BAB 2
TINJAUAN PUSTAKA

2.1 Profil Instansi

Berikut adalah profil dari Primer Koperasi Kartika Baja. Dalam profil ini akan membahas tentang sejarah, logo, struktur organisasi, dan visi misi mengenai Primer Koperasi Kartika Baja Cilegon.

2.1.1 Sejarah Primer Koperasi Kartika Baja

Berdirinya Primer Koperasi Kartika Baja tentunya tidak terlepas dari keberadaan satuan itu sendiri karena Primer Koperasi Kartika Baja selalu berdiri bernaung di dalam Kodim 0623 Cilegon. Karena keberadaan Primer Koperasi Kartika Baja tidak terlepas dari Kodim 0623 Cilegon, pertama akan diulas dahulu mengenai satuan itu sendiri. Kodim 0623 Cilegon berada di bawah naungan Korem 064/Maulana Yusuf Serang, Banten di mana Korem itu sendiri membawahi beberapa Kodim.

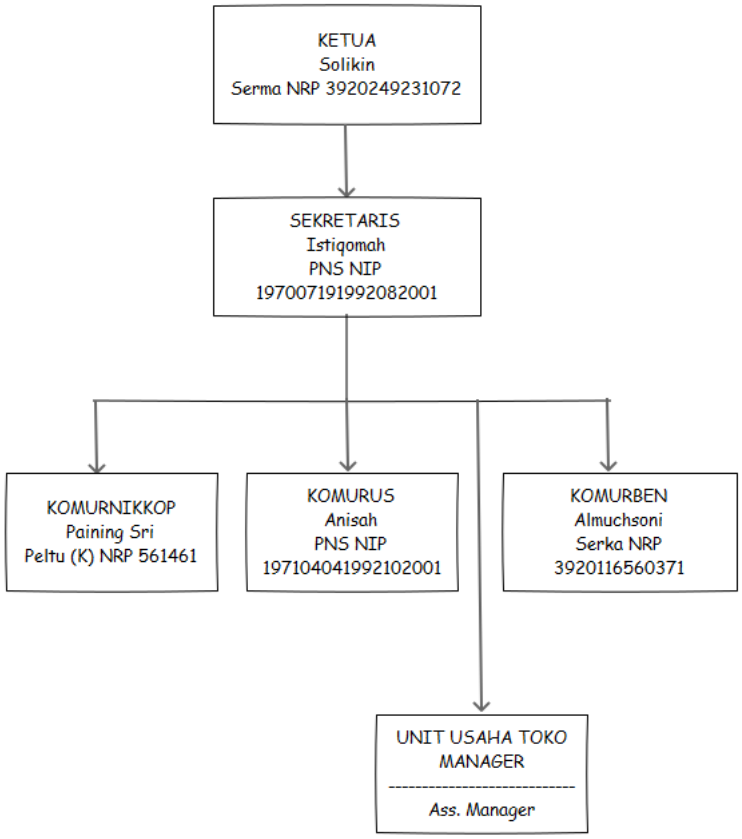
2.1.2 Logo

Berikut ini adalah logo Primer Koperasi Kartika Baja. Bisa dilihat pada gambar 2.1:



Gambar 2. 1 Logo Primer Koperasi Kartika Baja2.1.3 Struktur Organisasi

Berikut ini adalah struktur organisasi dari Primer Koperasi Kartika Baja yang dapat dilihat pada Gambar 2.2:



Gambar 2. 2 Struktur Organisasi Primkop Kartika Baja

Job description:

1. Ketua bertanggung jawab serta mengatur seluruh anggota koperasi. Tugas dan tanggung jawabnya adalah sebagai berikut:
   a. Memimpin, mengkoordinir dan mengendalikan serta mengawasi anggota koperasi.
   b. Menanda tamgani dan mengesahkan semua surat-surat yang menyangkut kegiatan koperasi.
2. Sekretaris memiliki tugas dan tanggung jawab sebagai berikut:
   a. Membantu Ketua dalam berbagai urusan koperasi.
   b. Mengelola dan merekap data koperasi.

c. Mendata semua kegiatan koperasi dan membuat laporan bulanan.

3. KOMURNIKKOP memiliki tugas dan tanggung jawab sebagai berikut:
   a. Melaksanakan pembinaan, pengawasan dan penelitiaan bidang dan teknik perkoperasian pada Primkopad.
   b. Menjalin hubungan dan kerjasama dengan Kakandepkop Dekopinda dan Primkopad-Primkopad lain dalam rangka meningkatkan usaha Primkopad.
   c. Menyusun rencana kerja dan rencana anggaran belanja untuk bidang tugasnya serta mengawasi pelaksanaannya.
4. KOMURUS memiliki tugas dan tanggung jawab sebagai berikut:
   a. Memantapkan, mengendalikan dan mengembangkan kegiatan usaha yang ada, baik yang langsung melayani kepentingan anggota maupun kepentingan umum.
   b. Merencanakan pola jaringan pasar antar Kopad, Kopad dengan koperasi lain dan masyarakat umum dengan berperan sebagai koordinator pengadaan dan pemasarannya.
   c. Meningkatkan dan mengembangkan usaha-usaha dengan mitra baik untuk kepentingan anggota dan masyarakat pada umumnya.
5. KOMURBEN memiliki tugas dan tanggung jawab sebagai berikut:
   a. Menyelenggarakan pembinaan sistem administrasi dan pembukuan keuangan pada Primkopadnya.
   b. Mengumpulkan, mengolah data menyajikan data/keterangan di bidang administrasi pembendaharaan.
   c. Mengadakan analisa dan evaluasi serta menyampaikan saran tindakan dalam rangka pengendalian pembendaharaan di lingkungan Primkopad maupun badan usaha Primkopad.
6. Unit Usaha Toko Manager memiliki tugas dan tanggung jawab sebagai berikut:
   a. Menyampaikan pertimbangan dan saran kepada Ketua Primkopad mengenai hal-hal yang berhubungan dengan koperasi.
   b. Membantu dalam megurus Primkopad.

2.1.4 Visi dan Misi

Visi dari Primer Koperasi Kartika Baja adalah:

1. Mendirikan koperasi yang bersifat kekeluargaan.
2. Mewujudkan kesejahteraan para pegawai Kodim 0623 Cilegon maupun warga sipil

Misi dari Primer Koperasi Kartika Baja adalah:

1. Mengutamakan pelanggan dengan pelayanan terbaik.
2. Memberi inovasi pada sistem penjualan secara berkala.
3. Mengembangkan koperasi agar lebih besar dan lebih cepat dalam meambah variasi-variasi barang baru.

2.2 Landasan Teori

Dalam sub bab ini akan dijelaskan mengenai definisi dan teori-teori yang berkaitan dengan penelitian ini. Berikut adalah teori-teori yang mendasari dari perancangan sistem penentuan paket hemat berbasis *desktop*.

2.2.1 Data

Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya [3]. Jumlah informasi yang tersedia bagi kita sangat banyak, dan nilai data sebagai asset organisasi telah diakui secara luas. Untuk memanfaatkan *dataset* yang besar dan kompleks, pengguna memerlukan alat yang memudahkan tugas mengatur data dan mengekstraksi informasi yang berguna dalam cara yang baik. Sebaliknya, data dapat menjadi sesuatu yang tidak berguna dengan biaya perolehan dan pengelolaannya lebih besar daripada nilai yang dapat kita ambil dari data tersebut [4].

2.2.2 Database

Database adalah kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi yang berhubungan atau lebih. Misalnya, *database* universitas mungkin berisi informasi mengenai hal berikut:

a. Entitas seperti mahasiswa fakultas, mata kuliah, dan ruang kuliah.
b. Hubungan antara entitas, seperti registrasi mahasiswa dalam mata kuliah, fakultas yang mengajarkan mata kuliah, dan penggunaan ruang untuk kuliah [4].

2.2.3 Data Mining

Berikut ini adalah penjelasan mengenai pengertian dan konsep yang terdapat pada *data mining.*

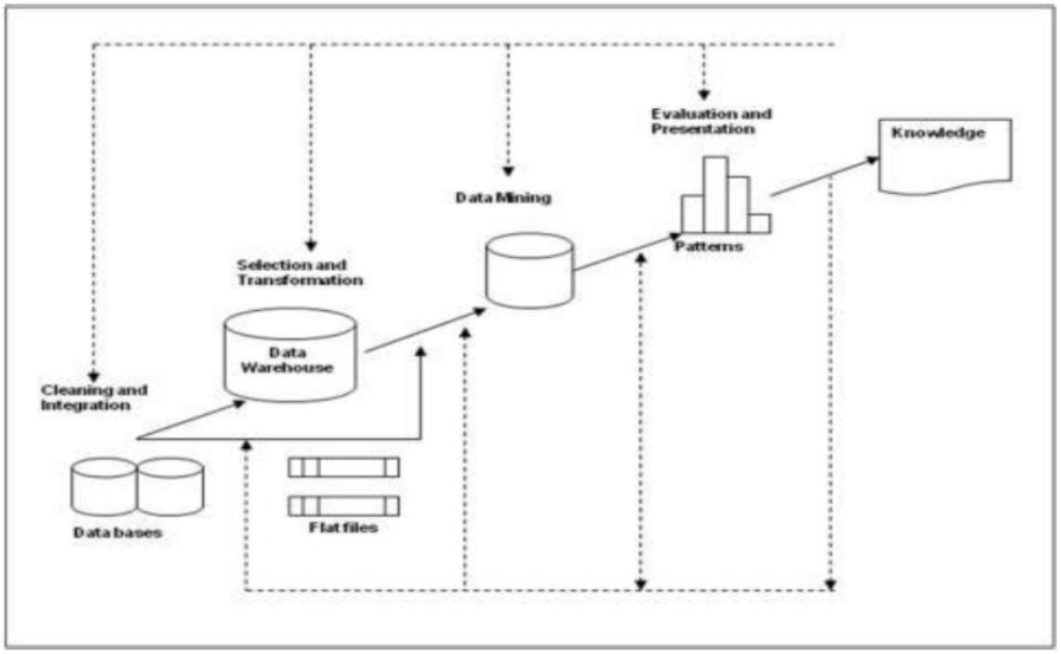
2.2.3.1 Pengertian Data Mining

Data mining sering juga disebut *knowledge discovery in database* (KDD), adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam *set data* berukuran besar. Keluaran dari *data mining* ini bisa dipakai untuk memperbaiki pengambilan keputusan di masa depan. Sehingga istilah *pattern recognition* sekarang jarang digunakan karena ia temasuk bagian dari *data mining* [5].

2.2.3.2 Konsep Data Mining

Data mining sangat diperlukan terutama dalam mengelola data yang sangat besar untuk memudahkan aktifitas *recording* suatu transaksi data untuk proses *data warehousing* agar dapat memberikan informasi yang akurat bagi pengguna *data mining*. Alasan utama *data mining* sangat dibutuhkan dalam industri informasi karena tersedianya data dalam jumlah yang besar dan semakin besarnya kebutuhan untuk mengubah data tersebut menjadi informasi dan pengetahuan yang berguna karena sesuai fokus bidang ilmu ini yaitu melakukan kegiatan mengekstraksi atau menambang pengetahuan dari data yang berukuran atau berjumlah besar. Informasi

inilah yang nantinya sangat berguna untuk pengembangan. Berikut adalah langkah-langkah dalam *data mining* [5], bisa dilihat pada gambar 2.3:



Gambar 2. 3 Konsep Data Mining

1. *Data cleaning* yaitu untuk menghilangkan *noise* data yang tidak konsisten.
2. *Data integration* yaitu menggabungkan beberapa *file* atau *database*.
3. *Data selection* yaitu data yang relevan dengan tugas analisis dikembalikan ke dalam *database* untuk proses *data mining*.
4. *Data transformation* yaitu data data berubah atau bersatu menjadi bentuk yang tepat untuk menambang dengan ringkasan perfoma atau operasi agresi.
5. *Data mining* yaitu proses esensial dimana metode yang intelejen digunakan untuk mengekstrak pola data.
6. *Knowledge Discovery* yaitu proses essential dimana metode yang intelejen digunakan untuk mengekstrak pola data.
7. *Pattern evolution* yaitu untuk mengindentifikasi pola yang benar-benar menark yang mewakili pengetahuan berdasarkan atas beberapa tindakan yang menarik.
8. *Knowledge presentation* yaitu gambaran teknik visualisasi dan pengetahuan digunakan untuk memberikan pengetahuan yang telah ditambah kepada *user*.

2.2.4 Association Rule

Aturan asosiasi (*Association rules*) atau analisis afinitas (*affinity analysis*) berkenaan dengan studi tentang 'apa bersama apa'. Ini bisa berupa studi transaksi di *supermarket*, misalnya seseorrang yang membeli susu bayi juga membeli sabun mandi. Di sini berarti susu bayi bersama dengan sabun mandi. Karena awalnya berasal dari studi tentang *database* transaksi pelanggan untuk menentukan kebiasaan suatu produk dibeli bersama produk apa, maka aturan asosiasi juga sering dinamakan *market basket analysis* [5].

Association Rule adalah bentuk jika "kejadian sebelumnya" kemudian "konseuensinya" (*if antecedent, then consequent),* yang diikuti dengan perhitungan aturan *support* dan *confident.* Bentuk umum dari *association rule* adalah *Antecedent -> Consequent.* Bila kita ambil contoh dalam sebuah transaksi pembelian barang di sebuah *minimarket* didapat bentuk *association rule* roti -> selai. Yang artinya bahwa pelanggan yang membeli roti ada kemungkinan pelanggan tersebut juga membeli selai, dimana tidak ada batasan dalam jumlah *item-item* pada bagian *antecedent* atau *consequent* dalam sebuah *rule.* Dalam menentukan suatu *association rule,* terdapat suatu *interestingness measure* (ukuran kepercayaan) yang didapat dari hasil pengolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran yaitu:

1. *Support*: suatu ukuran yang menunjukan seberapa besar tingkat dominasi suatu *item/itemset* dari keseluruhan transaksi. *Support* merupakan matrik pertama yang ditetapkan dalam analisis keranjang pasar, yang merupakan probabilitas dari asosiasi (probabilitas dari dua item yang diberi bersama-sama). *Support* dihasilkan dari beberapa kali jumlah *item* A dan B terjadi bersamaan dalam transaksi yang sama dibagi dengan jumlah total dari transaksi terebut. *Support* dapat dirumuskan sebagai berikut [6]:

$$Support = P(A \cap B)$$
$$= \frac{\text{Jumlah transaksi yang memuat A dan B}}{\text{Total jumlah transaksi}}$$

2. *Confident: Confident* dihasilkan dari seberapa kuat hubungan produk yang sudah dibeli. *Confident* dapat dirumuskan sebagai berikut [6]:

Confidence = P (B / A)

$$= \frac{Support\ (A \cap B\ )}{P\ (A)}$$

Kedua ukuran ini nantinya berguna dalam menentukan kekuatan suatu pola dengan membandingkan pola tersebut dengan nilai minimum kedua parameter tersebur yang ditentukan oleh pengguna. Bila suatu pola memenuhi kedua nilai minimum parameter yang sudah ditentukan sebelumnya, maka pola tersebut dapat disebut sebagai *interesting rule* atau *strong rule*. Metodologi dasar analisis asosiasi terbagi menjadi dua tahap yaitu [6]:

1. Analisis pola frekuensi tinggi

   Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dan *database*. Nilai *support* sebuah *item* diperoleh dengan rumus berikut:

   $$Support(A) = \frac{Jumlah\ transaksi\ \text{yang}\ mengandung\ A}{Jumlah\ total\ transaksi} \quad \text{X 100 \%}$$

   Persamaan (2-1)

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

   $support(A, B) = P(A \cap B)$…Persamaan (2-2)

   $$Support(A, B) = \frac{Jumlah\ transaksi\ \text{yang}\ mengandung\ A\ dan\ B}{Jumlah\ total\ transaksi}\text{x100\%.}$$

   Persamaan(2-2)

2. Pembentukan aturan asosiasi

   Setelah semua pola frekuensi tinggi ditemukan, kemudian mencari aturan asosiasi yang cukup kuat ketergantungan antar *item*. Dalam *antecedent* (pendahulu) dan *consequent* (pengikut) serta memenuhi syarat minimum untuk *confident* aturan asosiatif A -> B. Misalkan D adalah himpunan transaksi, dimana setiap transaksi T dalam D mempresentasikan himpunan *item* yang berada dalam I. I adalah himpunan *item* yang dijual.

Misalkan kita memilih himpunan *item* A dan himpunan *item* B, kemudian aturan asosiatif akan berbentuk:

Jika A, maka B (A->B)

Dimana *antecedent* A dan *consequent* B merupakan subset dari I, dan A dan B dimana aturan:

Jika A, maka B

Tidak berarti

Jika B, maka A

Sebuah *itemset* adalah himpunan *item-item* yang ada dalam I, dan *itemset. Frekuensi itemset* merupakan *itemset* yang memiliki frekuensi kemunculan lebih dari nilai minimum yang telah ditentukan.

Nilai *confident* dari aturan A->B diperoleh dari rumus berikut.

$$Confidence = P(B \mid A) = \frac{Jumlah\ transaksi \text{ yang } mengandung\ A\ dan\ B}{Jumlah\ transaksi\ yang\ mengandung\ A}$$

Persamaan (2-3)

2.2.5 Algoritma Fp-Growth

Algoritma yang sama dengan apriori, *Fp-Growth* mulai dengan menghitung *item* tunggal sesuai dengan jumlah kemunculan *item* yang ada di dalam *dataset.* Setelah proses perhitungan selesai maka akan dibuat struktur pohon pada tahap kedua. Pohon yang dibuat mulanya kosong yang nantinya akan diisi dengan hasil dari *dataset* yang telah didapat sebelumnya. Kunci untuk mendapatkan struktur pohon yang bisa didapatkan dengan proses lebih cepat untuk mencari *item set* yang besar menjadi sedikit dengan diurutkan secara *descending* dari frekuensi yang ada di dalam *dataset* tersebut [7].

Fp-Tree merupakan struktur penyimpanan data yang dimampatkan. *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 pemindaian data transaksi yang terbukti sangat efisien.

Adapun *Fp-Tree* adalah sebuah pohon dengan definisi sebagai berikut:

a. Fp-Tree dibentuk oleh sebuah akar yang diberi label *null,* sekumpulan berupa pohon yang beranggotakan *item-item* tertentu dan sebuah tabel *frequent header*.
b. 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 melalui simpul tersebut, dan *pointer* penghubung yang menghubungkan simpul-simpul dengan label atau *item* sama antar lintasan, ditandai dengan garis panah putus-putus.

2.2.5.1 Langkah-Langkah Proses Perhitungan Association Rule Dengan Algoritma FP-Growth

Proses perhitungan *association rule* terdiri dari beberapa tahap adalah sebagai berikut [8]:

1. Membuat *Header Item*

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

   FP-Tree dibangun dengan mencari *item* sesuai urutan pada *item* yang *frequent*. Data transaksi tidak perlu diurutkan, dan untuk tiap *item* yang ditemukan bisa langsung dimasukkan ke dalam FP-Tree. Sesudah membuat *root*, tiap *item* yang ditemukan dimasukkan berdasarkan *path* pada FP-Tree. Jika *item* yang ditemukan sudah ada, maka nilai *support item* tersebut yang ditambahkan. Namun jika *path* belum ada, maka dibuat *node* baru untuk

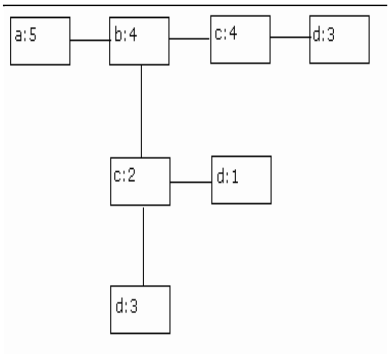
melengkapi *path* baru pada FP-Tree tersebut. Hal ini dilakukan selama item pada transaksi masih ada yang *qualified*, artinya memenuhi nilai minimum *support*. Jadi, *item-item* yang ditemukan dalam transaksi akan berurutan memanjang ke bawah. Dalam struktur FP-Tree, diterapkan alur *path* dari *child* hingga ke *root*. Jadi, suatu *path* utuh dalam FP-Tree adalah dari *child* terbawah hingga ke *root*. Tiap *node* pada FP-Tree memiliki *pointer* ke *parent*, sehingga pencarian harus dimulai dari bawah.

3. *Pattern Extraction*

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

4. Memasukkan setiap pattern yang ditemukan dalam *PatternTree*

   Setelah mengolah FP-Tree menjadi *pattern-pattern*, diperlukan proses akumulasi *pattern-pattern* yang ditemukan mengingat *pattern* yang sama dapat ditemukan pada path yang berbeda. Untuk itu digunakan struktur data *Pattern Tree* (lihat Gambar 2.4). Setiap node di *Pattern Tree* merepresentasikan dan menyimpan frekuensi suatu *pattern*. *Pattern Tree* terdiri atas *Pattern TreeNode* yang menyimpan nilai item, nilai *support* dan dilengkapi dengan dua *pointer* yaitu untuk horizontal dan vertikal.



Gambar 2. 4 Pattern Tree

Misalnya pada *node* d:1 di atas, berarti terdapat *pattern* a-c-d bernilai *support* 1. Kemudian bila ada pattern a-c-d lagi bernilai *support* n yang ditemukan dari FP-Tree maka nilai *support* 1 tersebut menjadi n+1.

Contoh hasil lengkap dari *PatternTree* tersebut:

1. a:5 menggambarkan bahwa ada pattern a sebanyak 5
2. b:4 menggambarkan bahwa ada pattern a-b sebanyak 4
3. c:4 menggambarkan bahwa ada pattern a-b-c sebanyak 4
4. d:3 menggambarkan bahwa ada pattern a-b-c-d sebanyak 3
5. c:2 menggambarkan bahwa ada pattern a-c sebanyak 2
6. d:1 menggambarkan bahwa ada pattern a-c-d sebanyak 1
7. d:3 menggambarkan bahwa ada pattern a-d sebanyak 3

5. Mengurutkan dan menyeleksi *Pattern.*

Pattern yang tidak memenuhi minimum *support*, dihapus dari daftar *pattern*. *Pattern-pattern* yang tersisa kemudian diurutkan untuk memudahkan pembuatan *rules*.