	8:04:03	6:21:04	7:12:50	8:27:22	8:41:25
3 mei 2010	7:35:05	7:56:01	7:45:29	8:00:55	7:44:00	6:44:48	7:56:06	7:56:03	7:58:44	8:01:03	6:55:20	7:45:50	7:48:00	8:16:16	7:33:21
4 mei 2010	7:40:54	8:00:03	8:02:29	7:59:15	7:55:43	8:00:06	7:56:20	8:04:46	7:58:34	8:03:56	7:45:56	7:40:59	7:59:31	7:41:01	7:53:06
5 mei 2010	7:46:12	8:12:53	7:29:33	7:58:58	8:10:20	8:00:03	7:50:02	7:55:55	7:44:00	8:07:40	7:53:30	7:46:15	7:50:09	8:00:05	8:10:43
6 mei 2010	7:47:21	7:23:20	7:48:30	7:54:23	7:51:23	8:00:05	7:50:37	8:01:18	7:58:24	8:01:47	7:49:04	7:47:19	7:50:08	7:49:07	7:10:29
7 mei 2010	7:46:24	8:06:16	7:52:05	8:00:32	8:02:04	7:56:12	8:14:00	8:18:00	7:54:13	7:55:54	7:54:38	7:54:09	7:56:34	7:40:50	7:51:35
8 mei 2010	7:21:12	6:22:27	7:01:25	8:15:33	7:23:56	6:50:13	7:04:00	6:23:07	8:41:31	7:43:22	7:55:09	6:21:38	7:58:21	8:05:29	7:00:46
10 mei 2010	7:14:04	8:05:45	7:35:12	8:01:43	8:08:25	8:08:12	7:45:04	8:07:30	7:53:42	8:04:00	8:26:53	7:13:56	7:48:34	7:38:08	7:45:09
11 mei 2010	7:50:59	7:34:59	7:58:45	8:04:56	7:54:49	7:34:55	7:57:15	7:55:23	7:51:43	7:56:08	7:55:59	7:00:00	7:39:23	7:56:34	7:51:48
12 mei 2010	7:55:06	8:08:05	7:51:00	8:03:47	8:00:37	8:00:10	7:41:35	7:59:05	7:03:00	8:00:45	7:51:21	7:55:10	7:51:51	8:00:26	7:49:01
14 mei 2010	7:33:13	8:00:10	7:58:57	8:07:28	8:01:36	8:00:06	7:50:00	7:36:50	7:54:47	8:03:24	7:54:41	7:33:19	8:06:37	8:03:57	7:44:46
15 mei 2010	6:34:03	9:00:00	7:32:23	8:09:02	8:43:00	8:03:50	7:33:16	8:22:57	7:10:41	8:05:06	8:51:54	6:45:00	7:58:37	7:06:00	7:29:51
17 mei 2010	7:43:56	8:07:44	8:00:11	7:50:48	7:53:05	8:00:05	7:46:21	7:55:45	7:51:15	8:03:43	7:53:54	6:47:01	7:41:51	8:12:20	7:38:26
18 mei 2010	7:51:27	8:01:05	7:58:08	8:03:28	7:51:33	7:15:03	7:57:03	8:05:27	7:53:00	8:07:52	7:59:38	7:59:19	7:58:19	8:08:46	8:01:38
19 mei 2010	7:33:26	8:05:57	7:51:03	8:00:08	7:50:42	7:22:44	7:54:11	8:07:45	7:51:46	8:06:08	7:50:57	7:39:21	8:00:11	7:14:39	7:52:23
20 mei 2010	7:43:27	8:10:09	7:35:00	8:04:12	8:01:00	8:00:23	7:37:37	7:54:21	7:56:00	8:00:29	7:58:06	7:49:31	7:53:30	8:11:07	7:58:55
21 mei 2010	7:52:28	8:02:12	7:55:25	8:07:05	7:54:04	7:32:19	7:41:04	8:00:00	7:54:06	8:04:15	7:54:10	7:39:00	7:39:03	8:00:45	7:55:55
22 mei 2010	6:25:58	8:03:58	8:54:24	8:01:49	7:47:34	6:51:15	7:25:18	8:10:00	6:58:19	6:51:36	8:38:31	6:34:25	7:15:14	8:06:49	9:07:39
24 mei 2010	7:50:08	8:08:05	7:51:15	7:56:56	7:45:28	7:52:00	7:54:37	7:55:26	7:33:17	8:04:45	7:57:38	7:39:56	7:51:35	8:03:38	8:06:16
25 mei 2010	7:45:27	8:10:42	7:50:00	7:58:08	7:58:38	8:15:00	7:40:09	8:00:52	7:53:02	8:00:37	7:55:52	7:00:00	8:00:09	8:00:00	7:54:26
26 mei 2010	7:30:15	8:00:00	8:00:26	7:00:12	8:08:18	7:50:05	8:17:11	7:50:13	7:55:03	8:00:48	8:00:53	7:39:04	7:45:04	7:45:34	7:55:56
27 mei 2010	7:50:05	8:00:42	7:55:29	7:55:57	8:00:00	8:10:00	7:50:08	8:00:35	7:57:53	8:40:40	7:50:12	7:55:25	8:00:12	8:00:58	7:55:10
29 mei 2010	6:35:38	7:00:56	8:00:00	7:50:56	6:50:50	6:30:59	7:57:43	8:00:58	6:50:05	6:45:55	6:30:03	6:35:31	7:00:51	8:15:49	7:00:15
31 mei 2010	7:50:05	8:05:13	7:55:05	8:07:00	7:59:43	8:21:48	7:51:51	7:58:00	7:54:46	8:03:27	7:47:59	7:52:14	7:50:12	8:01:59	7:57:17

Keterangan tabel absensi:

- a. K1 sampai dengan K15 merupakan inisial identitas karyawan (NIK). Dimana k1 merupakan karyawan pertama dan seterusnya hingga karyawan kelima belas.
- b. data lainnya berisi tanggal absensi dan waktu absensi.

4.2.2 *Pre-processing*

Pre-processing data adalah tahapan sebelum suatu data diproses, dapat berupa pembersihan ataupun integrasi data.

1. *Data cleaning* (Pembersihan data)

Proses ini digunakan untuk membuang data yang tidak konsisten dan bersifat *noise* dari data yang terdapat di berbagai basis data dan menangani *missing value*. Contoh data *cleaning* dapat dilihat pada tabel dibawah ini :

Tabel 4.2 Penggalan Tabel Absensi

tanggal	Nama Karyawan			
	K 1	K 2	K 3	k4
10 mei 2010	7:14:04	8:05:45	7:35:12	8:01:43
11 mei 2010	7:50:59	7:34:59	7:58:45	null
12 mei 2010	7:55:06	8:08:05	7:51:00	8:03:47

Dari acuan tabel pada halaman sebelumnya (Tabel 4.1) dapat dilihat data absensi karyawan dengan inisial k4 pada tanggal 11 mei 2010 terdapat *missing value*.

Beberapa cara penanganan *missing value* :

- a. Mengabaikan *recordnya*.

Record data absensi yang bernilai kosong di beri nilai *default*.

- b. Masukkan nilai secara manual

Record data absensi yang bernilai kosong di beri nilai secara manual yaitu dengan memasukkan waktu absensi sesuai dengan pertimbangan waktu kebiasaan karyawan tersebut melakukan absensi.

- c. Menggunakan konstanta umum untuk mengganti nilai yang hilang seperti memberikan nilai tak hingga atau “*unknow*” pada *record* yang kosong.

Penangan *missing value* yang digunakan yaitu memasukkan nilai secara manual dengan melihat waktu kedatangan karyawan pada hari hari sebelumnya atau data di beri nilai *default*. Dikarenakan data absensi merupakan data simulasi absensi jadi tidak terdapat *missing value* pada data absensi.

2. *Data integrity* (integrasi data)

Langkah-langkah yang dilakukan untuk memastikan data telah terintegrasi dengan baik yaitu :

- a. Menentukan *Primary key* pada entitas *database* absensi.

Entitas dari *database* absensi yaitu:

1. Karyawan dengan atribut NIK, nama, tempat lahir, tanggal lahir dan alamat. *Primary key* dari entitas karyawan ini yaitu NIK.
2. Absensi dengan atribut id absen, NIK, tanggal absensi, dan jam. *Primary key* dari entitas absensi yaitu id absen.
3. Kelompok dengan atribut tanggal dan kelompok. Entitas kelompok tidak memiliki *Primary key*.
4. Frekuensi dengan atribut NIK, dan jml. Entitas frekuensi tidak memiliki *Primary key*.
5. *Fptree* dengan atribut tanggal, level1, level2. Entitas *fptree* tidak memiliki *Primary key*.
6. *Fpgrowth* dengan atribut tanggal, id group, nama1, nama2. Entitas *fpgrowth* tidak memiliki *Primary key*.

- b. Menentukan relasi antar entitas.

Relasi dari *database* absensi yaitu:

1. Entitas karyawan memiliki relasi dengan entitas absensi, dengan atribut *foreign key* tabel yaitu NIK.
2. Entitas absensi menghasilkan entitas kelompok.
3. Entitas karyawan memiliki relasi dengan entitas kelompok menghasilkan entitas frekuensi dengan atribut *foreign key* yaitu NIK.
4. Entitas frekuensi memiliki relasi dengan entitas kelompok menghasilkan entitas *fptree*
5. Entitas *fptree* menghasilkan entitas *fpgrowth*.

- c. Atribut untuk *Primary key* bersifat unik, tidak memiliki nilai yang sama.

Atribut dari entitas *database* absensi yaitu:

1. Entitas karyawan dengan *Primary key* NIK bersifat unik dan tidak memiliki nilai yang sama.
2. Entitas absensi dengan *Primary key* id absen bersifat unik dan tidak memiliki nilai yang sama.

- d. Membatasi nilai yang bisa dimasukkan pada suatu atribut. Yaitu *type* dan *length* dari atribut agar data lebih konsisten.
- e. Menentukan spesifikasi dari suatu atribut, apakah atribut tersebut bisa memiliki nilai yang kosong. Untuk entitas absensi terdapat nilai *default* pada atribut waktu absensi.
- f. Mendefenisikan apa yang harus dilakukan bila data baru tidak memiliki data sama sekali pada suatu entitas yang akan disisipkan. Untuk itu dilakukan tahap *data cleaning* yang telah dijelaskan pada poin pertama *preprocessing*.

4.2.3 Transformation

Mengubah / mentransformasikan data ke dalam bentuk yang paling tepat / cocok untuk proses *data mining*. Seperti mentransformasikan data transaksi menjadi *compact transaction* data. Hal ini merupakan tahapan selanjutnya yang dilalui data sebelum data di proses dalam *Association rule*, data absensi akan diubah menjadi data transaksi dengan cara dikelompok-kelompokkan menggunakan metode *maxdiff histogram* berdasarkan waktu absensi karyawan yang berdekatan. Untuk melakukan pengelompokan *user* perlu memasukkan nilai β (*beta*) dalam satuan detik. Nilai *beta* ini harus lebih kecil dari pada waktu terpendek 2 orang melakukan absensi secara bergantian.

a. Analisa Nilai Beta

Untuk mengetahui nilai beta, nilai telah disesuaikan dengan hasil penelitian sebelumnya yaitu hal-hal yang terdapat sebagai berikut : (Setia Budhi, 2010).

1. Secara umum 2 buah data absensi yang berdekatan memiliki selisih waktu lebih dari 10 detik.
2. Bila ada 2 buah data absensi berdekatan memiliki selisih waktu kurang dari 5 detik, nama karyawan yang absen pada 2 data tersebut sama.
3. Beberapa data absensi yang memiliki selisih waktu absen kurang dari 10 detik dan nama karyawan dari kedua data tersebut tidak sama. Kedua karyawan itu, biasanya berada pada lokasi kerja yang sama atau berdekatan.

b. Asumsi nilai beta.

Dari ketiga fakta diatas dapat diasumsikan nilai beta yang disarankan adalah 10 detik. Ada 3 sekenario yang dicoba, yaitu:

1. 1 orang melakukan absensi dengan kartu karyawan yang sama. Langkah yang dilakukan: *Slashing* pertama langsung disusul dengan *Slashing* kedua. Selisih waktu rata-rata yang didapat dibawah 5detik.
2. 1 orang melakukan 2 kali absensi dengan 2 kartu karyawan yang berbeda. Langkah yang dilakukan : *Slashing* dengan kartu yang pertama, menunggu respon perangkat absensi sambil menukar kartu, setelah respon didapat melakukan *Slashing* kedua. Selisih waktu yang didapat diatas 5 detik dan dibawah 10 detik.
3. 2 orang melakukan absensi secara bergantian. Langkah yang dilakukan: karyawan pertama melakukakn *Slashing* dengan kartu karyawannya, menunggu respon dari perangkat absensi, setelah respon didapat segera meninggalkan perangkat absensi dan karyawan kedua maju melakukan *Slashing* dengan kartu karyawannya. Selisih waktu rata-rat yang didapat diats 10 detik.

Berikut adalah tabel hasil pengelompokan data yang telah dikelompokkan dengan frekuensi absensi 10 detik. Karyawan yang memiliki waktu absensi yang berdekatan dengan karyawan lain ditandai dengan kolom yang diwarnai abu-abu. Dari tabel kita bisa melihat selisih waktu absensi karyawan.

Berikut adalah tabel pengelompokan data absensi karyawan berdasarkan waktu kedekatan absensi.

Tabel 4.3 Pengelompokan Absensi

Tanggal	Nama Karyawan														
	K 1	k 2	K 3	K 4	K 5	K 6	K 7	K 8	K 9	K 10	K 11	K 12	K 13	K 14	K 15
1 mei 2010	6:21:01	6:54:14	8:31:05	8:11:24	7:03:56	8:03:59	7:13:09	7:06:25	8:36:10	6:52:05	8:04:03	6:21:04	7:12:50	8:27:22	8:41:25
3 mei 2010	7:35:05	7:56:01	7:45:29	8:00:55	7:44:00	6:44:48	7:56:06	7:56:03	7:58:44	8:01:03	6:55:20	7:45:50	7:48:00	8:16:16	7:33:21
4 mei 2010	7:40:54	8:00:03	8:02:29	7:59:15	7:55:43	8:00:06	7:56:20	8:04:46	7:58:34	8:03:56	7:45:56	7:40:59	7:59:31	7:41:01	7:53:06
5 mei 2010	7:46:12	8:12:53	7:29:33	7:58:58	8:10:20	8:00:03	7:50:02	7:55:55	7:44:00	8:07:40	7:53:30	7:46:15	7:50:09	8:00:05	8:10:43
6 mei 2010	7:47:21	7:23:20	7:48:30	7:54:23	7:51:23	8:00:05	7:50:37	8:01:18	7:58:24	8:01:47	7:49:04	7:47:19	87:50:0	7:49:07	7:10:29
7 mei 2010	7:46:24	8:06:16	7:52:05	8:00:32	8:02:04	7:56:12	8:14:00	8:18:00	7:54:13	7:55:54	7:54:38	7:54:09	7:56:34	7:40:50	7:51:35
8 mei 2010	7:21:12	6:22:27	7:01:25	8:15:33	7:23:56	6:50:13	7:04:00	6:23:07	8:41:31	7:43:22	7:55:09	6:21:38	7:58:21	8:05:29	7:00:46
10 mei 2010	7:14:04	8:05:45	7:35:12	8:01:43	8:08:25	8:08:12	7:45:04	8:07:30	7:53:42	8:04:00	8:26:53	7:13:56	7:48:34	7:38:08	7:45:09
11 mei 2010	7:50:59	7:34:59	7:58:45	8:04:56	7:54:49	7:34:55	7:57:15	7:55:23	7:51:43	7:56:08	7:55:59	7:00:00	7:39:23	7:56:34	7:51:48
12 mei 2010	7:55:06	8:08:05	7:51:00	8:03:47	8:00:37	8:00:10	7:41:35	7:59:05	7:03:00	8:00:45	7:51:21	7:55:10	7:51:51	8:00:26	7:49:01
14 mei 2010	7:33:13	8:00:10	7:58:57	8:07:28	8:01:36	8:00:06	7:50:00	7:36:50	7:54:47	8:03:24	7:54:41	7:33:19	8:06:37	8:03:57	7:44:46
15 mei 2010	6:34:03	9:00:00	7:32:23	8:09:02	8:43:00	8:03:50	7:33:16	8:22:57	7:10:41	8:05:06	8:51:54	6:45:00	7:58:37	7:06:00	7:29:51
17 mei 2010	7:43:56	8:07:44	8:00:11	7:50:48	7:53:05	8:00:05	7:46:21	7:55:45	7:51:15	8:03:43	7:53:54	6:47:01	7:41:51	8:12:20	7:38:26
18 mei 2010	7:51:27	8:01:05	7:58:08	8:03:28	7:51:33	7:15:03	7:57:03	8:05:27	7:53:00	8:07:52	7:59:38	7:59:19	7:58:19	8:08:46	8:01:38
19 mei 2010	7:33:26	8:05:57	7:51:03	8:00:08	7:50:42	7:22:44	7:54:11	8:07:45	7:51:46	8:06:08	7:50:57	7:39:21	8:00:11	7:14:39	7:52:23
20 mei 2010	7:43:27	8:10:09	7:35:00	8:04:12	8:01:00	8:00:23	7:37:37	7:54:21	7:56:00	8:00:29	7:58:06	7:49:31	7:53:30	8:11:07	7:58:55
21 mei 2010	7:52:28	8:02:12	7:55:25	8:07:05	7:54:04	7:32:19	7:41:04	8:00:00	7:54:06	8:04:15	7:54:10	7:39:00	7:39:03	8:00:45	7:55:55
22 mei 2010	6:25:58	8:03:58	8:54:24	8:01:49	7:47:34	6:51:15	7:25:18	8:10:00	6:58:19	6:51:36	8:38:31	6:34:25	7:15:14	8:06:49	9:07:39
24 mei 2010	7:50:08	8:08:05	7:51:15	7:56:56	7:45:28	7:52:00	7:54:37	7:55:26	7:33:17	8:04:45	7:57:38	7:39:56	7:51:35	8:03:38	8:06:16
25 mei 2010	7:45:27	8:10:42	7:50:00	7:58:08	7:58:38	8:15:00	7:40:09	8:00:52	7:53:02	8:00:37	7:55:52	7:00:00	8:00:09	8:00:00	7:54:26
26 mei 2010	7:30:15	8:00:00	8:00:26	7:00:12	8:08:18	7:50:05	8:17:11	7:50:13	7:55:03	8:00:48	8:00:53	7:39:04	7:45:04	7:45:34	7:55:56
27 mei 2010	7:50:05	8:00:42	7:55:29	7:55:57	8:00:00	8:10:00	7:50:08	8:00:35	7:57:53	8:40:40	7:50:12	7:55:25	8:00:12	8:00:58	7:55:10
29 mei 2010	6:35:38	7:00:56	8:00:00	7:50:56	6:50:50	6:30:59	7:57:43	8:00:58	6:50:05	6:45:55	6:30:03	6:35:31	7:00:51	8:15:49	7:00:15
31 mei 2010	7:50:05	8:05:13	7:55:05	8:07:00	7:59:43	8:21:48	7:51:51	7:58:00	7:54:46	8:03:27	7:47:59	7:52:14	7:50:12	8:01:59	7:57:17

Sebelumnya data mentah absensi akan diubah menjadi satuan-satuan transaksi yang disebut item (karyawan). Item ini akan dikelompokkan dengan item lainnya yang terbentuk dari waktu kedekatan absensi karyawan. Item-item yang sekelompok ini disebut *itemset*. Data yang telah di ubah inilah yang digunakan pada proses KDD selanjutnya.

Berikut ini adalah tabel hasil pengelompokan yang di filter dari tabel 4.3

Tabel 4.4 Data hasil pengelompokan

Tanggal	Kelompok dengan b: 10 detik	Tanggal	Kelompok dengan b: 10 detik
01 Mei 2010	{k1,k12}	14 Mei 2010	{k1,k12}
01 Mei 2010	{k6,k11}	14 Mei 2010	{k2,k6}
03 Mei 2010	{k2,k7}	14 Mei 2010	{k9,k11}
03 Mei 2010	{k2,k8}	17 Mei 2010	{k3,k6}
03 Mei 2010	{k7,k8}	18 Mei 2010	{k1,k5}
03 Mei 2010	{k4,k10}	19 Mei 2010	{k3,k11}
04 Mei 2010	{k1,k12}	19 Mei 2010	{k4,k13}
04 Mei 2010	{k2,k6}	20 Mei 2010	{k6,k10}
04 Mei 2010	{k1,k14}	21 Mei 2010	{k5,k9}
04 Mei 2010	{k12,k14}	21 Mei 2010	{k5,k11}
05 Mei 2010	{k1,k12}	21 Mei 2010	{k9,k11}
05 Mei 2010	{k6,k14}	21 Mei 2010	{k12,k13}
05 Mei 2010	{k7,k13}	25 Mei 2010	{k13,k14}
06 Mei 2010	{k1,k12}	26 Mei 2010	{k6,k8}
06 Mei 2010	{k11,k14}	26 Mei 2010	{k10,k11}
7 Mei 2010	{k9,k12}	27 Mei 2010	{k1,k7}
10 Mei 2010	{k1,k12}	27 Mei 2010	{k1,k11}
10 Mei 2010	{k7,k15}	27 Mei 2010	{k7,k11}
11 Mei 2010	{k2,k6}	27 Mei 2010	{k2,k8}
11 Mei 2010	{k9,k15}	27 Mei 2010	{k3,k12}
11 Mei 2010	{k10,k11}	29 Mei 2010	{k1,k12}
12 Mei 2010	{k1,k12}	29 Mei 2010	{k2,k13}
12 Mei 2010	{k5,k10}	31 Mei 2010	{k1,k13}

Keterangan tabel:

b : Batas jarak frekuensi absensi yang dilakukan dua orang karyawan secara berdekatan.

4.2.4 Data Mining

Dua tahapan dalam penggalian frekuensi itemset dalam pencarian kecurangan absensi karyawan, yaitu :

4.2.4.1 Tahap Pembangunan *FP-TREE*

Data absensi yang telah di ubah dalam bentuk data transaksi akan disimpan menggunakan struktur penyimpanan data *FP-Tree*. Seperti penjelasan sebelumnya, diberikan tabel data transaksi hasil pengolompokan dengan data absensi yang memiliki selisih waktu absen kurang dari 10 detik.

Tabel 4.5 Data transaksi mentah dengan nilai beta 10 detik

Nomor	itemset	Nomor	itemset
1	{k1,k12}	24	{k1,k12}
2	{k6,k11}	25	{k2,k6}
3	{k2,k7}	26	{k9,k11}
4	{k2,k8}	27	{k3,k6}
5	{k7,k8}	28	{k1,k5}
6	{k4,k10}	29	{k3,k11}
7	{k1,k12}	30	{k4,k13}
8	{k2,k6}	31	{k6,k10}
9	{k1,k14}	32	{k5,k9}
10	{k12,k14}	33	{k5,k11}
11	{k1,k12}	34	{k9,k11}
12	{k6,k14}	35	{k12,k13}
13	{k7,k13}	36	{k13,k14}
14	{k1,k12}	37	{k6,k8}
15	{k11,k14}	38	{k10,k11}
16	{k9,k12}	39	{k1,k7}
17	{k1,k12}	40	{k1,k11}
18	{k7,k15}	41	{k7,k11}
19	{k2,k6}	42	{k2,k8}
20	{k9,k15}	43	{k3,k12}
21	{k10,k11}	44	{k1,k12}
22	{k1,k12}	45	{k2,k13}
23	{k5,k10}	46	{k1,k13}

Frekuensi kemunculan tiap item (karyawan) dari hasil pengelompokan (Tabel 4.7) data dapat dilihat pada tabel berikut:

Tabel 4.6 Frekuensi kemunculan tiap karakter item

item	frekuensi
k1	13
k2	7
k3	3
k4	2
k5	4
k6	8
k7	6
k8	4
k9	5
k10	5
k11	10
k12	12
k13	6
k14	5
k15	2

Setelah dilakukan tahap pertama didapat karyawan yang memiliki frekuensi di atas *minimum support* =2 adalah k1, k2, k3 , k4, k5, k6, k7, k8, k9, k10, k11, k12, k13, k14, k15. Dalam kasus diatas keseluruhan berpengaruh dan akan dimasukkan ke dalam *FP-Tree*. Tabel berikut mendata kemunculan item yang *frequent* dalam setiap transaksi.

Berikut adalah tabel data transaksi yang telah di *filter* dengan *minimum support* kemunculan karyawan

Tabel 4.7 Data Transaksi min.support

TID	Itemset									
	MS:2	MS:3	MS:4	MS:5	MS:6	MS:7	MS:8	MS:9-10	MS:11-12	MS:13
TID1	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}
TID2	{k11,k6}	{k11,k6}	{k11,k6}	{k11,k6}	{k11,k6}	{k11,k6}	{k11,k6}	{k11,k6}	{k1,k12}	{k1,k12}
TID3	{k2,k7}	{k2,k7}	{k2,k7}	{k2,k7}	{k2,k7}	{k2,k7}	{k1,k12}	{k1,k12}	{k1,k14}	{k1,k14}
TID4	{k2,k8}	{k2,k8}	{k2,k8}	{k2,k8}	{k2,k8}	{k2,k8}	{k1,k14}	{k1,k14}	{k12,k14}	{k1,k12}
TID5	{k10,k4}	{k10,k4}	{k10,k4}	{k10,k4}	{k7,k8}	{k1,k12}	{k6,k2}	{k12,k14}	{k1,k12}	{k1,k12}
TID6	{k7,k8}	{k7,k8}	{k7,k8}	{k7,k8}	{k1,k12}	{k1,k14}	{k12,k14}	{k1,k12}	{k1,k12}	{k1,k12}
TID7	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k14}	{k6,k2}	{k1,k12}	{k1,k12}	{k12,k9}	{k1,k12}
TID8	{k1,k14}	{k1,k14}	{k1,k14}	{k1,k14}	{k6,k2}	{k12,k14}	{k6,k14}	{k11,k14}	{k1,k12}	{k1,k12}
TID9	{k6,k2}	{k6,k2}	{k6,k2}	{k6,k2}	{k12,k14}	{k1,k12}	{k1,k12}	{k12,k9}	{k1,k12}	{k1,k5}
TID10	{k12,k14}	{k12,k14}	{k12,k14}	{k12,k14}	{k1,k12}	{k6,k14}	{k11,k14}	{k1,k12}	{k1,k12}	{k1,k7}
TID11	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k6,k14}	{k1,k12}	{k12,k9}	{k11,k10}	{k1,k5}	{k1,k11}
TID12	{k6,k14}	{k6,k14}	{k6,k14}	{k6,k14}	{k7,k13}	{k11,k14}	{k1,k12}	{k1,k12}	{k12,k13}	{k1,k12}
TID13	{k7,k13}	{k7,k13}	{k7,k13}	{k7,k13}	{k1,k12}	{k12,k9}	{k6,k2}	{k1,k12}	{k1,k7}	{k1,k13}
TID14	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k11,k14}	{k1,k12}	{k11,k10}	{k11,k9}	{k1,k11}	
TID15	{k11,k14}	{k11,k14}	{k11,k14}	{k11,k14}	{k12,k9}	{k6,k2}	{k1,k12}	{k1,k5}	{k12,k3}	
TID16	{k12,k9}	{k12,k9}	{k12,k9}	{k12,k9}	{k1,k12}	{k11,k10}	{k1,k12}	{k11,k3}	{k1,k12}	
TID17	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k7,k15}	{k1,k12}	{k6,k2}	{k11,k5}	{k1,k13}	
TID18	{k7,k15}	{k7,k15}	{k7,k15}	{k7,k15}	{k6,k2}	{k1,k12}	{k11,k9}	{k11,k9}		
TID19	{k6,k2}	{k6,k2}	{k6,k2}	{k6,k2}	{k11,k10}	{k6,k2}	{k6,k3}	{k12,k13}		
TID20	{k9,k15}	{k9,k15}	{k9,k15}	{k9,k15}	{k1,k12}	{k11,k9}	{k1,k5}	{k11,k10}		
TID21	{k11,k10}	{k11,k10}	{k11,k10}	{k11,k10}	{k1,k12}	{k6,k3}	{k11,k3}	{k1,k7}		
TID22	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k6,k2}	{k1,k5}	{k6,k10}	{k1,k11}		
TID23	{k10,k5}	{k10,k5}	{k10,k5}	{k10,k5}	{k11,k9}	{k11,k3}	{k11,k5}	{k12,k3}		
TID24	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}	{k6,k3}	{k6,k10}	{k11,k9}	{k11,k7}		
TID25	{k6,k2}	{k6,k2}	{k6,k2}	{k6,k2}	{k1,k5}	{k11,k5}	{k12,k13}	{k1,k12}		

Tabel 4.7 Data Transaksi min.support (Lanjutan)

TID	Itemset									
	MS:2	MS:3	MS:4	MS:5	MS:6	MS:7	MS:8	MS:9-10	MS:11-12	MS:13
TID26	{k11,k9}	{k11,k9}	{k11,k9}	{k11,k9}	{k11,k3}	{k11,k9}	{k6,k8}	{k1,k13}		
TID27	{k6,k3}	{k6,k3}	{k6,k3}	{k6,k3}	{k13,k4}	{k12,k13}	{k11,k10}			
TID28	{k1,k5}	{k1,k5}	{k1,k5}	{k1,k5}	{k6,k10}	{k6,k8}	{k1,k7}			
TID29	{k11,k3}	{k11,k3}	{k11,k3}	{k11,k3}	{k11,k5}	{k11,k10}	{k1,k11}			
TID30	{k13,k4}	{k13,k4}	{k13,k4}	{k13,k4}	{k11,k9}	{k1,k7}	{k12,k3}			
TID31	{k6,k10}	{k6,k10}	{k6,k10}	{k6,k10}	{k12,k13}	{k1,k11}	{k11,k7}			
TID32	{k9,k5}	{k9,k5}	{k9,k5}	{k9,k5}	{k13,k14}	{k2,k8}	{k1,k12}			
TID33	{k11,k5}	{k11,k5}	{k11,k5}	{k11,k5}	{k6,k8}	{k12,k3}	{k1,k13}			
TID34	{k11,k9}	{k11,k9}	{k11,k9}	{k11,k9}	{k11,k10}	{k11,k7}				
TID35	{k12,k13}	{k12,k13}	{k12,k13}	{k12,k13}	{k1,k7}	{k1,k12}				
TID36	{k13,k14}	{k13,k14}	{k13,k14}	{k13,k14}	{k1,k11}	{k2,k13}				
TID37	{k6,k8}	{k6,k8}	{k6,k8}	{k6,k8}	{k2,k8}	{k1,k13}				
TID38	{k11,k10}	{k11,k10}	{k11,k10}	{k11,k10}	{k12,k3}					
TID39	{k1,k7}	{k1,k7}	{k1,k7}	{k1,k7}	{k11,k7}					
TID40	{k1,k11}	{k1,k11}	{k1,k11}	{k1,k11}	{k1,k12}					
TID41	{k2,k8}	{k2,k8}	{k2,k8}	{k2,k8}	{k2,k13}					
TID42	{k12,k3}	{k12,k3}	{k12,k3}	{k12,k3}	{k1,k13}					
TID43	{k11,k7}	{k11,k7}	{k11,k7}	{k11,k7}						
TID44	{k1,k12}	{k1,k12}	{k1,k12}	{k1,k12}						
TID45	{k2,k13}	{k2,k13}	{k2,k13}	{k2,k13}						
TID46	{k1,k13}	{k1,k13}	{k1,k13}	{k1,k13}						

Keterangan Tabel :

TID : Transaksi id (Nomor transaksi)

MS : Frekuensi kemunculan karyawan dengan karyawan yang sama maupun yang berbeda

Itemset : Hasil pengelompokan karyawan berdasarkan MS yang ditentukan.

FP-Tree yang merepresentasikan data transaksi pada tabel 4.7 dengan minimum support kemunculan karyawan 2 kali dibentuk dengan cara sebagai berikut:

1. Kumpulan data dipindai pertama kali untuk menentukan *frekuensi* kemunculan dari setiap karyawan. karyawan yang tidak *frequent* dibuang, sedangkan karyawan yang memenuhi batas frekuensi kemunculan dimasukkan dan disusun dengan urutan menurun, seperti yang terlihat pada tabel 4.9. berikut hasil pembacaan transaksi dengan min support 2.
2. Selanjutnya, yaitu pembacaan TID pertama {K1,K12} akan membuat simpul K1 dan K12, sehingga terbentuk lintasan transaksi $\text{Null} \rightarrow \text{K1} \rightarrow \text{K12}$. *Support count* dari setiap simpul bernilai awal 1
3. Setelah pembacaan transaksi kedua {k11,k6} terbentuk lintasan kedua yaitu $\text{Null} \rightarrow \text{K11} \rightarrow \text{K6}$. *Support count* masing-masing *count* juga bernilai awal 1.
4. Pembacaan transaksi ketiga {k2,k7} terbentuk lintasan ketiga yaitu $\text{Null} \rightarrow \text{K2} \rightarrow \text{K7}$. *Support count* masing-masing *count* juga bernilai awal 1.
5. Pembacaan transaksi keempat {k2,k8} memiliki *prefix* transaksi yang sama dengan transaksi ketiga, yaitu k2, maka lintasan transaksi lintasan transaksi keempat dapat ditimpakan di k2, dan selanjutnya membuat lintasan yang baru sesuai transaksi keempat.
6. Proses pembacaan transaksi kelima dan keenam hampir sama dengan transaksi pertama dan kedua dengan membuat lintasan yang baru dikarenakan *prefix* lintasan berbeda dengan lintasan yang telah ada.
7. Pembacaan TID ketujuh {K1,K12} memiliki lintasan yang sama dengan transaksi yang pertama $\text{Null} \rightarrow \text{K1} \rightarrow \text{K12}$. *Support count* dari setiap simpul bertambah dengan nilai 2
8. Proses pembacaan TID kedelapan sama dengan proses pada transaksi keempat.
9. Proses pembacaan TID kesepuluh {k12,k14}, terbentuk lintasan baru walaupun k12 ada pada transaksi pertama, namun karena prefix transaksinya

tidak sama, maka transaksi kesepuluh tidak bias dimapatkan pada transaksi pertama.

10. Proses ini dilanjutkan sampai *FP-Tree* berhasil dibangun berdasarkan tabel data transaksi yang diberikan.

Representasi untuk transaksi dengan *minimum support* selanjutnya memiliki tahapan yang sama dengan representasi dengan nilai *minimum support* dua yang telah dipaparkan diatas (Tabel 4.7).

Tabel dibawah ini memberikan ilustrasi mengenai pembentukan *FP-Tree* setelah pembacaan TID1 hasil representasi pada data transaksi dengan minimum support dua.

Tabel 4.8 Hasil pembangunan *FP-Tree*

TID	Itemset	Hasil pembentukan <i>FP-Tree</i>
TID1	{k1,k12}	
TID2	{k11,k6}	
TID3	{k2,k7}	
TID4	{k2,k8}	

Tabel 4.8 Hasil pembangunan *FP-Tree* (Lanjutan)

TID	Itemset	Hasil pembentukan <i>FP-Tree</i>
TID5	{k10,k4}	
TID6	{k7,k8}	
TID7	{k1,k12}	
TID8	{k1,k14}	

Tabel 4.8 Hasil pembangunan *FP-Tree* (Lanjutan)

TID	Itemset	Hasil pembentukan <i>FP-Tree</i>
TID9	{k6,k2}	
TID10	{k12,k14}	
TID11	{k1,k12}	
TID12	{k6,k14}	

Tabel 4.8 Hasil pembangunan *FP-Tree* (Lanjutan)

TID	Itemset	Hasil pembentukan <i>FP-Tree</i>
TID13	{k7,k13}	
TID14	{k1,k12}	
TID15	{k11,k14}	
TID16	{k12,k9}	

Tabel 4.8 Hasil pembangunan FP-Tree (Lanjutan)

TID	Itemset	Hasil pembentukan FP-Tree
TID17	{k1,k12}	
TID18	{k7,k15}	
TID19	{k6,k2}	
TID20	{k9,k15}	

Tabel 4.8 Hasil pembangunan *FP-Tree* (Lanjutan)

TID	Itemset	Hasil pembentukan <i>FP-Tree</i>
TID21	{k11,k10}	
TID22	{k1,k12}	
TID23	{k10,k5}	
TID24	{k1,k12}	

Tabel 4.8 Hasil pembangunan *FP-Tree* (Lanjutan)

TID	Itemset	Hasil pembentukan <i>FP-Tree</i>
TID25	{k6,k2}	
TID26	{k11,k9}	
TID27	{k6,k3}	
TID28	{k1,k5}	

Tabel 4.8 Hasil pembangunan *FP-Tree* (Lanjutan)

TID	Itemset	Hasil pembentukan <i>FP-Tree</i>
TID29	{k11,k3}	
TID30	{k13,k4}	
TID31	{k6,k10}	
TID32	{k9,k5}	

Tabel 4.8 Hasil pembangunan *FP-Tree* (Lanjutan)

TID	Itemset	Hasil pembentukan <i>FP-Tree</i>
TID33	{k11,k5}	
TID34	{k11,k9}	
TID35	{k12,k13}	
TID36	{k13,k14}	

Tabel 4.8 Hasil pembangunan *FP-Tree* (Lanjutan)

TID	Itemset	Hasil pembentukan <i>FP-Tree</i>
TID37	{k6,k8}	
TID38	{k11,k10}	
TID39	{k1,k7}	

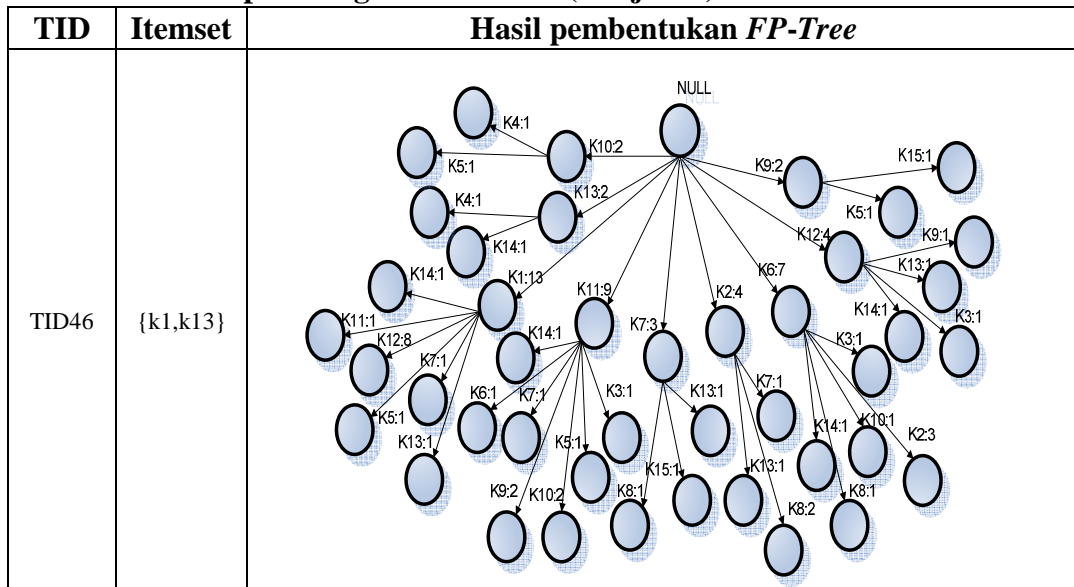
Tabel 4.8 Hasil pembangunan *FP-Tree* (Lanjutan)

TID	Itemset	Hasil pembentukan <i>FP-Tree</i>
TID40	{k1,k11}	
TID41	{k2,k8}	
TID42	{k12,k3}	

Tabel 4.8 Hasil pembangunan FP-Tree (Lanjutan)

TID	Itemset	Hasil pembentukan FP-Tree
TID43	{k11,k7}	
TID44	{k1,k12}	
TID45	{k2,k13}	

Tabel 4.8 Hasil pembangunan *FP-Tree* (Lanjutan)



Hasil ilustrasi pembangunan *FP-Tree* untuk keseluruhan transaksi dapat dilihat pada lampiran A.

4.2.4.2 Penerapan algoritma *FP-Growth*

Setelah tahap pembangunan *FP-Tree* dari sekumpulan data transaksi, akan diterapkan algoritma *FP-Growth* untuk mencari *frequent itemset* yang signifikan. Algoritma *FP-Growth* dibagi menjadi tiga langkah utama, yaitu :

1. Tahap Pembangkitan *Conditional Pattern Base*

Conditional Pattern Base merupakan subdatabase yang berisi *prefix path* (lintasan prefix) dan *suffix pattern* (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui *FP-Tree* yang telah dibangun sebelumnya. Algoritma *FP-Growth* menemukan frequent itemset yang berakhiran *suffix* menggunakan metode *divide and conquer* untuk memecah problem menjadi subproblem yang lebih kecil.

Jika kita ingin menemukan semua frekuensi itemset yang berakhiran item k, hal yang pertama dilakukan adalah pembangkitan conditional pattern base pada sub database yang berisikan *prefix path* (himpunan item terurut yang mengawali k-itemsets), dan *suffix pattern* (k-itemsets). Menghitung

support count dari k-itemset, jika memenuhi maka item k termasuk *frequent*. Karena item k *frequent*, maka perlu di pecah ke subproblem selanjutnya yaitu menemukan item k yang *frequent* dengan penggabungan item lainnya. Dari pemecahan problem yang ada, maka ditemukan himpunan itemset yang berakhiran k.

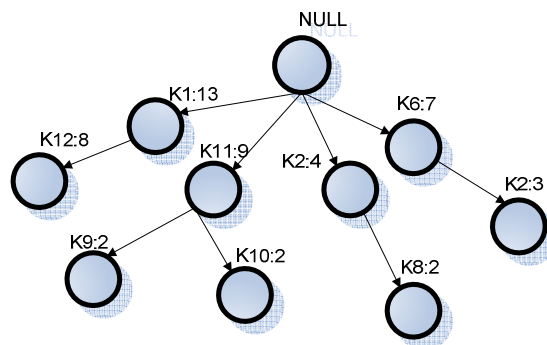
Dalam langkah ini akan dicari karyawan yang muncul bersamaan dengan menghitung nilai *support count* dari itemset. Dalam hal ini nilai *support count* dengan *minimum support* dengan nilai dua yang di didapat dari pemecahan tabel (tabel 4.8) setelah pembacaan transaksi yang terakhir.

Itemset yang muncul bersamaan :

Tabel 4.9 Itemset yang memenuhi *minimum support count*

Nomor	Itemset
1	{k1,k12}
2	{k1,k12}
3	{k1,k12}
4	{k1,k12}
5	{k1,k12}
6	{k1,k12}
7	{k1,k12}
8	{k1,k12}
9	{k11,k9}
10	{k11,k9}
11	{k11,k10}
12	{k11,k10}
13	{k2,k8}
14	{k2,k8}
15	{k6,k3}
16	{k6,k3}
17	{k6,k3}

Berikut adalah ilustrasi dari hasil pencarian itemset yang muncul bersamaan yang didapat dari pemecahan *FP-Tree* yang telah dibangun sebelumnya.



Gambar 4.1 Lintasan yang mengandung itemset yang bersamaan

2. Tahap Pembangkitan *Conditional FP-Tree*

Pada tahap ini, *support count* dari setiap item pada setiap *conditional pattern base* dijumlahkan, lalu setiap item yang memiliki jumlah *support count* lebih besar sama dengan *minimum support count* ξ akan dibangkitkan dengan *conditional FP-Tree*. Sehingga didapat *conditional FP-Tree* yang terdiri dari empat buah lintasan (Gambar 4.1), dimana 2 diantaranya memiliki *prefix* transaksi yang sama. Lintasan itu diantaranya $\{k1, k12\}$, $\{k2, k6\}$, $\{k2, k8\}$, $\{k9, k11\}$ dan $\{k10, k11\}$.

Menghitung *support count* pada setiap lintasan (Gambar 4.1) :

1. Lintasan $\{k1, k12\}$, *suffix* (pola akhir) pada lintasan ini yaitu $k12$ dengan nilai *support count* 8. Sehingga *prefix* dari $k1$ memiliki nilai *support count* 8 dan selebihnya dapat dibuang.

$K12: 8$ menggambarkan bahwa ada *pattern* $k1-k12$ sebanyak 8 kali

2. Selanjutnya lintasan $\{k11, k9\}$ dan $\{k11, k10\}$, lintasan ini memiliki *prefix* transaksi yang sama yaitu $k11$. Sehingga nilai *support count* dari $k11$ di dapat dari penjumlahan nilai *support count suffix* (pola akhir) pada dua lintasan yang terbentuk setelahnya yaitu $k9$ dan $k10$. *Support count* dari $k9$ bernilai 2 dan $k10$ bernilai 2, jadi jumlah *support count* dari $k2$ bernilai 4 dan selebihnya dapat dibuang.

$K9: 2$ menggambarkan bahwa ada *pattern* $k11-k9$ sebanyak 2 kali, dan

$K10: 2$ menggambarkan bahwa ada *pattern* $k11-k10$ sebanyak 2 kali

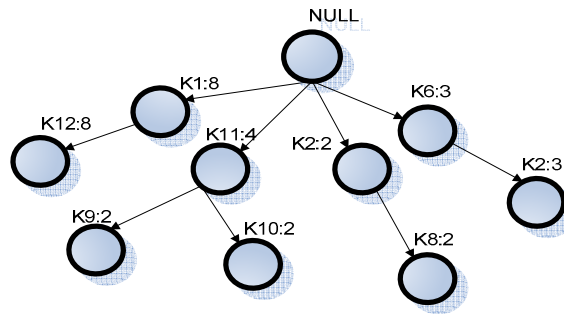
- Lintasan {k2,k8}, lintasan ini merupakan lintasan tunggal nilai *prefix* dan *suffix* pada lintasan ini bernilai sama.

K8:2 menggambarkan bahwa ada *pattern* k2-k8 sebanyak 2 kali

- Lintasan {k6,k2}, lintasan ini merupakan lintasan tunggal nilai *prefix* dan *suffix* pada lintasan ini bernilai sama.

K2:3 menggambarkan bahwa ada *pattern* k6-k2 sebanyak 3 kali.

Berikut adalah ilustrasi dari hasil perhitungan nilai *support count* pada setiap itemset.



Gambar 4.2 Lintasan itemset yang mengandung *support count* yang sama

- Tahap Pencarian *frequent itemset*

Dikarenakan *Conditional FP-Tree* merupakan lintasan tunggal (*single path*), maka didapatkan *frequent itemset* dengan melakukan kombinasi itemset untuk setiap *conditional FP-Tree*. Jika bukan lintasan tunggal dilakukan pembangkitan *FP-Tree* secara rekursif.

Hasil dari perhitungan algoritma *FP-Growth* dengan *minimum support count* dua yang didapat dari tahapan sebelumnya:

Tabel 4.10 Nilai *support count* masing-masing itemset

rule	<i>Support count</i>
k1->k12	8
K11-> k9	2
K11-> k10	2
K2->k8	2
K6->k2	3

Jadi transaksi {k1,k12}, {k11,k9}, {k11,k10}, {k2,k8} dan {K6,k2} memenuhi *minimum support* yang telah ditentukan.

Transaksi dengan *minimum support count* 3 hingga 8 (lampiran B)

4.2.5 Interpretation / Evaluation

Penerjemahan pola-pola yang dihasilkan dari *data mining*. Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan.

Dari hasil pencarian frekuensi kemunculan karyawan dapat disimpulkan, ada lima lintasan yang memenuhi *minimum support* yaitu lintasan {k1,k12}, lintasan {k11,k9},{k11,k10}, lintasan {k2,k8} dan lintasan {k6,k2} . Lintasan {k1,k12} memiliki jumlah frekuensi kemunculan sebanyak 8 kali, {k11,k9} memiliki jumlah frekuensi kemunculan sebanyak 2 kali, {k11,k10} memiliki jumlah frekuensi kemunculan sebanyak 2 kali, {k2,k8} memiliki jumlah frekuensi kemunculan sebanyak 2 kali, {k2,k8} memiliki jumlah frekuensi kemunculan sebanyak 3 kali. Karyawan dengan inisial k1 dan k12 lebih sering melakukan absensi secara berdekatan dibandingkan karyawan lainnya sehingga memungkinkan karyawan dengan inisial k1 dengan k12 melakukan absensi secara berbarengan dan memungkinkan melakukan penitipan absensi.

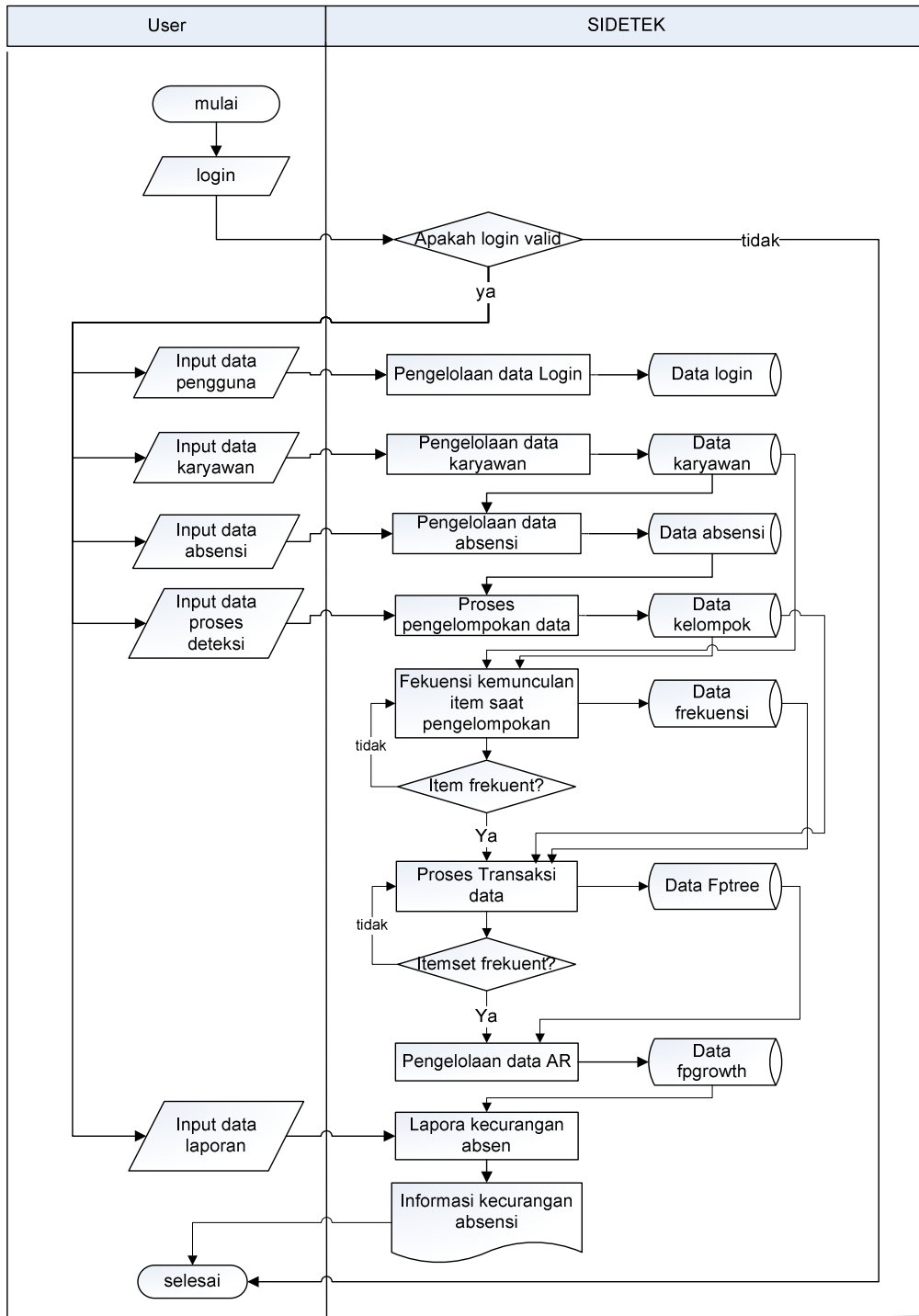
4.3 Perancangan Sistem

Sistem yang akan dirancang yaitu perancangan basis data, struktur menu dan *interface*. Dari proses data masukan hingga menghasilkan data keluaran akan digambarkan melalui Flowchart sistem, Diagram Kontek/*Data Context Diagram* (DCD), Diagram Aliran Data/*Data Flow Diagram* (DFD), Kamus Data/*Data Dictionary* (DD), dan *entity relationship diagram* (ERD).

Selanjutnya, untuk mempermudah penggunaan sistem perlu dirancang suatu antar muka (*Interface*). Hal-hal yang perlu dirancang dalam antarmuka sistem ini adalah rancangan *input* dan rancangan *outputnya*.

4.3.1 Flowchart Sistem

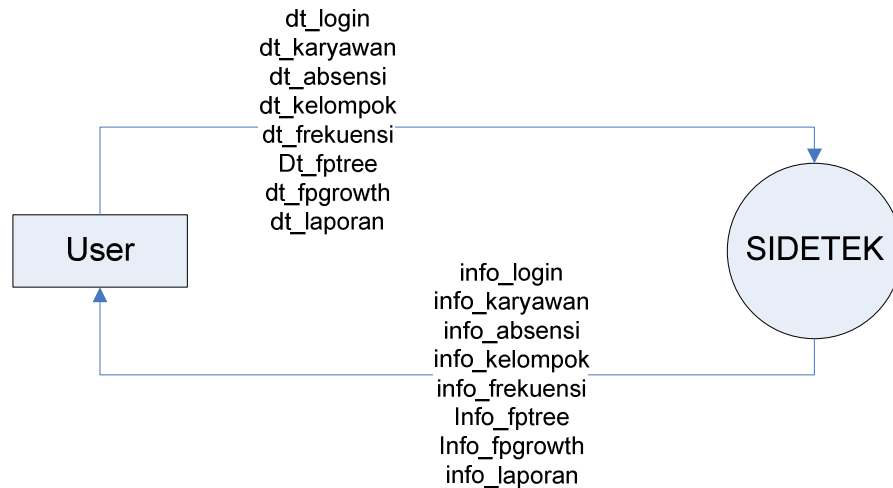
Proses-proses yang terjadi pada sistem deteksi kecurangan absensi bisa digambarkan dengan menggunakan *flowchart* dibawah ini



Gambar 4.3 Flowchart Sistem

4.3.2 Diagram Konteks (*Context Diagram*)

Diagram konteks digunakan untuk menggambarkan sistem secara garis besar dari aplikasi data *mining*. Seperti gambar yang dibawah ini :



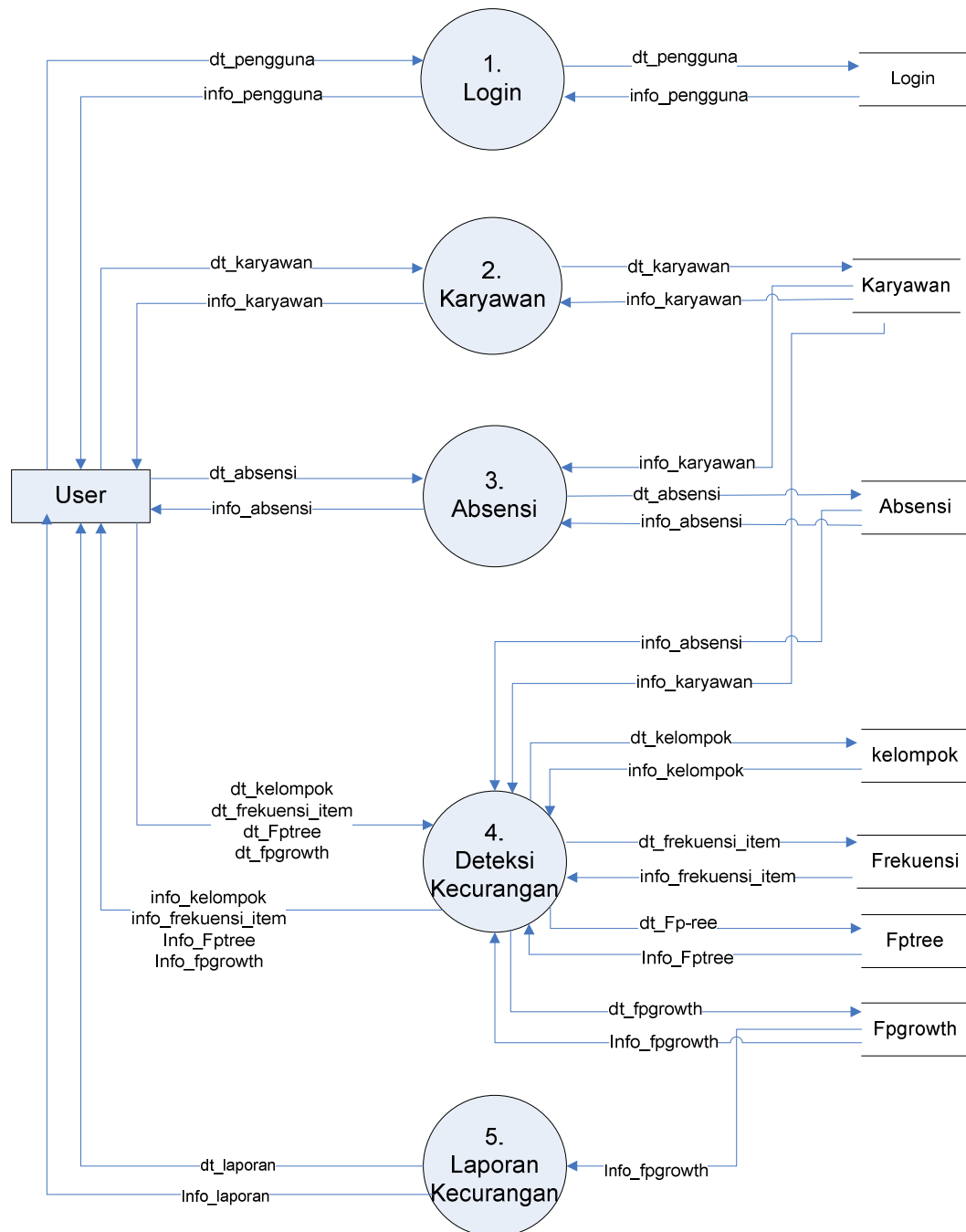
Gambar 4.4 *Contect Diagram*

Entitas yang berinteraksi dengan sistem adalah *user* memiliki peran antara lain:

- Memasukkan data karyawan
- Memasukkan data absensi
- Proses pengelompokan data, dengan menentukan batasan waktu kedekatan absensi antar karyawan dan menentukan batas tanggal absensi yang akan di uji.
- Setelah proses pengelompokan data selesai, maka akan diketahui frekuensi kemunculan sebuah item dari hasil pengelompokan.
- Menyimpan data absensi yang telah di kelompokkan dengan struktur penyimpanan data *FP-Tree* dan Menetapkan *minimum support* kemunculan absensi karyawan yang melakukan absen secara bersamaan.
- Memasukkan data *association rule* dengan penerapan algoritma *FP-Growth*.
- Mendapatkan informasi data karyawan yang mungkin melakukan kecurangan absensi.

4.3.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) sering digunakan untuk menggunakan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir, atau lingkungan fisik dimana data tersebut tersimpan.



Gambar 4.5 DFD Level 1

Tabel 4.11 Proses DFD Level 1 SIDETEK

Nama	Deskripsi
Login	Berisi data-data pengguna system
Karyawan	Berisi data-data karyawan
Absensi	Berisi data absensi karyawan
Deteksi Kecurangan	Berisi proses dalam pencarian frekuensi kecurangan karyawan dalam absensi
Laporan Kecurangan	Berisi laporan kecurangan absensi karyawan

Tabel 4.12 Aliran Data DFD Level 1 SIDETEK

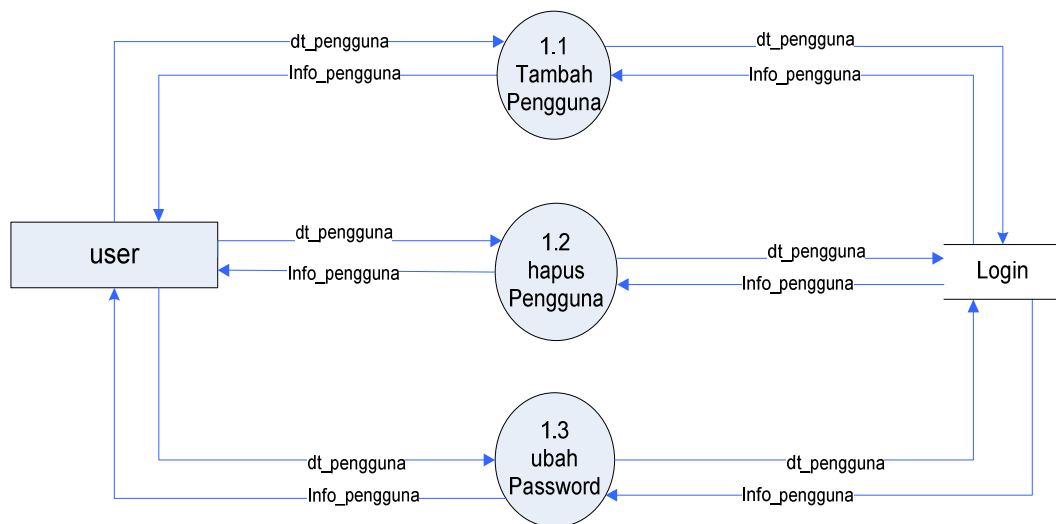
Nama	Deskripsi
dt_pengguna	Data Pengguna
dt_karyawan	Data Karyawan
dt_absensi	Data absensi karyawan
dt_kelompok	Data item yang telah dikelompokkan
dt_frekuensi	Data frekuensi kemunculan item
dt_FPTree	Data struktur penyimpanan data <i>FPTree</i>
dt_FPGrowth	Data <i>Association rule</i> kecurangan absensi
Dt_laporan	Data laporan kecurangan absensi
info_pengguna	Informasi mengenai data pengguna
info_karyawan	Informasi mengenai data karyawan
info_absensi	Informasi mengenai data absensi
info_kelompok	Informasi mengenai data itemset
info_frekuensi	Informasi frekuensi kemunculan item

Tabel 4.12 Aliran Data DFD Level 1 SIDETEK (Lanjutan)

Nama	Deskripsi
Info_FPTree	Informasi fasilitas penyimpanan data FPTree
info_FPGrowth	Informasi mengenai Association rule kecurangan absensi
Info_laporan	Informasi laporan kecurangan absensi

4.3.3.1 DFD Level 2 Proses 1 Login

Berikut adalah gambar DFD Level 2 Proses 1 Login



Gambar 4.6 DFD Level 2 Proses 1 Login

Tabel 4.13 Proses DFD Level 2 Proses 1 Login

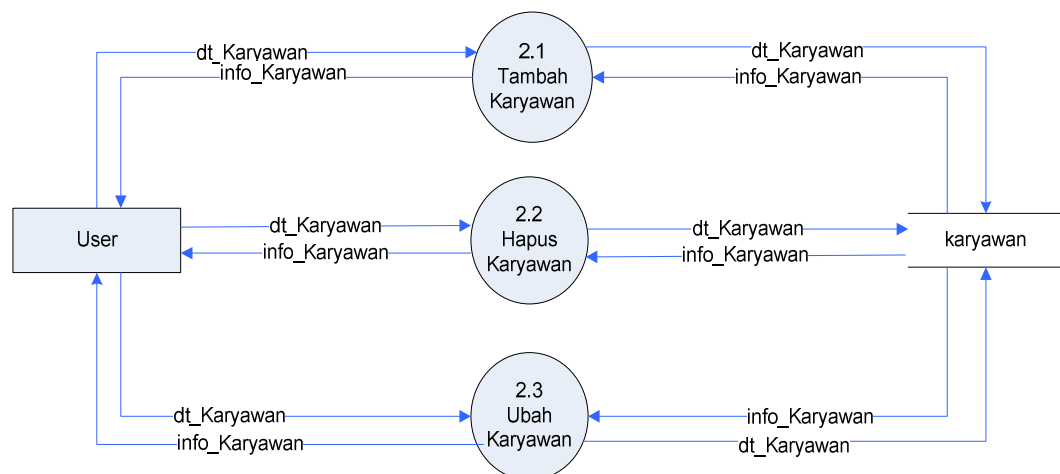
Nama	Deskripsi
tambah pengguna	Berisi proses untuk penambahan data pengguna
Hapus pengguna	Berisi proses untuk menghapus data pengguna
Ubah pengguna	Berisi proses untuk mengubah data pengguna

Tabel 4.14 Aliran DFD Level 2 Proses 1 Login

Nama	Deskripsi
Dt_ pengguna	Perubahan data pengguna yang telah mengalami penambahan, pengubahan dan penghapusan
Info_ pengguna	Informasi perubahan data pengguna yang telah mengalami penambahan, pengubahan dan penghapusan

4.3.3.2 DFD Level 2 Proses 2 Karyawan

Berikut adalah gambar DFD Level 2 Proses 2 Karyawan



Gambar 4.7 DFD Level 2 Proses 2 Karyawan

Tabel 4.15 Proses DFD Level 2 Proses 2 Karyawan

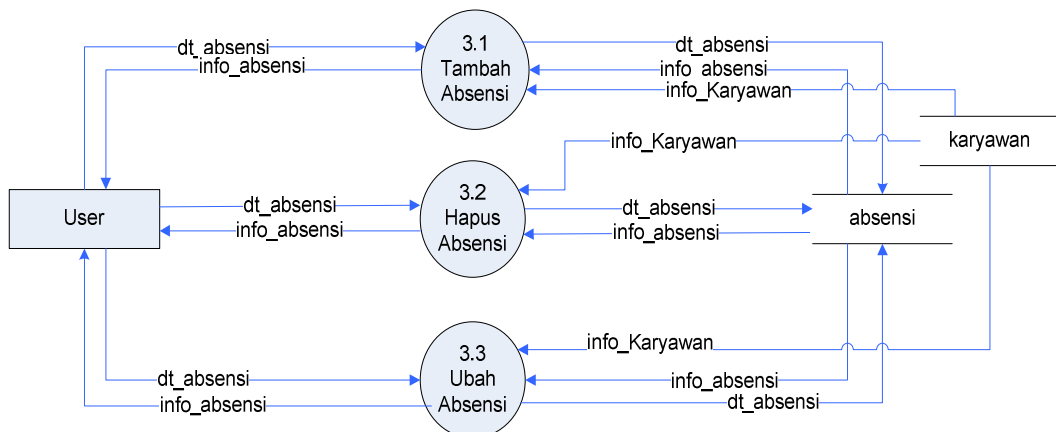
Nama	Deskripsi
tambah karyawan	Berisi proses untuk penambahan data karyawan
Hapus karyawan	Berisi proses untuk menghapus data karyawan
Ubah karyawan	Berisi proses untuk mengubah data karyawan

Tabel 4.16 Aliran DFD Level 2 Proses 2 Karyawan

Nama	Deskripsi
Dt_ karyawan	Perubahan data karyawan yang telah mengalami penambahan, pengubahan dan penghapusan
Info_ karyawan	Informasi perubahan data karyawan yang telah mengalami penambahan, pengubahan dan penghapusan

4.3.3.3 DFD Level 2 Proses Absensi

Berikut adalah gambar DFD Level 2 Proses 3 Absensi



Gambar 4.8 DFD Level 2 Proses 3 Absensi

Tabel 4.17 Proses DFD Level 2 Proses 3 absensi

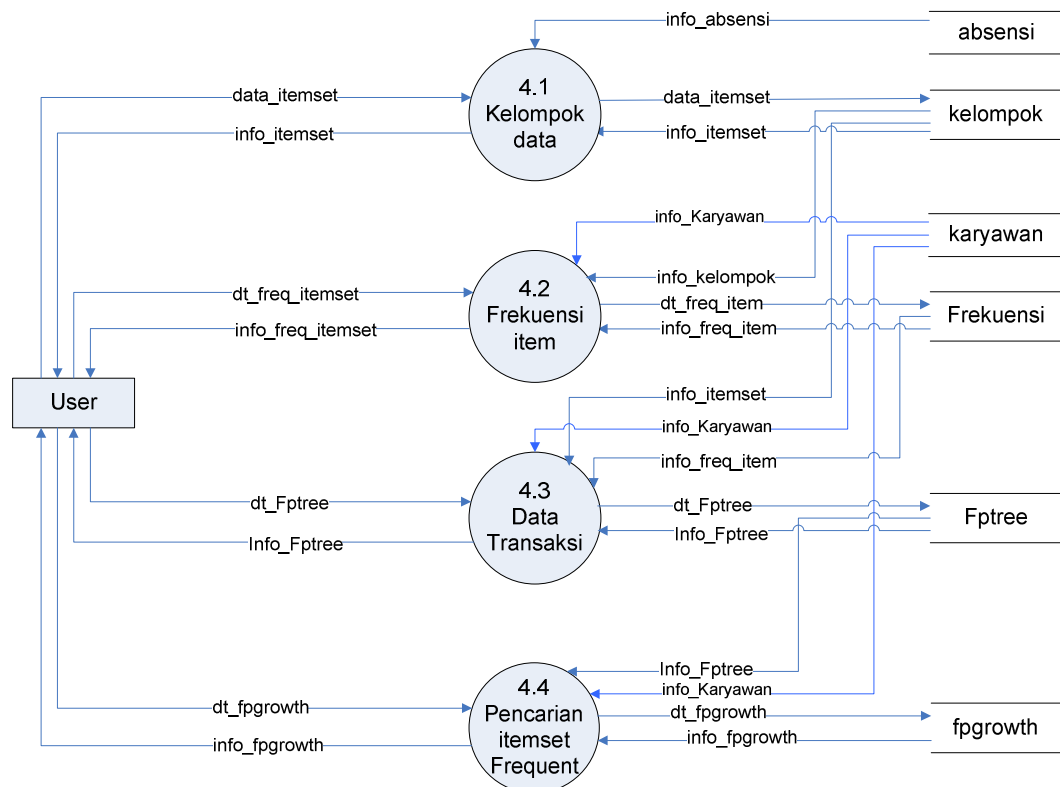
Nama	Deskripsi
Tambah absensi	Berisi proses untuk penambahan data absensi
Hapus absensi	Berisi proses untuk penghapusan data absensi
Ubah absensi	Berisi proses untuk pengubahan data absensi

Tabel 4.18 Aliran DFD Level 2 Proses 3 absensi

Nama	Deskripsi
Dt_absensi	Perubahan data absensi yang telah mengalami penambahan, pengubahan dan penghapusan
Info_absensi	Informasi perubahan data absensi yang telah mengalami penambahan, pengubahan dan penghapusan
Info_karyawan	Informasi data karyawan yang dibutuhkan dalam pengelolaan data absensi

4.3.3.4 DFD Level 2 Proses 4 Deteksi Kecurangan

Berikut adalah gambar DFD Level 2 Proses 4 Deteksi Kecurangan absensi karyawan



Gambar 4.9 DFD Level 2 proses 4 Kecurangan Absensi

Tabel 4.19 Proses DFD Level 2 Proses 4 Deteksi kecurangan absensi

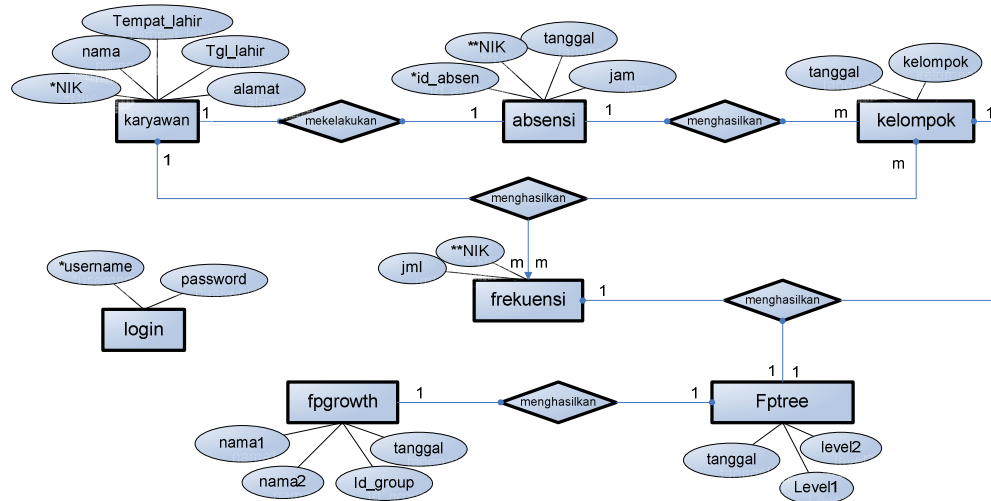
Nama	Deskripsi
Kelompok data	Berisi proses pengelompokan item menjadi itemset
Frekuensi item	Berisi proses frekuensi kemunculan item saat terjadinya pengelompokan data
Data transaksi	Berisi proses kelompok data yang telah diubah menjadi transaksi data sesuai dengan nilai <i>minimum support</i> yang ditentukan
Pencarian frekuensi itemset	Berisi proses transaksi data dengan frekuensi kemunculan data secara bersamaan

Tabel 4.20 Proses DFD Level 2 Proses 4 Deteksi Kecurangan

Nama	Deskripsi
dt_kelompok	Data itemset
dt_frekuensi	Data frekuensi kemunculan item
dt_ <i>FP-Tree</i>	Data transaksi dalam struktur <i>FP-Tree</i>
dt_AR	Data kecurangan absensi
info_kelompok	Informasi itemset
info_karyawan	Informasi mengenai karyawan
info_absensi	Informasi mengenai absensi
info_frekuensi	Informasi frekuensi kemunculan item
info_ <i>FP-Tree</i>	Informasi transaksi dalam struktur <i>FP-Tree</i>
info_AR	Informasi kecurangan absensi

4.3.4 Entity Relationship Diagram (ERD)

Gambar 5.14 adalah gambar ERD dari sistem informasi deteksi kecurangan absensi (SIDETEK)



Gambar 4.10 ERD SIDETEK

Tabel 4.21 Keterangan Entitas pada ERD

No	Nama	Deskripsi	Atribut	Primary key
1.	karyawan	Menyimpan data karyawan	- NIK - nama - Tempat lahir - tgl_lahir - alamat	NIK
2.	Absensi	Menyimpan data absensi	- id_absen - NIK - tanggal - jam	No_absen
3.	Kelompok	Menyimpan data itemset	- tanggal - kelompok	
4.	Frekuensi	Menyimpan data frekuensi kemunculan karyawan	- NIK - jml	
5.	<i>FPTree</i>	Menyimpan data itemset yang frequent	- Level1 - Level 2	

			- Tanggal	
6.	<i>FPGrowth</i>	Rule kecurangan Absensi	- Nama1 - Nama2 - Tanggal - id_group	
7.	<i>login</i>	Data login	- username - password	username

4.3.5 Perancangan Tabel Basis Data

4.3.5.1 Rancangan Tabel Data Login

Tabel berikut ini menjelaskan tabel login

Nama : login

Deskripsi : Berisi data pengguna

Primary key : username

Tabel 4.22 Basis Data login

Nama Field	type dan length	Deskripsi	Null	Default
username	varchar(30)	Nama pengguna	Yes	-
password	varchar(30)	Password pengguna	Yes	-

4.3.5.2 Rancangan Tabel Data karyawan

Tabel berikut ini menjelaskan tabel karyawan

Nama : karyawan

Deskripsi : Berisi data karyawan

Primary key : NIK

Tabel 4.23 Basis Data karyawan

Nama Field	type dan length	Deskripsi	Null	Default
NIK	varchar(10)	Nomor induk karyawan	Yes	-
nama	varchar(30)	Nama karyawan	Yes	-
Tempat_lahir	varchar(30)	Alamat karyawan	Yes	-
Tgl_lahir	Date	No telepon karyawan	Yes	-
Alamat	varchar(50)	Alamat Karyawan	Yes	-

4.3.5.3 Rancangan Tabel Data Absensi

Tabel berikut ini menjelaskan tabel absensi

Nama : absensi

Deskripsi : Berisi data absensi karyawan

Primary key : Id_absen

Tabel 4.24 Basis Data Absensi

Nama Field	Type dan Length	Deskripsi	Null	Default
Id_absen	int(4)	Nomor absensi karyawan	No	-
NIK	varchar(10)	Nomor induk karyawan	No	-
tanggal	Date	Tanggal absensi karyawan	Yes	-
jam	Time	Waktu absensi karyawan	Yes	-

4.3.5.4 Rancangan Tabel Kelompok

Tabel berikut ini menjelaskan tabel pengelompokan

Nama : Kelompok

Deskripsi : Berisi data itemset

Primary key : -

Tabel 4.25 Basis Data kelompok

Nama Field	Type dan Length	Deskripsi	Null	Default
kelompok	varchar(10)	Nomor absensi karyawan	Yes	-
tanggal	Date	Tanggal absensi	Yes	-

4.3.5.5 Rancangan Tabel Frekuensi

Tabel berikut ini menjelaskan frkuensi kemunculan item dari hasil pengelompokan.

Nama : Frekuensi

Deskripsi : Berisi data Frekuensi kemunculan karyawan saat proses pengelompokan data

Primary key :-

Tabel 4.26 Basis data frekuensi item

Nama Field	Type dan Length	Deskripsi	Null	Default
NIK	char(10)	Nomor induk karyawan	Yes	-
Jml	tinyint(3)	Jumlah Frekuensi kemunculan karyawan saat pengelompokan	Yes	-

4.3.5.6 Rancangan Tabel *FP-Tree*

Tabel berikut ini menjelaskan tabel *FPTree*

Nama : *FPTree*

Deskripsi : Berisi data Itemset Frequent

Primary key :-

Tabel 4.27 Basis data *FP-Tree*

Nama Field	Type dan Length	Deskripsi	Null	Default
tanggal	date	Tanggal absensi	Yes	-
Level1	char(5)	Lintasan pertama <i>FP-Tree</i>	Yes	-
Level2	char(5)	Lintasan kedua <i>FP-Tree</i>	Yes	-

4.3.5.7 Rancangan Tabel *FPGrowth*

Tabel berikut ini menjelaskan tabel *FPGrowth*

Nama : *FPGrowth*

Deskripsi : prediksi kecurangan absensi karyawan

Primary key : -

Tabel 4.28 Basis Data *FP-Growth*

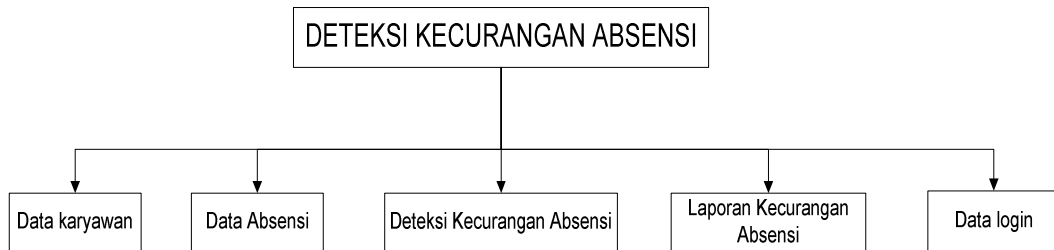
Nama Field	Type dan Length	Deskripsi	Null	Default
nama1	varchar(30)	Nama Karyawan	Yes	-
nama2	varchar(10)	Nama Karyawan	Yes	-
tanggal	Date	Tanggal absensi	Yes	-
id_group	Int(2)	Nomor group	Yes	

4.4 Perancangan Subsystem Dialog

Merancang subsystem dialog berupa tampilan menu sistem yang *user friendly* sehingga *user* paham dalam menggunakan atau memilih menu-menu pilihan yang terdapat pada sistem.

4.4.1 Struktur Menu

Berikut ini merupakan gambar struktur menu SPK pemilihan karyawan terbaik. Menu system terdiri dari enam menu dan setiap menu memiliki sub-sub menu.



Gambar 4.11 Struktur menu Deteksi Kecurangan Absensi

4.4.2 User Interface (Perancangan Antar Muka Sistem)

Perancangan antar muka sistem bertujuan untuk menggambarkan sistem yang akan dibuat. Menu utama dari aplikasi ini berisi menu *account*, data master, penilaian, F-AHP, dan laporan keputusan. Pada menu utama ini juga berisi informasi tentang tujuan dari pembuatan sistem dan bagaimana cara penggunaan sistem.

4.4.2.1 Rancangan Menu Login

Berikut ini adalah tampilan dari rancangan menu Log-In dari sistem deteksi kecurangan absensi (SIDETEK).

The screenshot shows a window titled "Login" with a blue border and a close button in the top right corner. The window content is as follows:

Absensi Karyawan

Login !
Isi username dan Password untuk masuk ke sistem

Username

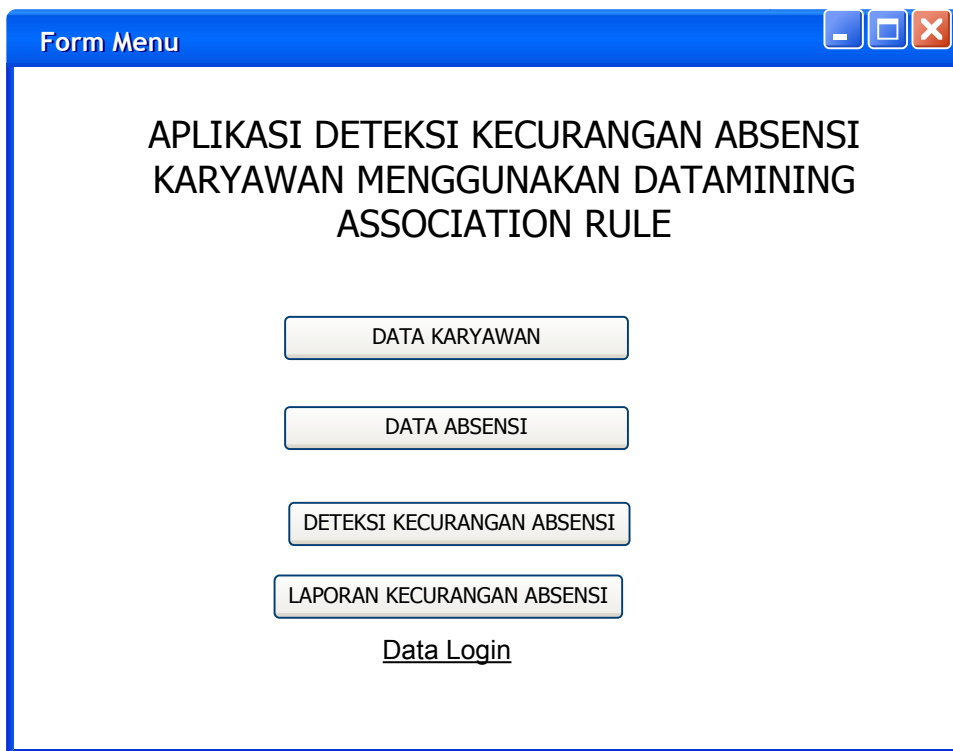
Password

Gambar 4.12 Rancangan Menu Login

4.4.2.2 Rancangan Menu Utama

Rancangan menu utama digunakan untuk menampilkan menu-menu yang ada dalam sistem deteksi kecurangan absensi (SIDETEK). Dalam menu utama ini terdapat tiga menu menu yaitu menu data karyawan, data absensi, menu deteksi kecurangan abseinsi karyawan dan menu laporan kecurangan absensi.

Tampilan berikut ini adalah rancangan tampilan menu utama



Gambar 4.13 RancanganMenu Utama

4.4.2.3 Rancangan Form Data Login

Rancangan dari form data login untuk data pengguna sistem terlihat pada *interface* berikut ini:

Gambar 4.14 Rancangan Form Data Login

4.4.2.4 Rancangan Form Karyawan

Rancangan dari form karyawan untuk data karyawan terlihat pada rancangan tampilan berikut ini:

Gambar 4.15 Rancangan Form Karyawan

4.4.2.5 Rancangan Form Absensi Karyawan

Rancangan dari form absensi untuk data absensi karyawan terlihat pada tampilan berikut ini:

The screenshot shows a window titled "Form Absensi Karyawan". It is divided into two main sections:

- Data Absensi:** This section contains four input fields: "ID Absen", "Nama Karyawan", "Tanggal Absen" (with a dropdown arrow), and "Jam". Below these fields are four buttons: "Simpan", "Ubah", "Hapus", and "Keluar".
- List Data Absensi Karyawan:** This section contains a table with 4 columns and 10 rows. Each cell in the top row contains the text "Enter Text".

Gambar 4.16 Rancangan Form Absensi

4.4.2.6 Rancangan Form Deteksi Kecurangan Absensi

Rancangan dari form deteksi kecurangan absensi untuk pencarian data kecurangan absensi karyawan dibagi menjadi 4 bagian tab, terlihat pada *interface* berikut ini:

1. Tab proses pengelompokan

The screenshot shows a window titled "Form processing Data Mining". It features several input fields and a table:

- Poses Frekuensi Itemset:** This section includes:
 - "Tanggal Absensi Karyawan" with a date range selector (Date - Date).
 - "Beta" with an "Enter Text" field.
 - "Minimum Support" with an "Enter Text" field.
 - "Support Count" with an "Enter Text" field.
 - "Proses" and "Keluar" buttons.
- Table:** A table with four tabs: "Pengelompokan", "Frekuensi item", "Transaksi data", and "Rule transaksi". The "Pengelompokan" tab is active, showing a table with two columns: "Tanggal" and "Itemset". The table has 7 rows, with the first row containing headers and the following rows being empty.

Gambar 4.17 Rancangan Form Deteksi Kecurangan Absensi (Tab1)

2. Tab proses frekuensi kemunculan karyawan

Form processing Data Mining

Poses Frekuensi Itemset

Tanggal Absensi Karyawan : -

Beta

Minimum Support

Support Count

Pengelompokan Frekuensi item Transaksi data Rule transaksi

Item	Frekuensi

Gambar 4.18 Rancangan Form Deteksi Kecurangan Absensi (Tab2)

3. Tab proses transaksi data

Form processing Data Mining

Poses Frekuensi Itemset

Tanggal Absensi Karyawan : -

Beta

Minimum Support

Support Count

Pengelompokan Frekuensi item Transaksi data Rule transaksi

TID	Itemset	Tanggal

Gambar 4.19 Rancangan Form Deteksi Kecurangan Absensi(Tab3)

4. Tab frekuensi kecurangan absensi

Form processing Data Mining

Poses Frekuensi Itemset

Tanggal Absensi Karyawan : -

Beta

Minimum Support

Support Count

Proses

Pengelompokan	Frekuensi item	Transaksi data	Rule transaksi																				
<table border="1"><thead><tr><th>Rule</th><th>Support count</th></tr></thead><tbody><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr></tbody></table>				Rule	Support count																		
Rule	Support count																						

Laporan

Gambar 4.20 Rancangan Form Deteksi Kecurangan Absensi(Tab4)

4.4.2.7 Rancangan Form Laporan Deteksi Kecurangan Absensi

Rancangan dari form laporan untuk melihat informasi prediksi kecurangan absensi karyawan dapat terlihat pada *interface* berikut ini:

Laporan kecurangan absensi karyawan

Enter Text

Nomor	Nama Karyawan	Tanggal

Gambar 4.21 Rancangan Form Laporan Kecurangan Absensi

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

Implementasi merupakan tahap pembuatan sistem berdasarkan dari analisa dan didesain sebelumnya hingga sistem dapat difungsikan dalam keadaan sebenarnya dan dapat dapat diketahui apakah sistem yang dibuat berhasil mencapai tujuan yang sebenarnya.

5.1.1 Batasan Implementasi

Batasan implementasi dalam pengembangan sistem ini antara lain:

1. Dikembangkan menggunakan *Microsoft Visual Basic 6.0*
2. *Database* menggunakan *my sql*.

Untuk mengimplementasikan aplikasi ini maka dibutuhkan perangkat pendukung, perangkat tersebut berupa perangkat keras dan perangkat lunak.

Minimum perangkat keras yang dibutuhkan :

1. *Processor* : Intel Pentium III 800 MHz
2. RAM : 128 MB
3. *Harddisk* : 10 GB

Perangkat lunak yang dibutuhkan :

1. *Platform* : *Windows*
2. Bahasa Pemrograman : *Ms. Visual Basic 6.0*
3. DBMS : *MySQL*
4. *Report Engine* : *Crystal Reports v10*

5.2 Implementasi Sistem

5.2.1 Menu *Login*

Form Login merupakan tampilan interface pertama kali yang akan muncul ketika menjalankan aplikasi ini. *User* harus menginputkan data *login* dengan mengisi *username* dan *password* yang tepat dan sesuai dengan data *login* yang sudah tersimpan di *database*, kemudian menekan tombol *Login*. Apabila data yang dimasukkan benar maka *user* akan dihadapkan ke menu utama dari Sistem Deteksi Kecurangan Asensi Karyawan.

Berikut ini adalah tampilan dari menu *Log-In* dari Sistem Deteksi Kecurangan Asensi Karyawan.



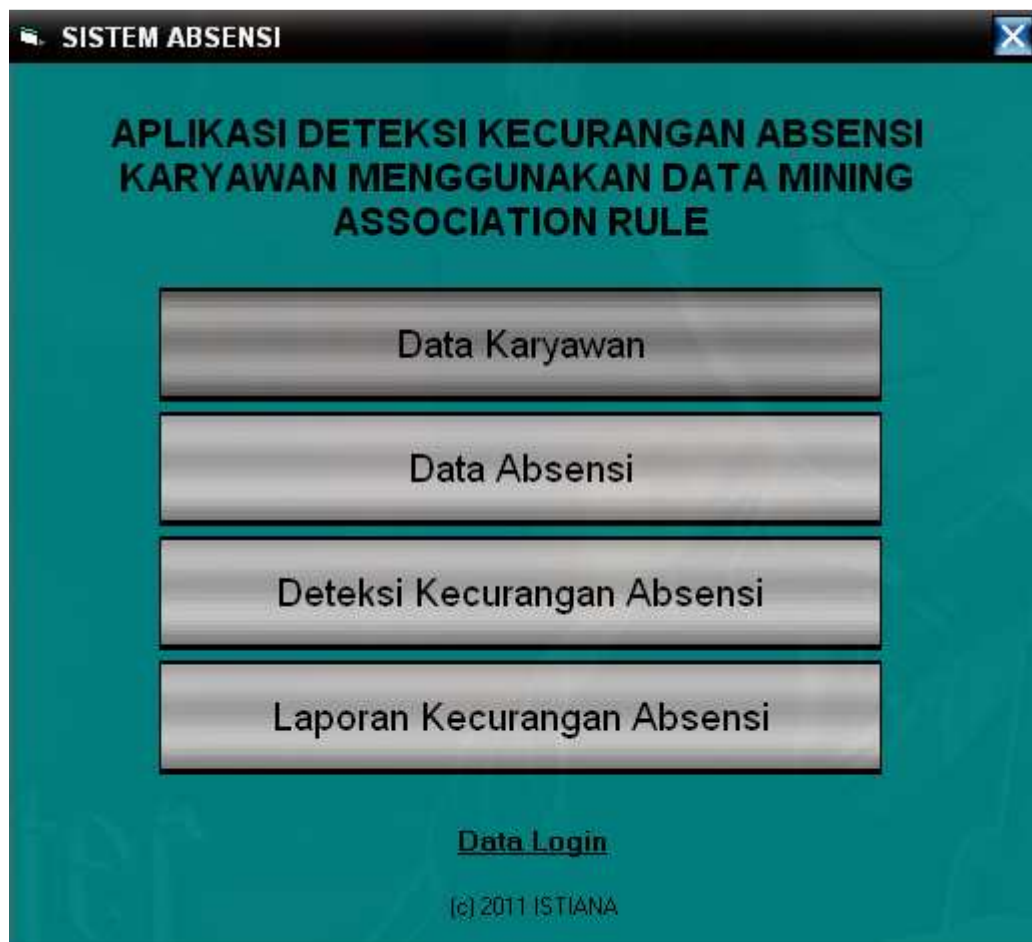
The image shows a screenshot of a web application's login page. The window title is "Login". The main heading is "Absensi Karyawan". Below the heading, the text "Login !" is displayed, followed by the instruction "Isi username dan password untuk masuk sistem." There are two input fields: "Username" with the value "admin" and "Password" with masked characters "xxxxxx". At the bottom, there are two buttons: "Login" and "Keluar".

Gambar 5.1 Menu Login

5.2.2 Menu Utama

Menu utama digunakan untuk menampilkan menu-menu yang ada dalam Sistem Deteksi Kecurangan Absensi Karyawan (SIDETEK). Dalam menu utama ini terdapat tiga menu yaitu menu karyawan, absensi dan menu deteksi kecurangan absensi.

Tampilan berikut ini adalah tampilan menu Utama



Gambar 5.2 Menu utama SIDETEK

5.2.2 Form Data Karyawan

Form karyawan ini berfungsi untuk menginputkan data karyawan. Fungsi lain yang terdapat pada *form* ini adalah menambah, menghapus, dan mengubah data karyawan



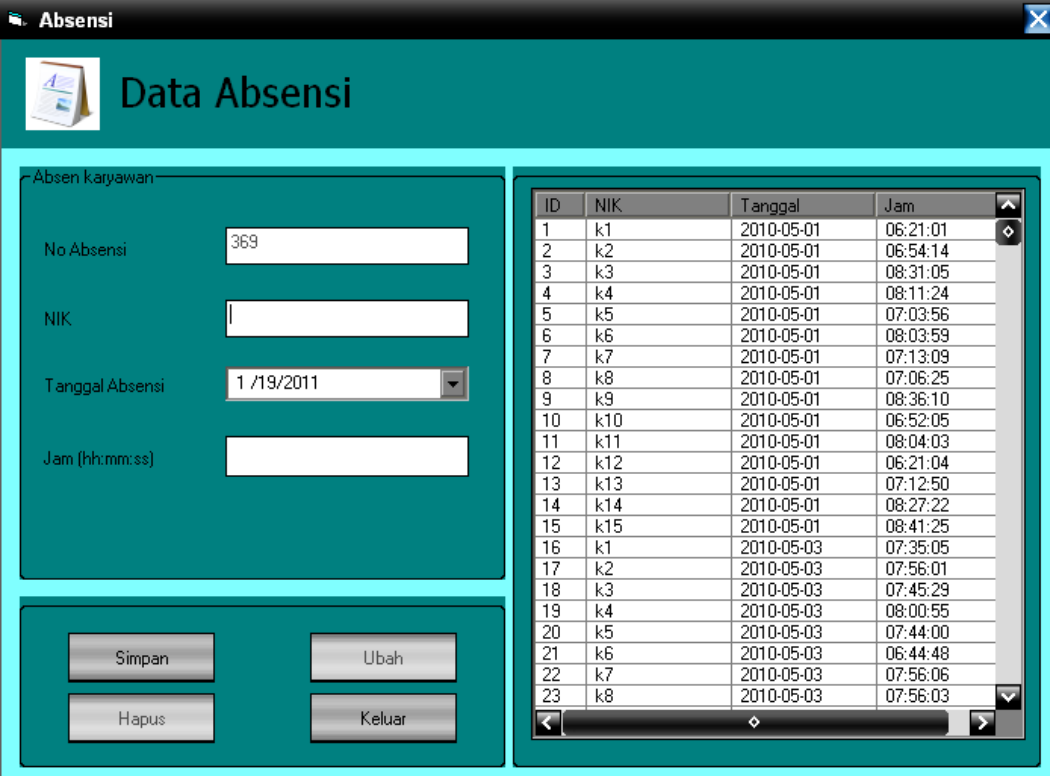
The screenshot shows a web application window titled "Karyawan" with a sub-header "Data Karyawan". On the left, there is a form with the following fields: NIK (text input), Nama Karyawan (text input), Tempat Lahir (text input), Tanggal Lahir (dropdown menu showing "11/21/2010"), and Alamat (text area). Below the form are four buttons: "Simpan", "Ubah", "Hapus", and "Keluar". On the right, there is a table with the following data:

NIK	Nama Karyawan	Tempat Lahir	Tanggal L.
k1	dodi	pekanbaru	2/13/1988
k2	rere	dumai	4/10/1988
k3	ratih	rengat	5/10/2000
k4	istiana	pekanbaru	8/18/1988
k5	iis afrianty	pekanbaru	4/28/2010
k6	nurwali	pelalawan	6/17/2010
k7	dwijayanty P	pekanbaru	11/28/198
k8	yuli handayani	bagan batu	7/20/1988
k9	anita febriani	bangkinang	2/4/1987
k10	sandra yunita	pekanbaru	5/7/1988
k11	yeni	dumai	5/5/1988
k12	sisi	pekanbaru	2/10/1988
k13	amal	pekanbaru	2/15/1988
k14	ara	pekanbaru	5/5/1997
k15	budi	pekanbaru	11/21/198
k100	qonita	pekanbaru	11/21/198
k16	tio	fsdg	11/12/198

Gambar 5.3 Form Karyawan

5.2.3 Form Data Absensi

Form absensi ini berfungsi untuk menginputkan data absensi karyawan. Fungsi lain yang terdapat pada *form* ini adalah menambah, menghapus, dan mengubah data absensi karyawan



ID	NIK	Tanggal	Jam
1	k1	2010-05-01	06:21:01
2	k2	2010-05-01	06:54:14
3	k3	2010-05-01	08:31:05
4	k4	2010-05-01	08:11:24
5	k5	2010-05-01	07:03:56
6	k6	2010-05-01	08:03:59
7	k7	2010-05-01	07:13:09
8	k8	2010-05-01	07:06:25
9	k9	2010-05-01	08:36:10
10	k10	2010-05-01	06:52:05
11	k11	2010-05-01	08:04:03
12	k12	2010-05-01	06:21:04
13	k13	2010-05-01	07:12:50
14	k14	2010-05-01	08:27:22
15	k15	2010-05-01	08:41:25
16	k1	2010-05-03	07:35:05
17	k2	2010-05-03	07:56:01
18	k3	2010-05-03	07:45:29
19	k4	2010-05-03	08:00:55
20	k5	2010-05-03	07:44:00
21	k6	2010-05-03	06:44:48
22	k7	2010-05-03	07:56:06
23	k8	2010-05-03	07:56:03

Gambar 5.4 Form Absensi

5.2.4 Form Deteksi Kecurangan Absensi

Form Deteksi kecurangan absensi ini berfungsi untuk mencari frekuensi karyawan yang memungkinkan melakukan kecurangan absensi, pada form ini terbagi menjadi empat tab yaitu, tab pertama yang berfungsi sebagai pengelompokan data, tab ke dua sebagai informasi frekuensi kemunculan sebuah item dari pengelompokan data, tab ketiga terdiri dari transaksi data yang telah mengalami proses pemindahan data, dan yang terakhir tab ke empat yang berfungsi sebagai hasil dari pencarian karyawan yang memungkinkan melakukan kecurangan absensi

1. Tab pertama

Pengelompokan data absensi

Processing

Deteksi Kecurangan Absensi Karyawan

Proses frekuensi itemset

Tanggal Absensi Karyawan: 5 / 1 /2010 - 5 /31/2010

Beta: 10 Detik

Minimum Support Karyawan: 2

Support Count Karyawan: 2

Proses Deteksi

Keluar

Lama Proses: 0 Detik

Data hasil pengelompokan

Tanggal	Itemset
01-05-2010	k1,k12
01-05-2010	k2,k100
01-05-2010	k6,k11
01-05-2010	k10,k16
03-05-2010	k2,k7
03-05-2010	k2,k8
03-05-2010	k2,k100
03-05-2010	k3,k16
03-05-2010	k4,k10
03-05-2010	k7,k8
03-05-2010	k7,k100
03-05-2010	k8,k100
04-05-2010	k1,k12
04-05-2010	k1,k14
04-05-2010	k2,k6
04-05-2010	k2,k100
04-05-2010	k6,k100
04-05-2010	k7,k16
04-05-2010	k12,k14
05-05-2010	k1,k12
05-05-2010	k1,k16

Gambar 5.5 Form Kecurangan Absensi (Tab1)

2. Tab kedua

Frekuensi kemunculan item dalam keseluruhan data yang telah berhasil dikelompokkan dari data absensi



Processing

Deteksi Kecurangan Absensi Karyawan

Proses frekuensi itemset

Tanggal Absensi Karyawan: 5 / 1 /2010 - 5 /31/2010

Beta: 10 Detik

Minimum Support Karyawan: 2

Support Count Karyawan: 2

Proses Deteksi

Keluar

Lama Proses : 0 Detik

Item	Frekuensi
k1	13
k2	10
k3	4
k4	3
k5	4
k6	10
k7	10
k8	5
k9	7
k10	6
k11	10
k12	12
k13	6
k14	6
k16	6
k15	2
k100	10

Gambar 5.6 Form Kecurangan Absensi (Tab2)

3. Tab Ketiga

Transaksi data yang telah mengalami pemindahan data

The screenshot shows a software window titled "Processing" with a sub-header "Deteksi Kecurangan Absensi Karyawan". Below the header, there is a section for "Proses frekuensi itemset" with the following fields and controls:

- Tanggal Absensi Karyawan: 5 / 1 / 2010 - 5 / 31 / 2010
- Beta: 10 Detik
- Minimum Support Karyawan: 2
- Support Count Karyawan: 2
- Buttons: "Proses Deteksi" and "Keluar"
- Status: "Lama Proses: 0 Detik"

Below the input fields is a table titled "Data Transaksi Berdasarkan nilai minimum support karyawan". The table has four columns: TID, Itemset, Tanggal, and an empty column. The data rows are as follows:

TID	Itemset	Tanggal	
1	k1.k12	01-05-2010	
2	k2.k100	01-05-2010	
3	k6.k11	01-05-2010	
4	k10.k16	01-05-2010	
5	k2.k7	03-05-2010	
6	k2.k8	03-05-2010	
7	k2.k100	03-05-2010	
8	k16.k3	03-05-2010	
9	k10.k4	03-05-2010	
10	k7.k8	03-05-2010	
11	k7.k100	03-05-2010	
12	k100.k8	03-05-2010	
13	k1.k12	04-05-2010	
14	k1.k14	04-05-2010	
15	k2.k6	04-05-2010	
16	k2.k100	04-05-2010	
17	k6.k100	04-05-2010	
18	k7.k16	04-05-2010	
19	k12.k14	04-05-2010	
20	k1.k12	05-05-2010	
21	k1.k14	05-05-2010	

Gambar 5.7 Form Kecurangan Absensi (Tab3)

Tanggal Absensi Karyawan :

5 / 1 /2010 - 5 /31/2010

Lihat Laporan Keluar

Gambar 5.8 Form Tampil laporan

Preview

LAPORAN KECURANGAN ABSENSI KARYAWAN

No	Nama Karyawan	Tanggal
1	dodi (k1) - sisi (k12)	1-M ay-10 4-M ay-10 5-M ay-10 6-M ay-10 10-M ay-10 12-M ay-10 14-M ay-10 29-M ay-10
2	rere (k2) - qonita (k100)	1-M ay-10 3-M ay-10 4-M ay-10
3	rere (k2) - nurwati (k6)	4-M ay-10 11-M ay-10 14-M ay-10
4	rere (k2) - yuli handayani (k8)	3-M ay-10 27-M ay-10
5	nurwati (k6) - qonita (k100)	4-M ay-10 5-M ay-10

Gambar 6.9 Form Laporan Kecurangan Absensi

5.2.6 Form Data Login

Form Data Login ini berfungsi untuk mengatur pengguna yang berhak menggunakan sistem ini. *Form* ini memiliki fungsi untuk menambah *username* dan *password*. Form data login dapat dilihat pada Form berikut ini:



Username	
admin	

Gambar 5.9 Form Data Login

5.3 Pengujian

Setelah tahap implementasi selesai dilakukan, maka dilanjutkan dengan pengujian sistem dari implementasi yang telah dibuat. Pengujian ini dilakukan dengan dua cara yaitu :

1. Pengujian Sistem
2. Pengujian Parameter data

5.3.1 Pengujian Sistem

Pengujian dilakukan untuk melihat hasil implementasi, apakah berjalan sesuai tujuan atau masih terdapat kesalahan-kesalahan. Pengujian sistem Deteksi Kecurangan Absensi dilakukan pada lingkungan pengujian sesuai dengan lingkungan implementasi. Pengujian dilakukan dengan menguji fungsi-fungsi perintah yang ada disetiap form-form pada sistem informasi deteksi kecurangan absensi tersebut. Hasil pengujian sistem dapat dilihat pada tabel-tabel pengujian dibawah ini.

5.3.1.1 Pengujian Menu *Login*

Tabel 5.1 Tabel Pengujian Menu *Login*

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran Yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian menu login	Tampilan layar login aplikasi	1.Masukan username dan password 2.Klik tombol Login untuk masuk ke menu utama 3.Tampil menu utama	Data username dan password benar	Data berhasil di validasi tidak ada instruksi error	Layar yang ditampilkan sesuai dengan yang diharapkan	Data berhasil di validasi tidak ada instruksi error	Diterima
			Data username kosong	Muncul pesan "Username belum diisi"		Muncul pesan "Username belum diisi"	Diterima
			Data password kosong	Muncul pesan "password belum diisi"		Muncul pesan "Username belum diisi"	Diterima
			Data username dan password salah	Muncul pesan "Data tidak ditemukan"		Muncul pesan "Data tidak ditemukan"	Diterima

5.3.1.2 Pengujian Menu Utama

Tabel 5.2 Tabel Pengujian Menu Utama

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran Yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian menu utama	Tampilan layar menu utama	1. Klik menu yang diinginkan	Klik “Data Karyawan ”	Muncul Form “Karyawan”	Layar yang ditampilkan sesuai dengan yang diharapkan	Muncul Form “Karyawan ”	Diterima
			Klik “Data Absensi”	Muncul Form “Absensi”		Muncul Form “Absensi”	Diterima
			Klik “Deteksi kecurangan absensi”	Muncul Form “Deteksi kecurangan absensi “		Muncul Form “ Deteksi kecurangan absensi “	Diterima
			Klik “laporan kecurangan absensi”	Muncul Form “Laporan kecurangan absensi “		Muncul Form “Laporan kecurangan absensi “	Diterima

5.3.1.3 Pengujian Menu Data Karyawan

Tabel 5.3 Tabel Pengujian Data karyawan

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran Yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian Form Data karyawan	Tampilan layar utama	1.Pilih menu pada tampilan utama,klik data karyawan 2.inputkan data karyawan untuk menambah data karyawan, inputkan NIK, nama, tempat lahir, tanggal lahir, dan alamat 3.Klik tombol	Data NIK, nama, tempat lahir, tanggal lahir, alamat benar	Data berhasil di validasi tidak ada instruksi error	Layar yang ditampilkan sesuai dengan yang diharapkan	Data berhasil di validasi tidak ada instruksi error	Diterima

		simpan untuk menyim pan 4.Klik tombol hapus untuk meng hapus 5.dan klik tombol ubah untuk mengubah					
--	--	--	--	--	--	--	--

5.3.1.4 Pengujian Menu Data Absensi

Tabel 5.4 Tabel Pengujian Data Absensi

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran Yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian Form Data absensi	Tampilan layar utama	<p>1.Pilih menu pada tampilan utama,klik data absensi</p> <p>2.inputkan data absensi untuk menambah data absensi harian karyawan inputkan id absen, NIK, tanggal, dan waktu absensi</p> <p>3.Klik tombol</p>	Data id absen, NIK, tanggal, dan waktu absensi	Data berhasil di validasi tidak ada instruksi error	Layar yang ditampilkan sesuai dengan yang diharapkan	Data berhasil di validasi tidak ada instruksi error	Diterima

		simpan untuk menyim pan 4.Klik tombol hapus untuk meng hapus 5.dan klik tombol ubah untuk mengubah					
--	--	--	--	--	--	--	--

5.3.1.5 Pengujian Menu Deteksi Kecurangan Absensi

Tabel 5.5 Tabel Pengujian Deteksi Kecurangan Absensi

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran Yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian Form Data absensi	Tampilan layar utama	1.Pilih menu pada tampilan utama,klik ” deteksi kecurangan absensi” 2.inputkan inputkan tanggal, nilai beta, minimum support, dan support count 3.Klik tombol proses untuk melakukan pencarian	Data tanggal, nilai beta, minimum support, dan support count benar	Data berhasil di validasi tidak ada instruksi error	Layar yang ditampilkan sesuai dengan yang diharapkan	Data berhasil di validasi tidak ada instruksi error	Diterima

5.3.1.6 Pengujian Form Data login

Tabel 5.6 Tabel Pengujian Data login

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran Yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
Pengujian Form data login	Tampilan layar utama	1.pilih data login 2. input username dan password jika menambah pengguna 3.klik list pengguna, pilih pengguna,ubah data pengguna, klik ubah jika ingin mengubah data pengguna 4. klik list pengguna, pilih pengguna, klik hapus jika ingin menghapus data pengguna	Data username, password ,dan jenis benar	Data berhasil di validasi tidak ada instruksi error	Layar yang ditampilkan sesuai dengan yang diharapkan	Data berhasil di validasi tidak ada instruksi error	Diterima
			Data username kosong	Muncul pesan “Data belum lengkap”		Muncul pesan “Data belum lengkap”	Diterima
			Data pasword kosong	Muncul pesan “Data belum lengkap”		Muncul pesan “Data belum lengkap”	Diterima
			Data username telah ada	Muncul pesan “username telah ada”		Muncul pesan “username telah ada”	Diterima

5.3.2 Pengujian Peubah Parameter data

Pengujian dilakukan untuk melihat hasil dari parameter data yang diubah-ubah dan kecepatan akses data dari sistem. Pengujian ini dilakukan pada lingkungan pengujian sesuai dengan lingkungan implementasi. Pengujian dilakukan dengan menguji nilai parameter dalam proses pencarian data yaitu dengan mengubah-ubah inputan dari proses yang akan dijalankan. Hasil pengujian sistem dapat dilihat pada tabel-tabel pengujian dibawah ini.

Tabel 5.7 Hasil pengujian Peubah Parameter dari perangkat lunak

pengujian	Data Simulasi (1 Bulan Absensi)			
	Proses	Inputan	Lama Proses	Total Transaksi & Rule
Pengujian 1	Binning	Beta = 10 detik	1 detik	46
	Frekuensi item	Min. Support = 2	1 detik	46
	<i>Association rule</i>	Min.Support Count = 2	1 detik	17
Pengujian 2	Proses	Inputan	Lama Proses	Total Transaksi & Rule
	Binning	Beta = 10 detik	1 detik	46
	Frekuensi item	Min. Support = 2	1 detik	46
	<i>Association rule</i>	Min.Support Count = 3	1 detik	11
Pengujian 3	Proses	Inputan	Lama Proses	Total Transaksi & Rule
	Binning	Beta = 30 detik	1 detik	46
	Frekuensi item	Min. Support = 2	1 detik	46
	<i>Association rule</i>	Min.Support Count = 4	1 detik	8
Pengujian 4	Proses	Inputan	Lama Proses	Total Transaksi & Rule
	Binning	Beta = 10 detik	1 detik	46
	Frekuensi item	Min. Support = 2	1 detik	46
	<i>Association rule</i>	Min.Support Count = 5	1 detik	8
Pengujian 5	Proses	Inputan	Lama Proses	Total Transaksi & Rule
	Binning	Beta = 10 detik	1 detik	46
	Frekuensi item	Min. Support = 2	1 detik	46
	<i>Association rule</i>	Min.Support Count = 6	1 detik	8
Pengujian 6	Proses	Inputan	Lama Proses	Total Transaksi & Rule
	Binning	Beta = 10 detik	1 detik	46
	Frekuensi item	Min. Support = 2	1 detik	46
	<i>Association rule</i>	Min.Support Count = 7	1 detik	8
Pengujian 7	Proses	Inputan	Lama Proses	Total Transaksi & Rule
	Binning	Beta = 10 detik	1 detik	46
	Frekuensi item	Min. Support = 2	1 detik	46
	<i>Association rule</i>	Min.Support Count = 8	1 detik	8
Pengujian 8	Proses	Inputan	Lama Proses	Total Transaksi & Rule
	Binning	Beta = 10 detik	1 detik	46
	Frekuensi item	Min. Support = 2	1 detik	46
	<i>Association rule</i>	Min.Support Count = 9	1 detik	0

Hasil pengujian lanjutan pada lampiran D.

5.3.3 Kesimpulan Pengujian

Setelah melakukan pengujian pada sistem, yaitu berdasarkan pengujian aplikasi sistem dapat berjalan dengan lancar tanpa kendala apapun dan berdasarkan pengujian parameter data diketahui semakin tinggi batas frekuensi yang ditentukan maka semakin sedikit transaksi dan *rule* yang dihasilkan. Hal ini dikarenakan frekuensi berpengaruh saat menentukan seberapa sering kemunculan item atau itemset (karyawan) saat proses dijalankan. Dari hasil pengujian sistem ditemukan pola bahwa karyawan dengan inisial k1 dan k12 lebih sering melakukan absensi berdekatan dibandingkan dengan karyawan lain sehingga memungkinkan karyawan dengan inisial k1 dengan k12 melakukan absensi secara bersamaan dan memungkinkan melakukan penitipan absensi. Akan tetapi *Rule* yang dihasilkan dalam pencarian itemset sangat tergantung dari batasan nilai frekuensi yang di inputkan. Sistem tidak dapat menentukan frekuensi yang tepat untuk menyatakan batasan frekuensi yang ideal. Hal ini menyebabkan jika batasan frekuensi ditingkatkan memungkinkan itemset yang tidak memenuhi batas nilai tidak terdeteksi ataupun sebaliknya. Secara keseluruhan dapat diambil kesimpulan sistem dapat di implementasikan menggunakan struktur penyimpanan data *FP-tree* dan penerapan algoritma *FP-Growth*.

BAB VI

P E N U T U P

6.1. Kesimpulan

Setelah melalui tahap pengujian pada sistem deteksi kecurangan absensi, dapat diambil kesimpulan bahwa :

1. Dari hasil pengujian sistem dapat disimpulkan struktur penyimpanan data *FP-tree* dan penerapan algoritma *FP-Growth* dapat dimanfaatkan untuk menggali kebiasaan absensi karyawan yaitu dengan melihat tingakat frekuensi kemunculan dua orang karyawan melakukan absensi secara berdekatan.
2. Struktur penyimpanan data *FP-Tree* dan Algoritma *FP_Growth* terbukti dapat mengoptimalkan pencarian *frequent itemset* .
3. *Rule* yang dihasilkan dalam pencarian itemset sangat tergantung dari batasan nilai frekuensi. Sitem tidak dapat menentukan frekuensi yang tepat untuk menyatakan batasan frekuensi yang ideal. Hal ini menyebabkan jika batasan frekuensi ditingkatkan memungkinkan itemset yang tidak memenuhi batas nilai tidak terdeteksi ataupun sebaliknya.

6.2. Saran

Saran yang dapat diberikan penulis untuk pengembangan selanjutnya yaitu:

1. Perbandingan untuk pengelompokan data (*Binning*) dilakukan lebih dari dua orang karyawan yang melakukan absensi secara berdekatan.
2. Melakukan analisa batasan frekuensi dalam menentukan pola kehadiran karyawan secara bersamaan.

**RANCANG BANGUN APLIKASI
DETEKSI KECURANGAN ABSENSI KARYAWAN
MENGUNAKAN *DATA MINING ASSOCIATION RULE***

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

oleh :

ISTIANA
NIM : 10651004301



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2011**

DAFTAR PUSTAKA

Fayyad, U. Piatetsky-Shapiro, dan G. Smyth. *From Data Mining to Knowledge Discovery in Databases*. 1996. Jurnal. AAAI and The MIT Pres, 37-53. Diakses 20 Februari 2011.

Gunawan, *Knowledge Discovery in Database(IS704) dan Data Mining(CS704)* http://.paper.no.7.UTY.ac.id/files/2006/tex_mining_market_basket.pdf diakses 16 juni 2010.

Han, Jiawei, Micheline Kamber, *Data Mining : Concepts and Techniques*. Morgan Kaufmann, 2001.

Larose, Daniel T. *Discovering Knowledge in Data: An Introduction to Data mining*, John Willey & Sons. Inc, 2005. Diakses 11 juli 2010

M. Fairuzabadi , Data-Mining-Bab-01 [online] Available <http://fairuz.upy.ac.id/wp-content/uploads/2009/09/Data-Minining-Bab-01.pdf> , diakses tanggal 16 Juni 2010).

Prihastomo. “ *Algoritma Divide and Conquer* ” [online] available <http://prihastomo.files.wordpress.com/2008/01/divideconquer.pdf> diakses tanggal 26 Agustus 2010

Samuel, David. “*Penerapan Struktur Fp-Tree Dan Algoritma Fp-Growth Dalam Optimasi Penentuan Frekuensi Itemset*”, www.itb.ac.id/penelitian/makalah/IF2153-0708.pdf diakses tanggal 1 juli 2010.

Santosa, Budi , *DATAMINING : Teknik pemanfaatan Data untuk Keperluan bisnis*, Graha Ilmu, Yogyakarta, 2007.

Setia Budhi, Gregorius , *Jurnal Market Basket Untuk Deteksi Kecurangan Absensi*
<http://fportofolio.petra.ac.id/user-files/02-030/jurnal%20Market%20Basket%20untuk%20deteksi%20kecurangan%20absensi.pdf> , diakses tanggal 6 Juni 2010).

Soelaiman, Ruly. Arini, Ni Made. “*Analisis Kinerja Algoritma Fold-Growth Dan Fp-Growth Pada Penggalian Pola Asosiasi*”,
www.si.ac.id/Penelitian/JURNAL/Arif.pdf Tanggal akses 1 juli 2010

Sucahyo, Yudho Giri. *Data Mining : Menggali informasi yang terpendam.* (2006),
<http://ilmukomputer.com/populer/yudho-datamining.php>, Diakses tanggal 6 Juni 2010

Tessy, Data Mining Bab 10 [online] available www.lecturer.eepis-its.edu/~tessy/lecturenotes/db2/bab10.pdf, diakses tanggal 1 juli 2010

Turban, E., dkk, *Decicion Support Systems and Intelligent Systems*, Andi Offset, 2005. Diakses 11 juli 2010

www.thearling.com, 2002. Diakses 6 Juni 2010

DAFTAR LAMPIRAN

Lampiran	Halaman
A Pembangunan <i>FP-Tree</i> (Lanjutan).....	A-1
B Hasil Pencarian Menggunakan Metode <i>FP-Growth</i>	B-1
C PEngujian Peubah Parameter (Lanjutan)	C-1
D Daftar Istilah	D-1
E. Daftar Simbol	E-1

DAFTAR TABEL

Tabel		Halaman
4.1	Absensi.....	IV-3
4.2	Penggalan tabel absensi	IV-4
4.3	Pengelompokan absensi	IV-9
4.4	Data hasil pengelompokan	IV-10
4.5	Data transaksi mentah dengan nilai beta 10 detik.....	IV-11
4.6	Frekuensi kemunculan tiap karakter item	IV-12
4.7	Data transaksi <i>Min.Support</i>	IV-13
4.8	Hasil Pembangunan <i>FP-Tree</i>	IV-16
4.9	Itemset yang memenuhi <i>minimum support count</i>	IV-29
4.10	Nilai <i>support count</i> masing-masing itemset	IV-31
4.11	Proses DFD Level 1 SIDETEK	IV-36
4.12	Aliran Data DFD Level 1 SIDETEK	IV-36
4.13	Proses DFD Level 2 Proses 1 Login	IV-37
4.14	Aliran DFD Level 2 Proses 1 Login	IV-38
4.15	Proses DFD Level 2 Proses 2 Karyawan	IV-38
4.16	Aliran DFD Level 2 Proses 2 Karyawan	IV-39
4.17	Proses DFD Level 2 Proses 3 Absensi.....	IV-39
4.18	Aliran DFD level 2 Proses 3 Absensi	IV-40
4.19	Proses DFD Level 2 Proses 4 Deteksi Kecurangan Absensi	IV-41
4.20	Aliran DFD Level 2 Proses 4 Deteksi Kecurangan Absensi	IV-41
4.21	Keterangan Entitas pada ERD.....	IV-42
4.22	Basis Data Login.....	IV-43
4.23	Basis Data Karyawan	IV-44
4.24	Basis Data Absensi	IV-44
4.25	Basis DataKelompok	IV-45
4.26	Basis Data frekuensi item	IV-45
4.27	Basis Data <i>FP-Tree</i>	IV-46

4.28	Basis Data <i>FP-Growth</i>	IV-46
5.1	Pengujian Menu <i>login</i>	V-12
5.2	Pengujian Menu Utama.....	V-13
5.3	Pengujian data karyawan	V-14
5.4	Pengujian data absensi	V-16
5.5	Pengujian deteksi kecurangan absensi	V-18
5.6	Pengujian data login.....	V-19
5.7	Hasil pengujian proses kecepatan data.....	V-20

DAFTAR GAMBAR

Gambar	Halaman
2.1 Tahap <i>knowledge discovery in database</i> (KDD)	II-1
2.2 Algoritma <i>FP-Growth</i>	II-16
3.1 <i>Flowchart</i> Metodologi Penelitian	III-2
4.1 Lintasan yang mengandung itemset yang bersamaan.....	IV-30
4.2 Lintasan itemset yang mengandung <i>support count</i> yang sama.....	IV-31
4.3 <i>Flowchart</i> Sistem	IV-33
4.4 <i>Contact Diagram</i>	IV-33
4.5 DFD Level 1	IV-34
4.6 DFD Level 2 Proses Data Login.....	IV-37
4.7 DFD Level 2 Proses Pengelolaan Data karyawan	IV-38
4.8 DFD Level 2 Proses absensi	IV-39
4.9 DFD Level 2 Proses kecurangan absensi.....	IV-40
4.10 ERD SIDETEK.....	IV-42
4.11 Struktur Menu deteksi kecuran absensi	IV-47
4.12 Rancangan form login.....	IV-47
4.13 Rancangan form utama	IV-48
4.14 Rancangan form Login	IV-49
4.15 Rancangan form karyawan.....	IV-49
4.16 Rancangan form absensi	IV-50
4.17 Rancangan form deteksi kecurangan absensi(Tab1).....	IV-50
4.18 Rancangan form deteksi kecurangan absensi(Tab2).....	IV-51
4.19 Rancangan form deteksi kecurangan absensi(Tab3).....	IV-51
4.20 Rancangan form deteksi kecurangan absensi(Tab4).....	IV-52
4.21 Rancangan form Laporan.....	IV-52
5.1 Menu Login.....	V-2
5.2 Menu Utama.....	V-3
5.3 Form Karyawan	V-4

5.4	Form Absensi	V-5
5.5	Form Deteksi kecurangan absensi(Tab1).....	V-6
5.6	Form Deteksi kecurangan absensi(Tab2).....	V-7
5.7	Form Deteksi kecurangan absensi(Tab3).....	V-8
5.8	Form Deteksi kecurangan absensi(Tap4).....	V-9
5.9	Form Tampil Laporan	V-10
5.10	Form Laporan Kecurangan Absensi	V-10
5.11	Form Data Login.....	V-11

DAFTAR RIWAYAT HIDUP



Penulis dilahirkan di Kota Pekanbaru Riau pada tanggal 18 Agustus 1987 dari Ayahanda St.Zainuddin dan Ibunda Nurhayati. Penulis anak Terakhir dari enam bersaudara. Pada tahun 1994, penulis masuk Sekolah Dasar Negeri 015 Kec. Bukit Raya Kel. Tengkerang Barat Pekanbaru dan menyelesaikannya pada tahun 2000. Kemudian melanjutkan di tingkat Sekolah Lanjutan Tingkat Pertama DMP (Diniyyah Menengah Pertama) Perguruan Diniyyah Putri Padang Panjang Sumatera Barat dan menamatkannya pada tahun 2003. Tahun 2006 berhasil menamatkan Sekolah Menengah Umum di MA Negeri 2 Model Pekanbaru.

Setelah menamatkan pendidikan formal di tahun 2006, penulis masuk ke Universitas Islam Negeri Sultan Syarif Kasim, sebagai mahasiswa Jurusan Teknik Informatika hingga menyelesaikan Laporan Tugas Akhir.

Selama menjadi mahasiswa, penulis juga pernah melaksanakan Kerja Praktek di Koran Haria Riau Mandiri yang saat ini bernama Haluan Riau pada tahun 2010.