BAB III
ANALISIS DAN PERANCANGAN SISTEM

III.1. Analisis Sistem

Analisis sistem adalah pengurain dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan. Tahap analisis sistem merupakan tahapan yang sangat kritis dan penting, karena kesalahan di dalam tahap ini akan menyebabkan juga kesalahan di tahap selanjutnya. Tugas utama analisis sistem dalam tahap ini adalah menemukan kelemahan-kelemahan dari sistem yang berjalan sehingga dapat diusulkan perbaikannya. Dalam analisis sistem ini meliputi beberapa bagian, yaitu :

a. Analisis Masalah
b. Analisis Penerapan Data Mining
c. Analisis Spesifikasi Kebutuhan Perangkat Lunak
d. Analisis Kebutuhan *Non-Fungsional*
e. Analisis Kebutuhan *Fungsional*

III.1.1. Analisis Masalah

Adapun analisis masalah di PT. Bandung Perkasa Jaya adalah sebagai berikut :

a. Banyaknya jenis hasil benang yang diproduksi oleh PT. Bandung Perkasa Jaya membuat pihak bagian produksi sulit untuk menemukan jenis benang apa saja yang harus disediakan.
b. Penentuan jenis benang yang akan diproduksi berdasarkan *history* penjualan benang yang sering dijual sering terjadi kesalahan sehingga mengurangi pendapatan.

Untuk meminimalisir atau bahkan mengatasi permasalahan di atas, pihak terkait di perusahaan harus dapat mengambil keputusan untuk menentukan

jenis benang yang akan diproduksi. Untuk melakukan hal tersebut, perusahaan memerlukan informasi yang cukup untuk dapat dianalisis lebih lanjut.

III.1.2. Analisis Penerapan Data Mining

Adapun tahapan-tahapan dalam penerapan data mining yang digunakan dalam penelitian ini adalah sebagai berikut :

III.1.2.1. Pemahaman Bisnis

Tahapan ini merupakan tahapan awal pada metode *Data Mining*. Tahap ini bertujuan untuk memahami tujuan dan kebutuhan dari sudut pandang bisnis PT. Bandung Perkasa Jaya. Dalam tahapan ini ada beberapa tahapan lainnya, yaitu:

Tujuan Bisnis

Memasarkan berbagai produk jenis benang yang dijual kepada konsumen merupakan proses bisnis dari PT. Bandung Perkasa Jaya. Tujuannya adalah untuk memenuhi permintaan konsumen sehingga dapat meningkatkan pendapatan.

Penentuan Sasaran *Data Mining*

Tujuan dari penerapan *data mining* pada penjualan produk jenis benang ini adalah untuk mengetahui bagaimana pola pembelian dari pelanggan dan jenis produk benang apa saja yang sering dipesan secara bersamaan oleh pelanggan.

III.1.2.2. Pemahaman Data

Berdasarkan tujuan bisnis pada PT. Bandung Perkasa Jaya data yang digunakan dalam penelitian ini adalah laporan transaksi penjualan produk jenis benang pada bulan Januari 2014 sampai dengan Desember 2014 yang berasal dari *file microsoft excel* (.xlsx). Untuk lebih jelasnya dapat dilihat di Tabel D-1 pada lampiran D. Atribut yang terdapat pada laporan transaksi penjualan terdiri dari:

1. No. Nota
2. Kode Barang
3. Nama Barang
4. Jumlah
5. Tanggal

Berikut merupakan penjelasan dari masing-masing atribut yang terdapat pada tabel laporan transaksi penjualan yang dapat dilihat pada Tabel III.1.

Tabel III.1. Penjelasan Atribut

No.	Atribut	Tipe Data	Definisi
1	No. Nota	Varchar	No transaksi pada nota penjualan
2	Kode Barang	Varchar	Kode dari barang pada transaksi penjualan
3	Nama Barang	Varchar	Nama dari barang pada transaksi penjualan
4	Jumlah	Int	Jumlah produk yang dibeli oleh konsumen
5	Tanggal	Date	Tanggal transaksi penjualan

III.1.2.3. Persiapan Data

Tahapan persiapan data harus dilakukan dalam proses *data mining* karena tidak semua *record* atau *attribute* digunakan dalam proses *data mining.* Proses ini dilakukan agar data yang akan digunakan sesuai dengan kebutuhan. Adapun tahapan-tahapan persiapan data dalam penelitian ini adalah sebagai berikut:

1. Pemilihan Atribut

   Berdasarkan informasi yang ingin didapat oleh pengguna mengenai jenis produk apa saja yang dibeli secara bersamaan oleh pelanggan, maka dalam tahap ini atribut yang akan digunakan dari tabel D-1 dalam lampiran D hanyalah NoNota dan KodeBarang. Atribut NoNota merupakan identitas dari transaksi dan atribut KodeBarang merupakan kode dari jenis barang yang dibeli pelanggan. Hasil pemilihan atribut dapat dilihat pada tabel D-2 dalam lampiran D.

2. Pembersihan Data

   Pada tahap ini, hasil pemilihan atribut pada tabel D-2 dalam lampiran D akan dibersihkan dari data transaksi yang hanya memiliki satu produk dalam satu nota. Hasil pembersihan *record-record* tersebut dapat dilihat pada tabel D-3 dalam lampiran D.

3. Penyiapan Data Awal

   Setelah semua proses telah berhasil dilakukan dan data transaksi telah sesuai dengan kebutuhan yang diperlukan dalam proses *data mining*, maka data transaksi pada tabel D-3 dalam lampiran D disederhanakan agar mempermudah dalam pembacaan. Untuk lebih jelasnya dapat dilihat pada tabel D-4 dalam lampiran D.

III.1.2.4. Pemodelan

Pemodelan ini dilakukan dengan menggunakan metode *Association Rule* dengan algoritma *FP-Growth* terhadap data awal pada tabel D-4 dalam lampiran D untuk mendapatkan *rule* yang dihasilkan. Adapun tahapan-tahapan dalam metode *Association Rule* dengan algortima *FP-Growth* adalah sebagai berikut :

Pembangunan *FP-Tree*

Hasil data awal pada tabel D-4 pada lampiran D akan diolah untuk mendapatkan *rule* dengan mengasumsikan *minimum support* 10. Karena dengan menggunakan nilai *minimum support* ini akan memperoleh hasil yang maksimal sesuai dengan tujuan yang telah ditentukan sebelumnya. Maka bagi item yang memiliki frekuensi ≤ 10 akan dihapus.

a. Penentuan Frequent Itemset

Data awal pada tabel D-4 dalam lampiran D akan dihitung nilai frekuensui kemunculan tiap item dengan hasil sebagai berikut :

Tabel III.2. Frekuensi Kemunculan Item

Item	Frequent
CA16	6
CA20	19
CA24	17
CA28	3
CA30	19
CA40	30
CO20	26
CO24	24
CO30	24

Setelah mendapatkan frekuensi dari tiap item, selanjutnya membandingkan frekuensi dari tiap item dengan nilai *minimum support* dan menghilangkan item yang memiliki frekuensi lebih kecil dari *minimum support*. Pada tabel III.2 ada 2 item yang memiliki frekuensi kurang dari *minimum support* yaitu CA16, CA28. Maka item dan transaksi yang mengandung item tersebut pun dihilangkan . Dan hasilnya sebagai berikut :

Tabel III.3. Item yang memenuhi *minimum support*

Item	Frequent
CA20	19
CA24	17
CA30	19
CA40	30
CO20	26
CO24	24
CO30	24

Setelah didapatkan data yang memenuhi *minimum support*, selanjutnya mencari *frequency of occurance*. Tabel *frequency of occurance* ini didapatkan dengan cara mengurutkan item dari tabel III.3 yang memiliki frekuensi terbesar hingga terkecil, lalu ditentukan *priority*-nya sesuai dengan urutan frekuensi tersebut.

Tabel III.4. *Frequency* of Occurance

Item	Frequent	*Priority*
CA40	30	1
CO20	26	2
CO24	24	3
CO30	24	4
CA20	19	5
CA30	19	6
CA24	17	7

Setelah menentukan nilai *priority*, selanjutnya data yang berada pada tabel III.3 diurutkan sesuai *priority* dari masing-masing item. Berikut adalah hasil pengurutan data berdasarkan *priority :*

Tabel III.5. Hasil Pengurutan Data Transaksi

NoNota	KodeBarang
1	CA20, CA30
2	CA30, CO24
3	CA24, CA40, CA20
5	CA30, CA40
6	CA30, CA24
7	CO30, CA24
8	CA30, CA40, CA20
11	CA40, CA30
12	CA24, CA20
18	CA30, CA40, CA20
19	CA20, CA30, CA40, CO24
20	CO24, CA30
23	CO30, CA40, CO24, CO20

NoNota	KodeBarang
24	CA20, CO24
25	CA24, CA30, CA40
28	CO20, CO30
33	CO20, CA40
34	CO20, CO30, CO24, CA40
35	CA24, CA20
46	CA40, CA20, CO30
47	CA40, CO20, CA24, CA30
49	CA40, CA20, CA30
50	CA40, CO20, CA24
51	CA40, CO24, CA30
52	CO30, CA20
53	CO20, CA24
55	CO20, CO24, CA20
61	CO30, CA20
62	CO24, CA30, CA40, CO20
63	CA24, CO30
64	CO30, CA40, CA20, CO24
65	CO24, CA30,CA40, CO20
67	CO24, CO20
68	CO30, CA40, CO24, CO20
69	CO20, CO24, CO30
71	CO30, CO24
75	CA24, CO20,CO30, CA40
77	CO24, CA20, CO30
78	CA30, CA40
79	CO30, CA24
80	CO30, CO24, CA40, CO20
83	CO20, CO30, CA30
84	CO20, CA40
86	CO20, CO24, CA40, CA24
87	CA40, CO20, CO24
91	CO30, CO20, CA24, CA40, CO24
92	CO30, CA20, CA40
93	CO20, CA30, CA24, CO24
94	CA30, CA20, CA40
95	CO20, CO30
96	CO30, CA40, CO24
97	CO30, CA40, CO20
98	CO30, CO24

NoNota	KodeBarang
99	CO24, CA20, CA24, CO20
100	CA20, CA24, CO20

Tabel III.6. Data Transaksi Setelah Diurutkan

NoNota	KodeBarang (*Ordered*)
1	CA20, CA30
2	CO24, CA30
3	CA40, CA20, CA24
5	CA40, CA30
6	CA30, CA24
7	CO30, CA24
8	CA40, CA20, CA30
11	CA40, CA30
12	CA20, CA24
18	CA40, CA20, CA30
19	CA40, CO24, CA20, CA30
20	CO24, CA30
23	CA40, CO20, CO24, CO30
24	CO24, CA20
25	CA40, CA30, CA24
28	CO20, CO30
33	CA40, CO20
34	CA40, CO20, CO24, CO30
35	CA20, CA24,
46	CA40, CO30, CA20
47	CA40, CO20, CA30, CA24
49	CA40, CA20, CA30
50	CA40, CO20, CA24
51	CA40, CO24, CA30
52	CO30, CA20
53	CO20, CA24
55	CO20, CO24, CA20
61	CO30, CA20
62	CA40, CO20, CO24, CA30
63	CO30, CA24,
64	CA40, CO24, CO30, CA20,
65	CA40, CO20, CO24, CA30
67	CO20, CO24,
68	CA40, CO20, CO24, CO30

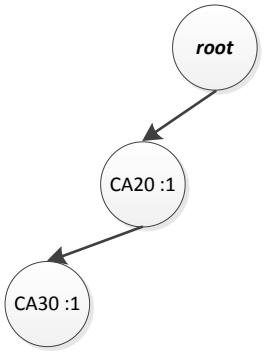
NoNota	KodeBarang (*Ordered*)
69	CO20, CO24, CO30
71	CO24, CO30
75	CA40, CO20, CO30, CA24
77	CO24, CO30, CA20
78	CA40, CA30
79	CO30, CA24
80	CA40, CO20, CO24, CO30
83	CO20, CO30, CA30
84	CA40, CO20
86	CA40, CO20, CO24, CA24
87	CA40, CO20, CO24
91	CA40, CO20, CO24, CO30, CA24
92	CA40, CO30, CA20
93	CO20, CO24, CA30, CA24
94	CA40, CA20, CA30
95	CO20, CO30
96	CA40, CO24, CO30
97	CA40, CO20, CO30
98	CO24, CO30
99	CO20, CO24, CA20, CA24
100	CO20, CA20, CA24

Setelah mendapatkan data yang telah diurutkan, selanjutnya membuat *fp-tree* dari data yang ada pada tabel III.6. Berikut adalah langkah-langkah dalam pembuatan *fp-tree* :

a. Jika *root* belum mempunyai *child*, maka data yang keluar akan menjadi *child* pertama dari *root* tersebut dengan nilai *support count*-nya adalah 1.
b. Jika *root* sudah mempunyai *child*, namun *child* tersebut berbeda dengan data yang keluar, maka *root*-nya memiliki *child* baru lagi dengan nilai *support count*-nya 1.
c. Jika *root* sudah mempunyai *child* dan data yang keluar sama dengan *child* tersebut, maka *support count*-nya bertambah 1.
d. Jika ada *child* yang sama dengan data yang keluar, namun berbeda *path*, *support count*-nya tidak bertambah.

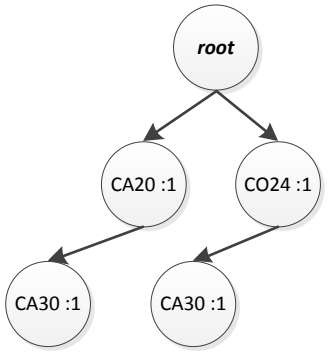
Selanjutnya adalah pembentukan *fp-tree* dari data pada tabel III.6 dengan menggunakan langkah-langkah di atas.

- Untuk tranasaksi dengan NoNota 1 : CA20, CA30. *Support count* dari *node* yang muncul adalah 1 karena data baru pertama kali muncul.



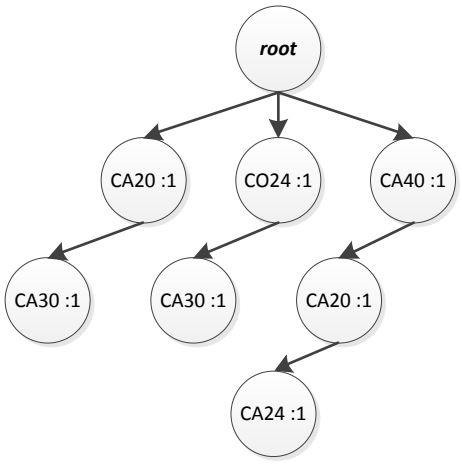
Gambar III.3. *Fp-Tree* setelah pembacaan NoNota 1

- Untuk tranasaksi dengan NoNota 2 : CO24, CA30. *Support count* dari *node* yang muncul adalah 1 karena data baru pertama kali muncul.



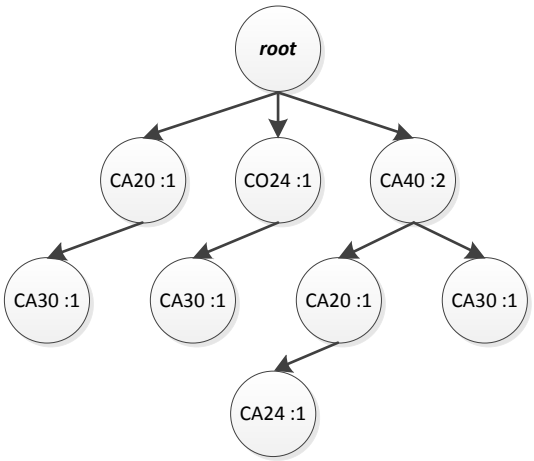
Gambar III.4. *Fp-Tree* setelah pembacaan NoNota 2

- Untuk transaksi dengan NoNota 3 : CA40, CA20, CA24. *Support count* dari *node* yang muncul adalah 1 karena data baru pertama kali muncul.



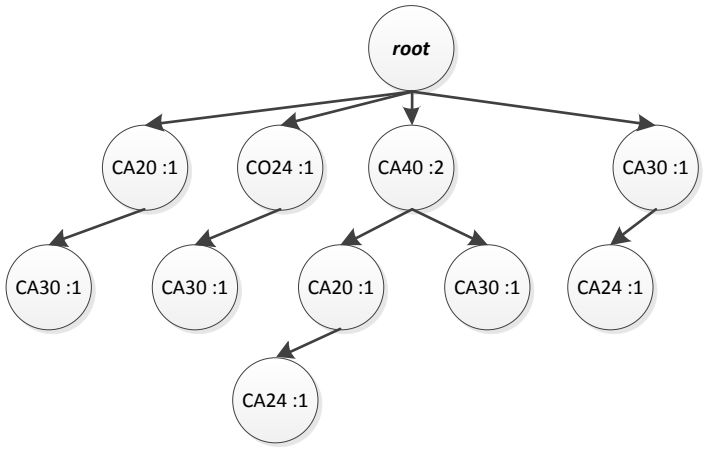
Gambar III.5. *Fp-Tree* setelah pembacaan NoNota 3

- Untuk transaksi dengan NoNota 5 : CA40, CA30. *Support count* dari *node* CA40 bertambah 1 sedangkan CA30 menjadi *child* kedua dari CA40 dengan *support count* 1.



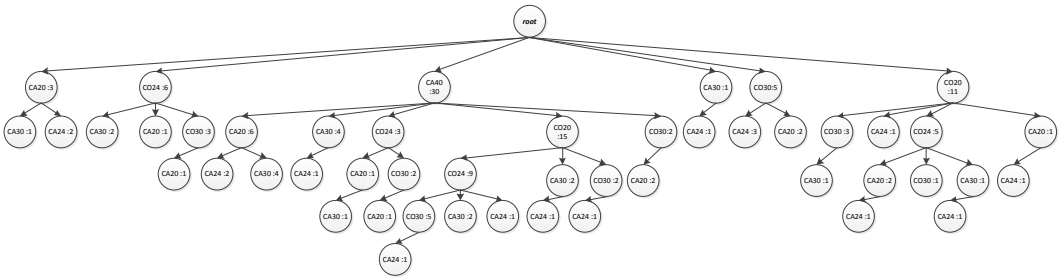
Gambar III.6. *Fp-Tree* setelah pembacaan NoNota 5

- Untuk transaksi dengan NoNota 6 : CA30, CA24. *Support count* dari *node* yang muncul adalah 1 karena data baru pertama kali muncul..



Gambar III.7. *Fp-Tree* setelah pembacaan NoNota 6

- Untuk transaksi dengan NoNota 100 : CO20, CA20, CA24. Karena *node* ini pernah muncul sebelumnya, maka *support count* bertambah 1.



Gambar III.8. *Fp-Tree* setelah pembacaan NoNota 100

b. Penerapan Algoritma *FP-Growth*

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

a. Tahap Pembangkitan *Conditional Pattern Base*

Conditional Pattern Base merupakan subdatabase yang berisi *prefix path* dan *suffix pattern*. Pembangkitan *conditinal pattern base* didapatkan melalui FP-Tree yang telah dibangun sebelumnya.

b. 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 atau sama dengan *minimum support count* ξ akan dibangkitkan dengan *conditional* FP-Tree.

c. Tahap Pencarian *frequent itemset*

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

Tahap Pembangkitan *Conditional Pattern Base*

Berikut adalah langkah-langkah pembentukan *conditional pattern base* :

- Dimulai dari item yang memiliki *priority* paling rendah dalam tabel III.6.
- Telusuri *fp-tree* yang sebelumnya telah dibentuk mulai dari *root* sampai pada item yang memiliki *priority* terendah.
- Akumulasikan semua *path* yang mengandung item tersebut kedalam tabel *conditonal pattern base*.
- Lakukan hal tersebut hingga semua item yang tersedia di tabel III.5 ditelusuri *path*-nya.

Berikut adalah hasil dari *conditional pattern base* dari setiap item :

Tabel III.7. *Conditional Pattern Base*

Item	***Cond. Pattern Base***
CA24	• CA20 : 2 • CA24, CA40 : 2 • CA30, CA40 : 1 • CO30, CO24, CO20, CA40 : 1 • CO24, CO20, CA40 : 1 • CA30, CO20, CA40 : 1

	• CO30, CO20, CA40 : 1 • CA30 : 1 • CO30 : 3 • CO20 : 1 • CA20, CO24, CO20 : 1 • CA30, CO24, CO20 : 1 • CA20, CO20 : 1
CA30	• CA20 : 1 • CO24 : 2 • CA20, CA40 : 4 • CA40 : 4 • CA20, CO24, CA40 : 1 • CO24, CO20, CA40 : 2 • CO20, CA40 : 2 • *Null* : 1 • CO30, CO20 : 1 • CO20 : 1
CA20	• *NULL :* 2 • CO24 : 1 • CO30, CO24 : 1 • CA40 : 6 • CO24, CA40 : 1 • CO30, CA40 : 1 • CO30, CA40 : 2 • CO30 : 2 • CO24, CO20 : 2 • CO20 : 1
CO30	• CO24 : 2 • CO24, CA40 : 2 • CO24, CO20, CA40 : 5 • CO20, CA40 : 2 • CA40 : 2 • *NULL :* 5

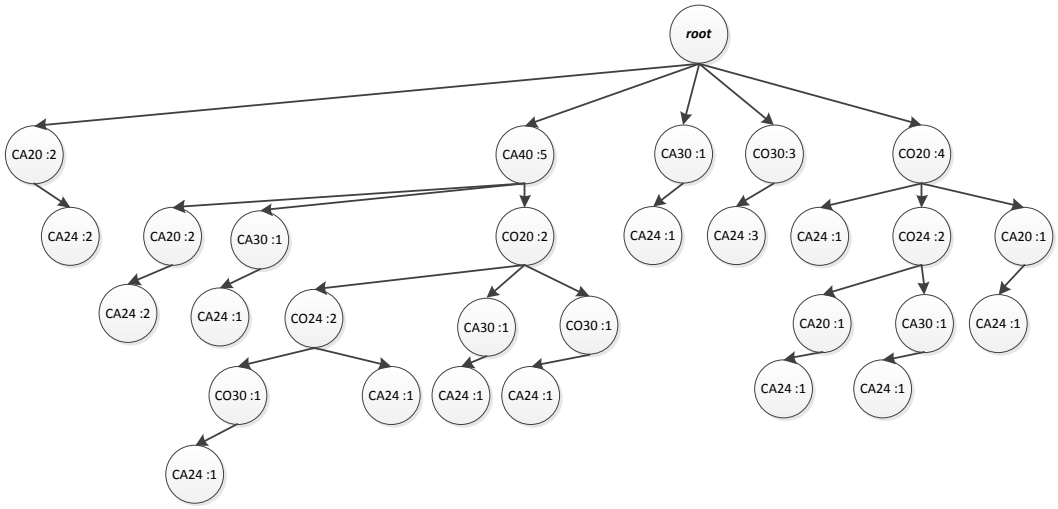
	• CO20 : 3 • CO24, CA40 : 2
CO24	• *NULL* : 6 • CA40 : 3 • CO20, CA40 : 9 • CO20 : 5
CO20	• CA40 : 15 • *NULL :* 11
CA40	• *NULL* : 30

Setelah membuat *conditonal pattern base*, selanjutnya membuat *conditional fp-tree*. Untuk setiap *pattern base* lakukan hal berikut :

a. Akumulasikan *count* untuk setiap item dalam tabel III.7
b. Jika jumlah *count* pada item yang dicari lebih besar atau sama dengan jumlah *count* pada item yang berada pada *path* yang dilewati maka *conditonal fp-tree* dapat dibentuk karena termasuk item yang *frequent.*
c. Selanjutnya jika item tersebut dapat terbentuk *conditional fp-tree*-nya, maka *conditional fp-tree* untuk item tersebut dapat di-*mining* secara rekursif hingga tidak terdapat lagi itemset yang bisa dibentuk.

Berikut adalah tahapan pembentukan *conditional fp-tree* dari tabel III.7 :

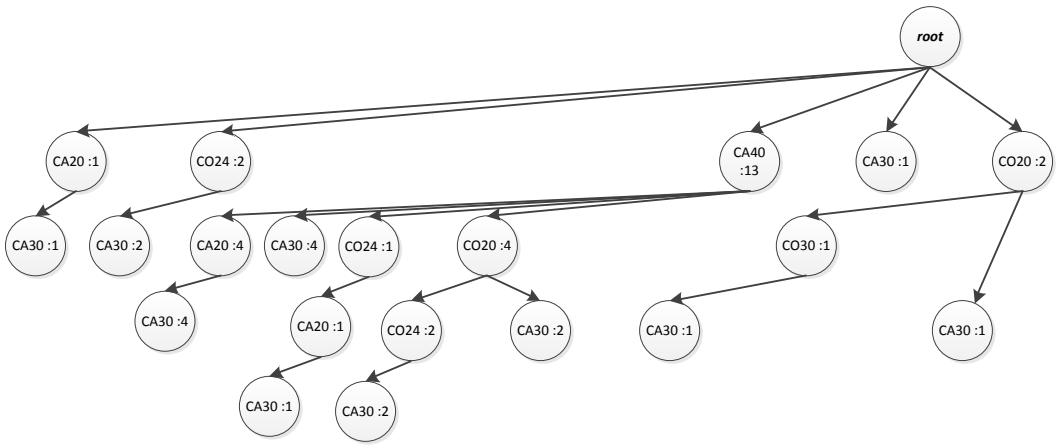
1. Tahap pembentukan *Conditional Fp-Tree* untuk item CA24.



Gambar III.9. FP-Tree yang mengandung CA24

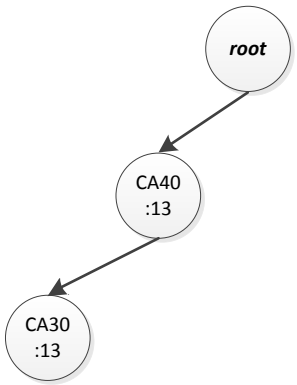
Karena tidak ada kombinasi item yang berakitan dengan CA24 yang memenuhi *minimum support*, maka *conditional fp-tree* dari CA24 tidak bisa dibentuk.

2. Tahap pembentukan *Conditional FP-Tree* untuk item CA30



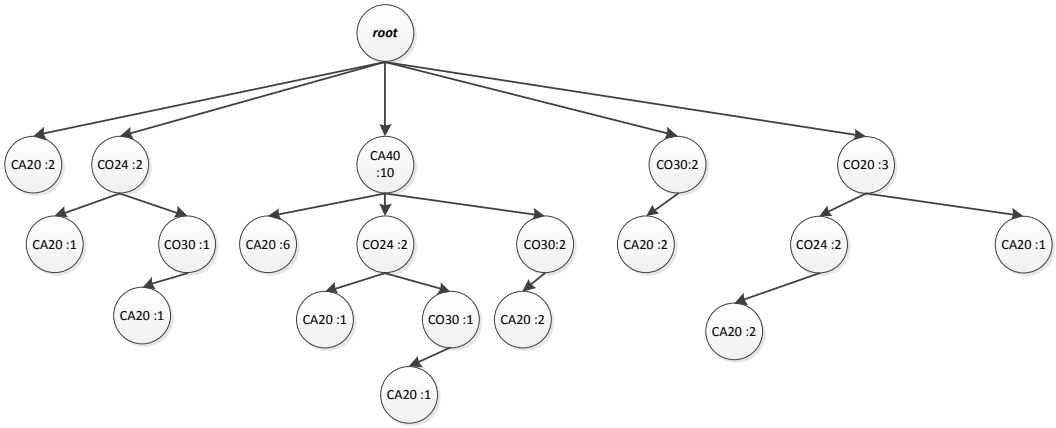
Gambar III.10. FP-Tree yang mengandung CA30

Karena hanya kombinasi CA30, CA40 yang memiliki *support count* yang memenuhi *minimum support*, maka hanya CA30, CA40 yang bisa dibentuk.



Gambar III.11. *Conditional FP-Tree* CA30, CA40

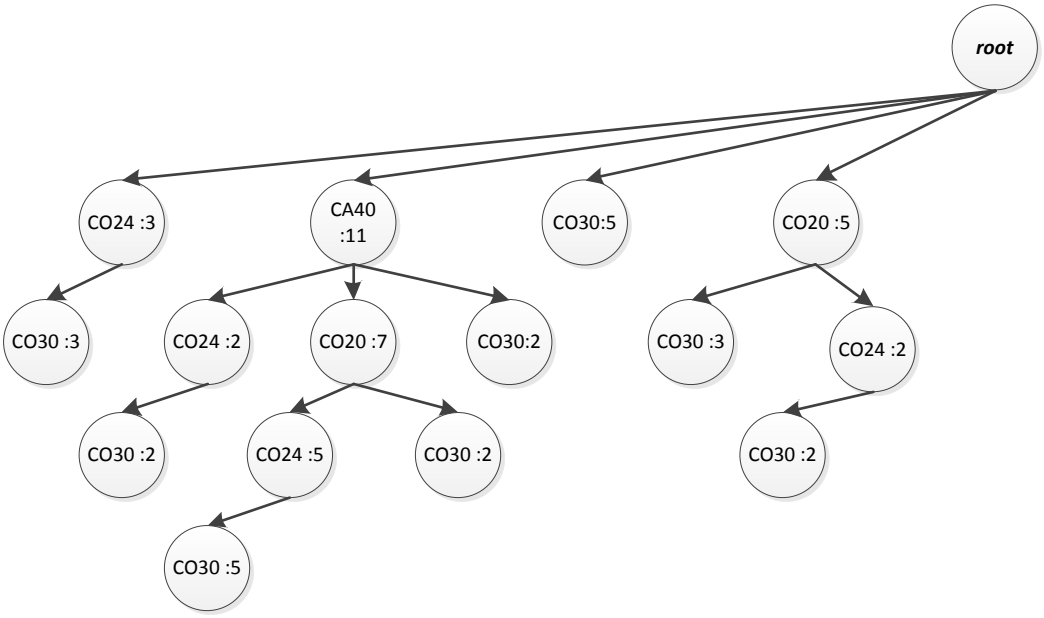
3. Tahap pembentukan *Conditional FP-Tree* untuk item CA20



Gambar III.12. *FP-Tree* yang mengandung CA20

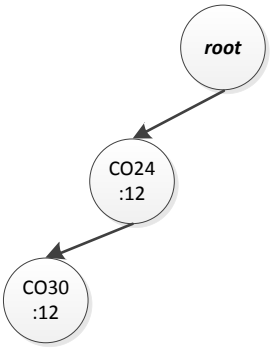
Karena tidak ada kombinasi item yang berakitan dengan CA24 yang memenuhi *minimum support*, maka *conditional fp-tree* dari CA24 tidak bisa dibentuk.

4. Tahap pembentukan *Conditional FP-Tree* untuk item CO30

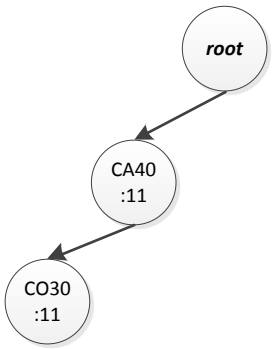


Gambar III.13. *Fp-Tree* yang mengandung CO30

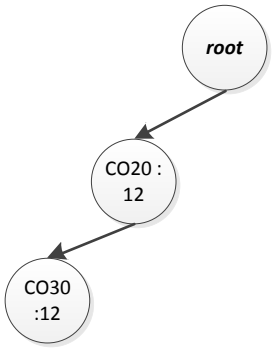
Ada 3 pasanga 2-kombinasi yang memenuhi *minimum support* yaitu (CO30, C024), (CO30, CA40), (CO30, CO20).



Gambar III.14. *Conditional FP-Tree* CO30, CO24

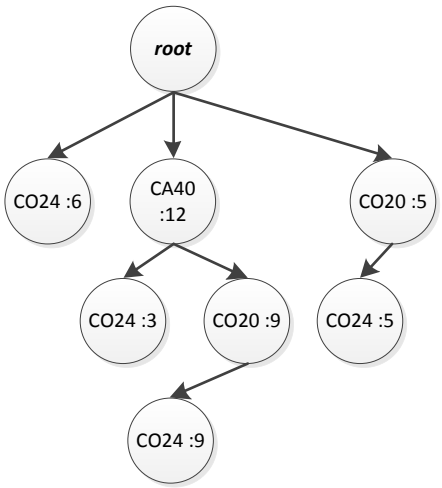


Gambar III.15. *Conditional FP-Tree* CO30, CA40



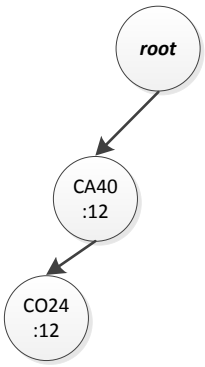
Gambar III.16. *Conditional FP-Tree* CO30, CO20

5. Tahap pembentukan *Conditional FP-Tree* untuk item CO24

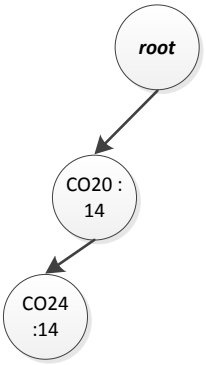


Gambar III.17. *FP-Tree* yang mengandunug CO24

Hanya ada 2 pasanga 2-kombinasi yang mmenuhi *minimum support* yaitu, (CO24, CO20) dan (CO24, CA40).

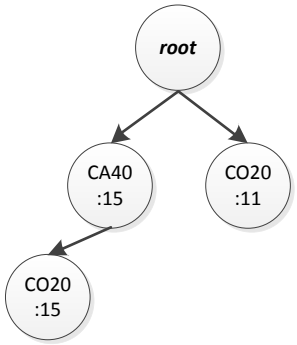


Gambar III.18. *Conditional FP-Tree* untuk CO24, CA40



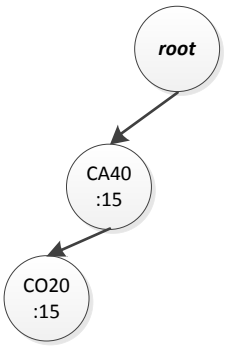
Gambar III.19. *Conditional FP-Tree* untuk CO24, CO20

6. Tahap pembentukan *Conditional FP-Tree* untuk item CO20



Gambar III.20. *FP-Tree* yang mengandung CO20

Hanya kombinasi CO20, CA40 yang memenuhi *minimum support.*



Gambar III.21. *Conditonal FP-Tree* untuk CO20, CA40

Setelah proses pembentukan *conditional fp-tree*, selanjutnya dilakukan pencarian *frequent itemset*. Agar memudahkan dalam pembacaan data, maka hasil dari *conditional fp-tree* dimasukkan ke dalam bentuk tabel. Berikut adalah hasil dari *conditional fp-tree :*

Tabel III.8. Hasil Conditional FP-Tree

Item	*Cond. Pattern Base*	*Cond. FP-Tree*
CA24	• CA20 : 2 • CA24, CA40 : 2 • CA30, CA40 : 1	*empty*

	• CO30, CO24, CO20, CA40 : 1 • CO24, CO20, CA40 : 1 • CA30, CO20, CA40 : 1 • CO30, CO20, CA40 : 1 • CA30 : 1 • CO30 : 3 • CO20 : 1 • CA20, CO24, CO20 : 1 • CA30, CO24, CO20 : 1 • CA20, CO20 : 1	
CA30	• CA20 : 1 • CO24 : 2 • CA20, CA40 : 4 • CA40 : 4 • CA20, CO24, CA40 : 1 • CO24, CO20, CA40 : 2 • CO20, CA40 : 2 • *Null* : 1 • CO30, CO20 : 1 • CO20 : 1	CA30,CA40 : 13
CA20	• *NULL :* 2 • CO24 : 1 • CO30, CO24 : 1 • CA40 : 6 • CO24, CA40 : 1 • CO30, CA40 : 1 • CO30, CA40 : 2 • CO30 : 2 • CO24, CO20 : 2 • CO20 : 1	*empty*
CO30	• CO24 : 2 • CO24, CA40 : 2	• CO30, CO24 : 12 • CO30, CA40 : 11

	• CO24, CO20, CA40 : 5 • CO20, CA40 : 2 • CA40 : 2 • *NULL :* 5 • CO20 : 3 • CO24, CA40 : 2	• CO30, CO20 : 12
CO24	• *NULL* : 6 • CA40 : 3 • CO20, CA40 : 9 • CO20 : 5	• CO24, CA40 : 12 • CO24, CO20 : 14
CO20	• CA40 : 15 • *NULL :* 11	CO20, CA40 : 15
CA40	• *NULL* : 30	*empty*

Setelah proses pembentukan *conditional fp-tree* dilakukan, selanjutnya melakukan prmbentukan tabel *frequent itemset* dari hasil *conditional fp-tree* yang telah dibentuk sebelumnya. Hasil dari tahap pembentukan *conditional fp-tree* dapat dilihat pada tabel III.9.

Tabel III.9. Frequent Itemset

Item	***Frequent Itemset***
CA24	*Empty*
CA30	• *CA30* • CA30,CA40 : 13
CA20	*Empty*
CO30	• *CO30* • CO30, CO24 : 12, CO30, CA40 : 11, CO30, CO20 : 12
CO24	• CO24 • CO24, CA40 : 12, CO24, CO20 : 14
CO20	• CO20 • CO20, CA40 : 15
CA40	*Empty*

Generate Rule

Dari hasil *frequent itemset* selanjutnya dilakukan perhitungan *confidence* dari setiap itemset dengan persamaan II.4. Hasil dari perhitungan dapat dilihat pada tabel dibawah ini :

Tabel III.10. Perhitungan Confidence

Item	***Confidence***
CO20 → CA40	57.6 %
CO24 → CA40	54.1 %
CO24 → CO20	58.3 %
CO30 → CA40	45.8 %
CO30 → CO20	45.8 %
CO30 → CO24	45.8 %
CA30 → CA40	68. 4%

Setelah *confidence* dari setiap *itemset* dihitung, selanjutnya dilakukan *generate* dengan *minimum support* = 10 dan *minimum confidence* = 50%. Adapun hasil *generate rule* dapat dilihat pada tabel berikut ini :

Tabel III.11. Generate Rule

Item	***Confidence***
CO20 → CA40	57.6 %
CO24 → CA40	54.1 %
CO24 → CO20	58.3 %
CO30 → CA40	45.8 %
CO30 → CO20	45.8 %
CO30 → CO24	45.8 %
CA30 → CA40	68. 4%

Kesimpulan berdasarkan *rule* yang dihasilkan maka pola pembelian pelanggan adalah sebagai berikut :

Tabel III.12. Hasil perhitungan pengujian metode association rule dengan algoritma fp-growth

Kombinasi	*Support*	*Confidence*
CO20 → CA40	15	57.6 %
CO24 → CA40	13	54.1 %
CO24 → CO20	14	58.3 %
CA30 → CA40	13	68.4 %

III.1.2. Analisis Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak adalah kebutuhan-kebutuhan apa saja yang diperlukan untuk membangun sebuah perangkat lunak berdasarkan kebutuhan pengguna. Ada dua bagian dalam spesifikasi kebutuhan perangkat lunak yaitu kebutuhan *non-fungsional* dan kebutuhan *fungsional*.

III.1.3. Analisis Kebutuhan Non-Fungsional

Analisis kebuthuan *non-fungsional* dilakukan untuk mengetahui spesifikasi kebutuhan sistem. Spesifikasi kebuthan melibatkan analisis perangkat keras (*hardware*), analisis perangkat lunak (*software*), dan analisis pengguna (*user*).

III.1.4.1. Analisis Kebutuhan Perangkat Keras

Perangkat keras pendukung dalam pembangunan perangkat keras lunak ini adalah sebagai berikut :

1. Analisi spesifikasi kebutuhan perangkat keras pada sistem yang sedang berjalan di PT. Bandung Perkasa Jaya adalah :
   a. *Processor :* Intel Core Duo
   b. *Harddisk :* 160 GB
   c. *Memory :* 2 GB
2. Analisis spesifikasi *minimum* kebutuhan perangkat keras pada sistem yang akan dibangun membutuhkan :

a. *Processor* berkecepatan minimal 2 Ghz
b. *Harddisk* 1 Gb untuk menyimpan data
c. *Memory* 512 MB

3. Evaluasi kebutuhan perangkat keras

   Dari data spesifikasi perangkat keras yang dimiliki oleh PT. Bandung Perkasa Jaya, sudah cukup untuk menjalankan aplikasi pengolahan data transaksi yang akan dibangun.

III.1.4.2. Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak pendukung merupakan hal penting bagi pembangunan perangkat lunak yang sedang dibangun. Perangkat lunak pendukung dalam pembangunan perangkat lunak ini adalah sebagai berikut :

1. Analisis kebutuan perangkat lunak pada sistem yang sedang berjalan di PT. Bandung Perkas Jaya adalah :
   a. Sistem operasi Windows 7
   b. *Microsoft Office*
2. Analisis kebutuhan perangkat lunak untuk menjalankan sistem yang akan dibangun adalah :
   a. Sistem operasi Windows 7
   b. Wamserver sebagai *database server*
   c. *Visual Studio* 2012
3. Evaluasi kebutuhan perangkat lunak

   Berdasarkan analisis spesifikasi perangkat lunak yang ada di PT. Bandung Perkasa Jaya, spesifikasi pernagkat lunak yang akan digunakan kurang memenuhi spesifikasi lunak yang dibutuhkan. Kekurangan tersebut dapat diatasi dengan meng-*install* WampServer dan Visual Studio pada komputer yang digunakan.

III.1.4.3. Analisis Pengguna

Analisis pengguna dibuat untuk memenuhi siapa saja dan seperti apa karakteristik yang dimiliki oleh orang yang dapat menggunakan aplikasi yang akan dibuat. Untuk lebih detailnya dapat dilihat pada tabel III.13.

Tabel III.13. Karakteristik Pengguna

Pengguna	Tanggung Jawab	Hak Akses	Tingkat Pendidikan	Tingkat Keterampilan	Pengalaman
User	Mengoperasikan aplikasi sesuai dengan tugas yang diberikan	Melakukan kegiatan yang berhubungan dengan isis dari tampilan yang ada pad aaplikasi	minimal Sekolah Menengah Atas (SMA)	Memahami konten yang tersedia pada aplikasi	Minimal mampu mengoperasikan komputer

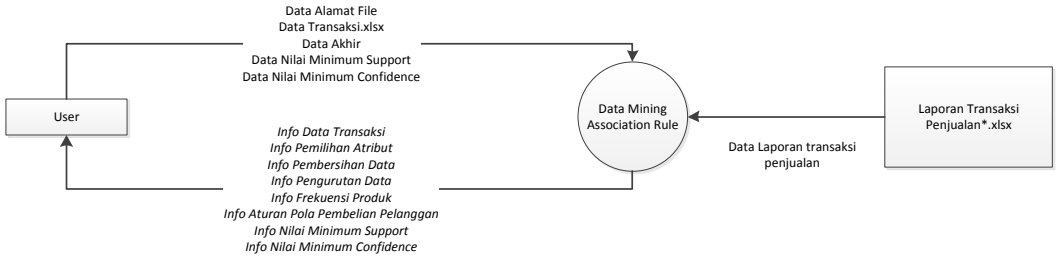
Dengan kondisi yang terdapat diatas, dapat diambil kesimpulan bahwa aplikasi yang dibuat dapat dijalankan oleh staf diperusahaan tersebut. Staf dalam hal ini adalah orang yang bekerja di bagian produksi yang akan menggunakan aplikasi yang dibangun.

III.1.4. Analisis Kebutuhan Fugsional

Analisis kebutuhan fungsional yang akan dibuat meliputi diagram konteks, *Data Flow Diagram* (DFD), spesifikasi proses dan kamus data.

III.1.5.1. Diagram konteks

Berikut adalah diagram konteks dari sistem pendekatan struktur yang menggambarkan sistem secara garis besar yang kemudian akan dipecah menjadi bagian-bagian yang lebih detail. Adapaun diagram konteks dari sistem yang akan dibangun adalah sebagi berikut :



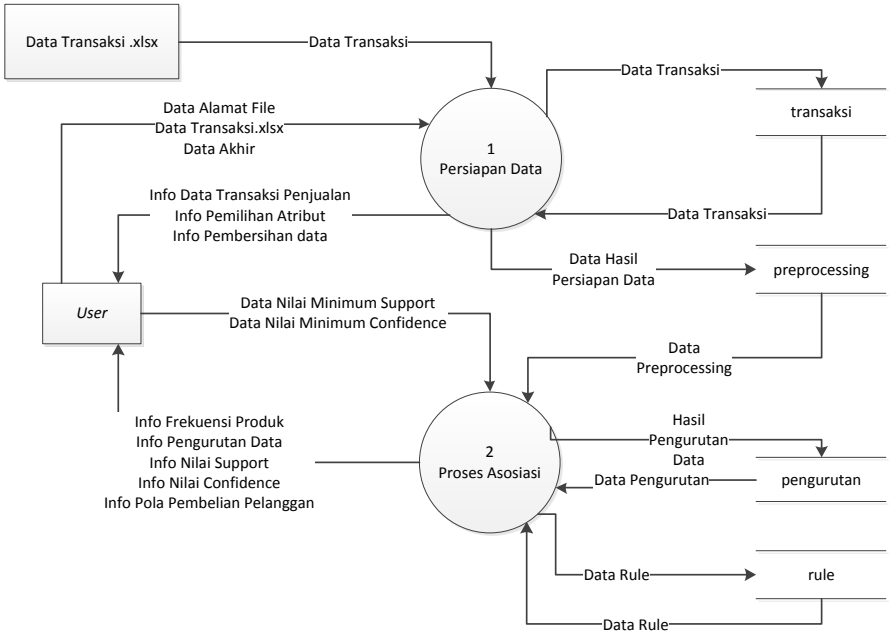
Gambar III.22. Diagaram Konteks

III.1.5.2. Data Flow Diagram

Data Flow Diagram (DFD) merupakan proses yang menggambarkan suatu alir informasi yang lebih terperinci yang merupakan pengembangan dari diagram konteks. Berikut ini merupakan *data flow diagram* dari sistem yang akan dibangun :

a. DFD Level 1

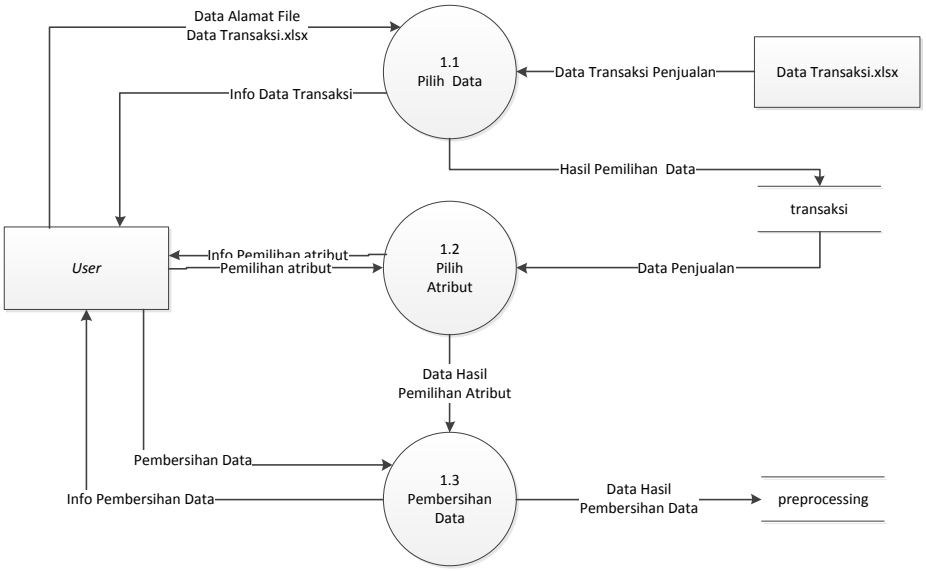
Berikut adalah gambar DFD level 1 dapat dilihat pada Gambar III.23 dibawah ini :



Gambar III.23. DFD Level 1 Aplikasi Data mining Association Rule PT.Bandung Perkasa Jaya

b. DFD Level 2 untuk Proses 1

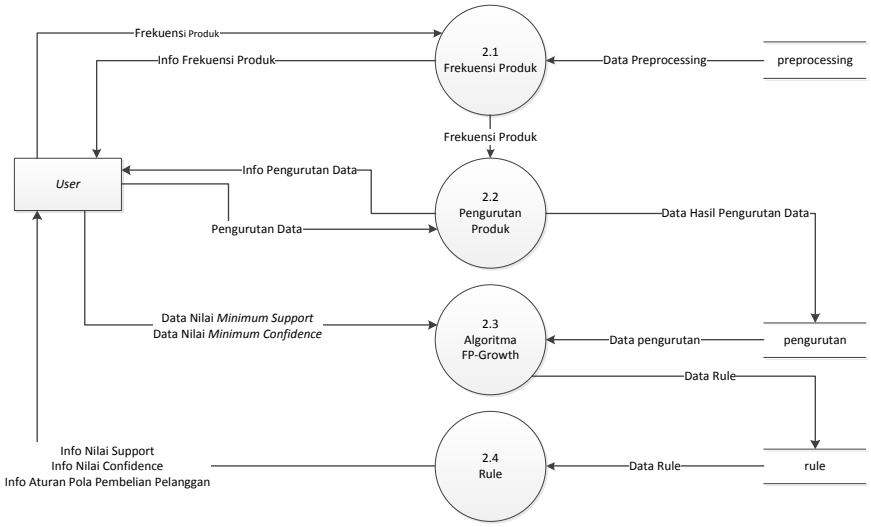
Berikut adalah gambar DFD level 2 Proses 1 dapat dilihat pada Gambar 3.24 dibawah ini :



Gambar III.24. DFD Level 2 Proses 1

c. DFD Level 2 Proses 2

Berikut adalah gambar DFD level 2 Proses 2 dapat dilihat pada Gambar 3.25 dibawah ini :



Gambar III.25. DFD Level 2 Proses 2

III.1.5.3. Spesifikasi Proses

Spesifikasi proses berfungsi untuk menjelaskan proses model aliran yang terdapat pada DFD. Adapun spesifikasi proses dari DFD yang telah dibuat akan dijelaskan pada tabel berikut :

Tabel III.14. Spesifikasi Proses

No	Proses	Keterangan
1	No Proses	1
	Nama Proses	Persiapan data
	Source (sumber)	*User*
	Input	-Alamat data transaksi penjualan.xlsx -Pemilihan atribut -Pembersihan data
	Output	-Info data transaksi penjualan -Info pemilihan atribut -Info pembersihan data
	Destination (tujuan)	simpan data ke *database*
	Logika Proses	1. Aplikasi menampilkan form pilih data 2. *User* memilih data yang akan dianalisa 3. *User* menyimpan data ke *database* 4. Melakukan pemilihan atribut 5. Melakukan pembersihan data
2	No Proses	1.1
	Nama Proses	Pilih Data
	Source (sumber)	*User*
	Input	Alamat data transaksi penjualan
	Output	-Info data transaksi penjualan -Info jumlah data transaksi penjualan
	Destination (tujuan)	Simpan data ke *database*
	Logika Proses	1. Aplikasi menampilkan *form* untuk memilih data yang akan digunakan 2. Aplikasi menerima alamat data transaksi dan mengekstraknya ke dalam aplikasi 3. *User* dapat melihat data
3	No Proses	1.2
	Nama Proses	Pilih Atribut
	Source (sumber)	*User*
	Input	Data transaksi
	Output	Info pemilihan atribut
	Destination (tujuan)	- *User* - Pembersihan data

	Logika Proses	1. *User* melakukan pemilihan atribut 2. Aplikasi menampilkan hasil pemilihan atribut
4	No Proses	1.3
	Nama Proses	Pembersihan data
	Source (sumber)	*User*
	Input	Data hasil Pemilihan atribut
	Output	Info pembesihan data
	Destination (tujuan)	- *User* - *Preprocessing*
	Logika Proses	1. Aplikasi menerima data hasil pemilihan atribut 2. *User* melakukan pembersihan data 3. Aplikasi menampilkan hasil pembersihan data 4. hasil pembersihan data di simpan ke tabel *preprocessing*
5	No Proses	2
	Nama Proses	Penentuan *rule* pola pembelian pelanggan
	Source (sumber)	*User*
	Input	-Frekuensi Produk -Pengurutan data -Data nilai *minimum support* -Data nilai *minimum confidence*
	Output	-info frekuensi produk -info pengurutan data -info nilai *support* -info nilai *confidence* -info *rule* pola pembelian pelanggan
	Destination (tujuan)	*User*
	Logika Proses	1. Aplikasi menampilkan *form* aturan asosiasi 2. Aplikasi menampilkan frekuensi setiap produk 3. Setelah frekuensi diketahui *user* melakukan pengurutan data 4. Aplikasi menampilkan hasil pengurutan data 5. *User* menginputkan nilai *minimum support* dan nilai *minimum confidence* 6. Aplikasi menampilkan Info nilai *support*, nilai *confidence* dan aturan penentuan kombinasi produk
6	No Proses	2.1
	Nama Proses	Frekuensi produk
	Source (sumber)	*User*
	Input	Data *Preprocessing*
	Output	Info frekuensi setiap produk
	Destination (tujuan)	- *User* - Pengurutan data
	Logika Proses	1. Aplikasi mengambil data dari tabel *preprocessing* 2. Menampilkan info frekuensi produk

7	No Proses	2.2
	Nama Proses	Pengurutan Data
	Source (sumber)	*User*
	Input	Data frekuensi produk
	Output	Menampilkan info pengurutan data
	Destination (tujuan)	- *User* - simpan pengurutan data ke *database*
	Logika Proses	1. Aplikasi mengambil nilai dari hasil frekuensi produk 2. Aplikasi mengeluarkan tabel dari proses hasil frekuensi produk 3. Setelah tabel dikeluarkan aplikasi mengurutkan berdasarkan frekuensi kemunculan terbanyak 4. Menampilkan hasil pengurutan data
8	No Proses	2.3
	Nama Proses	Algoritma *FP-Growth*
	Source (sumber)	*User*
	Input	-Input nilai *support* -Input nilai *confidence*
	Output	Data *rule*
	Destination (tujuan)	*Rule*
	Logika Proses	1. *User* menginputkan data nilai *minimum support* dan data nilai *minimum confidence* 2. Aplikasi mengambil data hasil *preprocessing* yang sudah diurutkan 3. Aplikasi membuat struktur data *fp-tree* 4. Aplikasi menghasilkan frekuensi *itemset* dan *rule*

III.1.5.4. Kamus Data

Kamus data berfungsi untuk menjelaskan semua data yang digunakan di dalam sistem. Berikut adalah kamus data untuk perangkat lunak yang akan dibangun :

Nama Data	**Data Transaksi**
Deskripsi	File excel data transaksi
Struktur Data	NoNota, KodeBarang, NamaBarang, Jumlah, Tanggal
NoNota	{0..9}
KodeBarang	{A..Z | a..z} {0..9}
Nama Barang	{A..Z | a..z} {0..9}
Jumlah	{0..9}
Tanggal	{0..9}
Nama Data	***Preprocessing***

Deskripsi	Berisi data hasil preprocessing dari data transaksi
Struktur Data	NoNota, KodeBarang
No Nota	{0..9}
KodeBarang	{A..Z | a..z} {0..9}
Nama Data	**Pengurutan**
Deskripsi	Berisi data yang sudah diurutkan dan siap di *mining*
Struktur Data	KodeBarang, Frekuensi
KodeBarang	{A..Z | a..z} {0..9}
Frekuensi	{0..9}
Nama Data	**Rule**
Deskripsi	Berisi aturan pola pembelian pelanggan
Struktur Data	Rule, *Support*, *Confidence*
Rule	{A..Z|a..z|0..9}
Support	{0..9}
Confidence	{0..9}
Deskripsi	Presentase nilai kepastian kuatnya hubungan antar-item dalam aturan asosiasi yang diinginkan *user*

III.1.5.5. Perancangan Sistem

Proses perancangan yang akan dilakukan pada tahapan ini adalah perancangan sistem basis data, perancangan struktur menu, perancangan antar muka, perancangan pesan, jaringan semantik dan perancangan procedural

Perancangan Basis Data

Perancangan tabel pada aplikasi yang akan dibangun meliputi struktur tabel

a. Struktur Tabel

Struktur tabel pada aplikasi yang akan dibangun terdiri dari 3 tabel dengan rincian sebagai berikut :

Tabel III.15. . Struktur Tabel Data Transaksi

Nama Atribut	Tipe Data	Panjang	Keterangan
No Nota	Varchar	6	*Not Null*
KodeBarang	Varchar	4	*Not Null*
Nama Barang	Varchar	25	*Not Null*
Jumlah	Int	11	*Not Null*
Tanggal	Varchar	15	*Not Null*

Tabel III.16. Struktur Tabel *Preprocessing*

Nama Atribut	Tipe Data	Panjang	Keterangan
No Nota	Varchar	6	*Not Null*
KodeBarang	Varchar	4	*Not Null*

Tabel III.17. Struktur Tabel Pengurutan

Nama Field	Tipe Data	Panjang	Keterangan
Kodebarang	Varchar	6	*Not Null*
Frekuensi	Int	8	*Not Null*

Tabel III.18. Struktur Tabel *Rule*

Nama Field	Tipe Data	Panjang	Keterangan
Rule	Varchar	25	*Not Null*
Support	Float		*Not Null*
Confidence	Float		*Not Null*

Perancangan Antar Muka

Dalam perancagan sebuah perangkat lunak dibutuhkan struktur menu yang berisikan menu dan submenu yang berfungsi untuk memudahkan *user* dalam menggunakan perangkat lunak tersebut. Berikut ini digambarkan mengenai struktur menu dalam aplikasi ini :

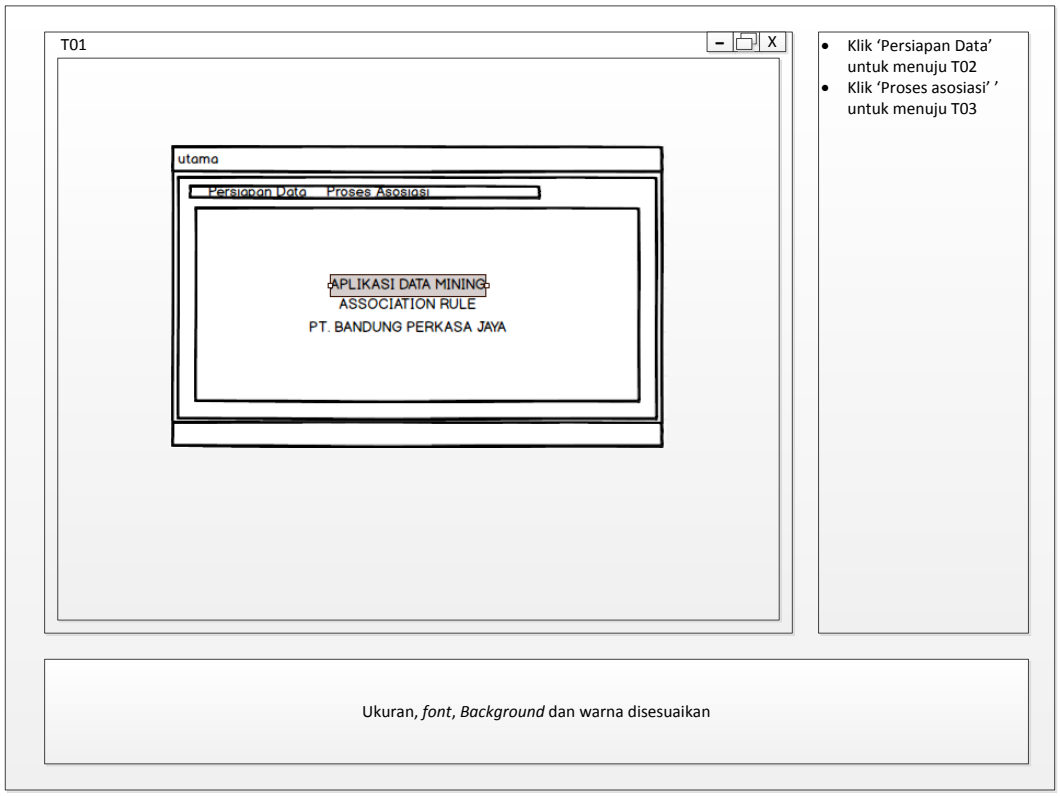


Gambar III.26. Struktur Menu

a. Perancangan Tampilan Program

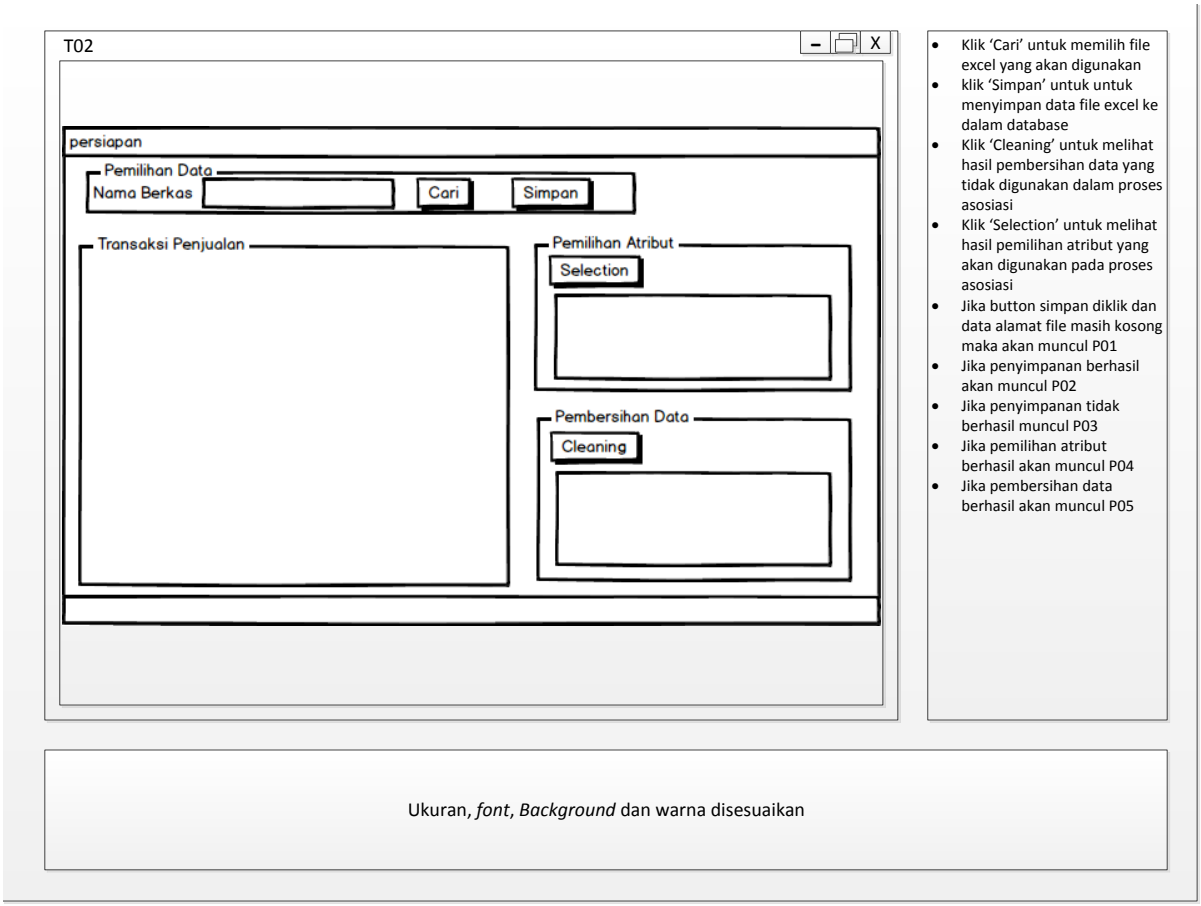
Perancangan tampilan program dari aplikasi data mining association rule yang akan dibangun adalah sebagai berikut :

a. Tampilan Halaman Utama



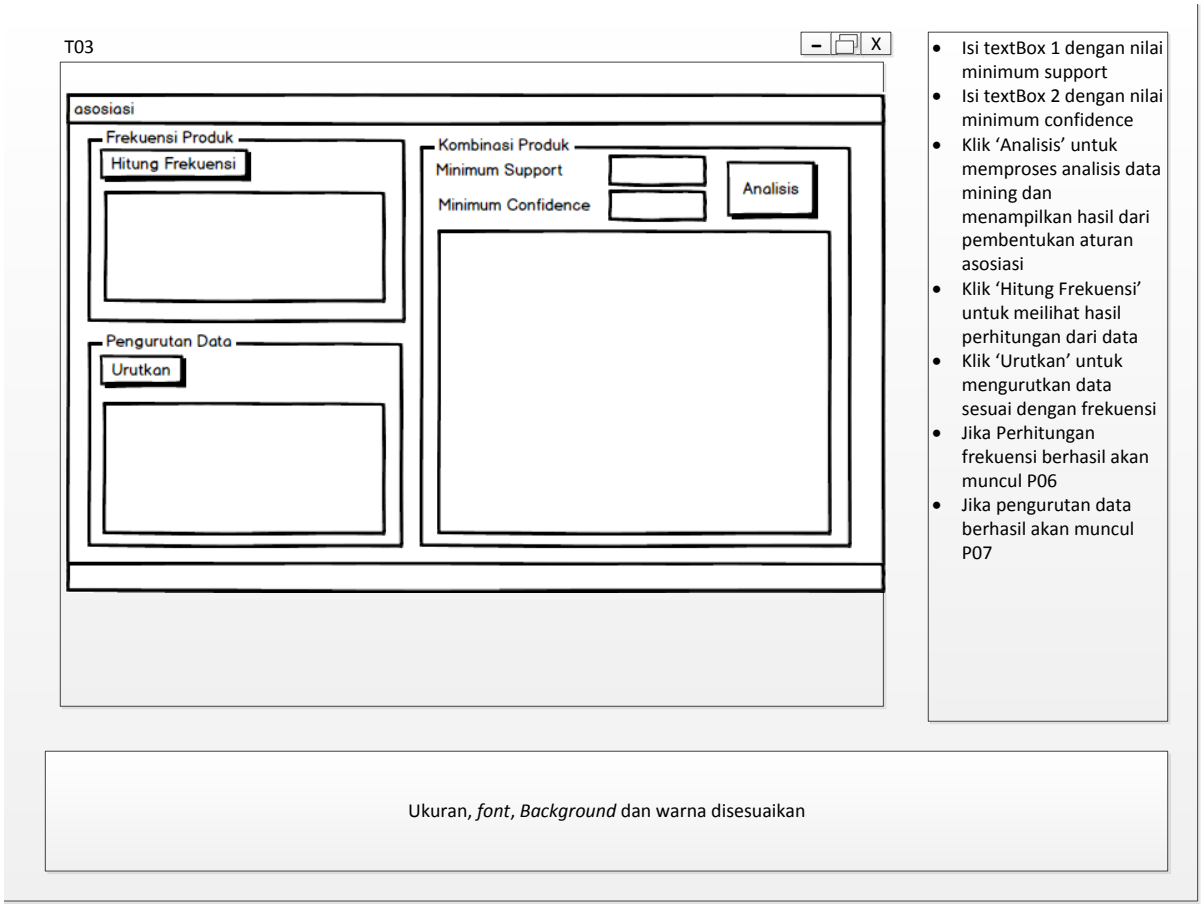
Gambar III.27. Tampilan Halaman Utama (T01)

b. Tampilan Preprocessing Data



Gambar III.28. Tampilan Preprocessing Data (T02)

c. Tampilan Proses Asosiasi



Gambar III.29. Tampilan Pembentukan Aturan Asosiasi (T03)

Perancangan Tampilan Pesan

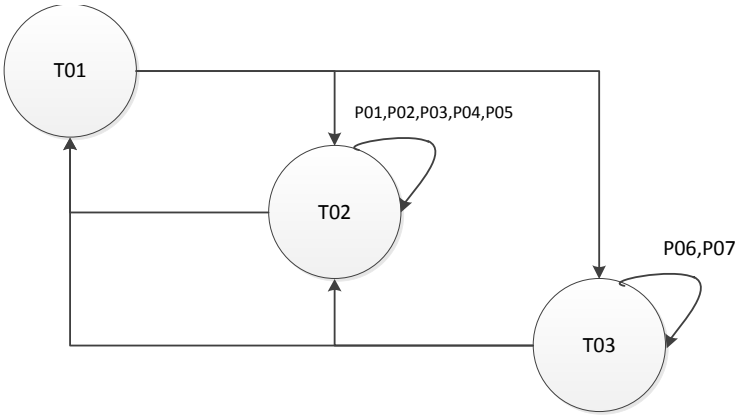
Berikut ini adalah perancangan pesan dari aplikasi yang akan dibangun.



Gambar III.30. Perancangan Pesan

Jaringan Semantik

Berikut ini adalah jaringan sematik dari perangkat lunak yang akan dibuat untuk menunjukan keterhubungan antar objek perancangan.



Gambar III.31. Jaringan Semantik