

BAB II

TINJAUAN PUSTAKA

2.1 Decision Support System

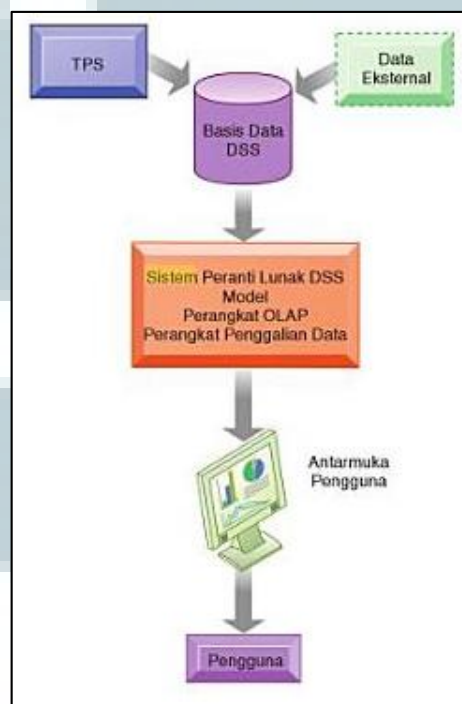
Turban mendefinisikan *Decision Support System* sebagai sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu para pengambilan keputusan untuk mengambil keputusan (Turban, 2005).

Menurut Laudon *Decision Support System* di masa paling awalnya sangatlah digerakkan oleh model, menggunakan beberapa jenis model untuk menunjukkan analisis "bagaimana jika" dan analisis lainnya. Kemampuan analisis DSS didasarkan pada teori atau model yang kuat, digabungkan dengan suatu antarmuka pengguna yang baik yang membuat sistem tersebut mudah digunakan. Beberapa DSS yang kontemporer sifatnya digerakkan oleh data, menggunakan pemrosesan analitis *online* (OLAP), dan penggalian data untuk menganalisis data berukuran besar (Laudon, 2007).

Laudon menjelaskan komponen-komponen suatu DSS melalui gambar 2.1. Di antaranya adalah basis data untuk analisis dan *query*; sistem piranti lunak dengan banyak model, penggalian data dan perangkat analitis lainnya; serta antarmuka pengguna (Laudon, 2007).

Basis data DSS (*DSS Database*) adalah sekumpulan data yang sekarang dan historis dari sejumlah aplikasi atau kelompok. Basis data DSS dapat berupa basis data kecil dalam PC yang berisi sebagian data perusahaan yang telah diunduh dan mungkin digabungkan dengan data eksternal. Alternatif lainnya, basis data DSS

dapat berupa gudang data besar yang secara terus-menerus diperbarui oleh TPS (*Transaction Processing Systems*) perusahaan pusat. Data dalam basis data DSS secara umum mengekstrak atau menyalin basis data produksi sehingga dengan menggunakan DSS tidak akan mengganggu sistem-sistem operasional yang penting. Antarmuka pengguna DSS memberikan kemudahan berinteraksi antara pengguna sistem dan peranti lunak DSS.



Gambar 2.1 Gambaran Umum Sistem Pendukung Pengambil Keputusan (Laudon, 2007).

Sistem peranti lunak DSS (*DSS software system*) berisi peranti lunak yang digunakan untuk menganalisis data. Ini dapat berisi berbagai perangkat OLAP, perangkat penggalian data, atau sekumpulan model matematis dan analitis yang dapat dengan mudah diakses oleh pengguna DSS. Model adalah representasi abstrak yang mengilustrasikan beberapa komponen atau hubungan dari suatu

fenomena. Model dapat berupa model fisik (seperti model pesawat terbang), model matematis (seperti persamaan), atau model verbal (seperti penjelasan mengenai prosedur untuk melakukan pemesanan) (Laudon, 2007).

Menurut Turban, *Decision Support System (DSS)* atau Sistem Pendukung Keputusan adalah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi-terstruktur (Turban, 2005). Laudon menyatakan, DSS juga mendukung analisis masalah semi-terstruktur dan tidak terstruktur (Laudon, 2007).

Menurut Schell dan McLeod, masalah semi-terstruktur adalah masalah yang terdiri atas beberapa elemen atau hubungan yang dipahami oleh si pemecah masalah dan beberapa yang tidak dapat dipahami. Salah satu contoh adalah pemilihan lokasi untuk membangun sebuah pabrik baru. beberapa elemen, seperti harga tanah, pajak, dan biaya-biaya untuk mengirimkan bahan baku, dapat diukur dengan tingkat ketepatan yang tinggi. Tetapi elemen-elemen lain, seperti bahaya dari lingkungan dan perilaku masyarakat sekitar, sulit untuk diidentifikasi dan diukur. Masalah yang tidak terstruktur adalah masalah yang tidak memiliki elemen atau hubungan antar elemen yang dipahami oleh orang yang memecahkan masalah. salah satu contoh dari masalah yang tidak terstruktur adalah memutuskan film yang mana yang paling kita sukai. Manajer bisnis sering kali tidak memiliki perangkat yang cukup untuk mendefinisikan masalah seperti ini dengan cara yang terstruktur (Junior dan Schell, 2007).

Menurut Wibisono, *Decision Support System (DSS)* atau Sistem Pendukung Keputusan adalah sistem berbasis komputer yang membantu para pengambil

keputusan mengatasi berbagai masalah melalui interaksi langsung dengan sejumlah database dan perangkat lunak analitik. Tujuan dari sistem adalah untuk menyimpan data dan mengubahnya ke informasi yang terorganisir yang dapat diakses dengan mudah, sehingga keputusan-keputusan yang diambil dapat dilakukan dengan cepat dan akurat (Wibisono, 2003).

Turban menambahkan karakteristik dan kapabilitas yang dimiliki oleh DSS yaitu sebagai berikut (Efraim Turban, 2005).

1. Membantu mengambil keputusan, baik dalam kondisi semi-terstruktur maupun tidak terstruktur.
2. Mendukung manajer di semua lapisan.
3. Mendukung penggunaan baik secara individu maupun kelompok.
4. Menyediakan keputusan yang bersifat interdependent dan sequential.
5. Mendukung berbagai proses pengambilan keputusan.
6. Fleksibel dan mudah diadaptasikan.
7. Mudah dan interaktif dalam penggunaannya.
8. Meningkatkan keefektifan dalam pengambilan keputusan (*accuracy, quality, timeline*).
9. Pengambilan keputusan memiliki kendali sepenuhnya dalam mengambil keputusan. DSS hanya memberikan saran, bukan keputusan mutlak.
10. Pengguna DSS dapat mengubah dan membangun sistem sederhana sesuai dengan keinginannya.
11. DSS secara umum digunakan sebagai model untuk menganalisa keputusan yang akan diambil.

12. DSS dapat menyediakan akses ke berbagai sumber data, format data, dan tipe data.
13. DSS dapat dijalankan sebagai alat yang berdiri sendiri jika digunakan oleh individu dan dapat diintegrasikan dengan DSS lainnya dan dapat didistribusikan melalui web.

2.2 Algoritma Apriori

Menurut Erwin, algoritma Apriori adalah salah satu algoritma yang melakukan pencarian *frequent itemset* atau jumlah kombinasi *itemset* dengan menggunakan teknik *association rule* (Erwin, 2009).

Algoritma *Apriori* menggunakan pengetahuan frekuensi atribut yang telah diketahui sebelumnya untuk memproses informasi selanjutnya. Pada algoritma *Apriori* menentukan kandidat yang mungkin muncul dengan cara memperhatikan dua parameter yaitu *minimum support* dan *minimum confidence*.

Menurut Kusriani, *support* adalah suatu ukuran yang menunjukkan seberapa besar tingkat dominasi atau persentase suatu kombinasi *item* dari keseluruhan transaksi (Kusriani, 2009). Menurut Erwin, *support* adalah nilai pengunjung atau persentase kombinasi sebuah *item* dalam *database*. *Confidence* adalah suatu ukuran atau nilai kepastian untuk menunjukkan kuatnya hubungan antar *item* dalam sebuah *Apriori*, contohnya seberapa sering *item* B dibeli jika orang membeli *item* A (Erwin, 2009).

Kedua ukuran ini nantinya berguna dalam menentukan kekuatan suatu pola dengan membandingkan pola tersebut dengan nilai minimum kedua parameter yang

ditentukan oleh pengguna yaitu *minimum support* dan *minimum confidence*. Bila suatu pola memenuhi kedua nilai minimum parameter yang sudah ditentukan maka sebuah pola dapat disebut *interesting rule* atau *strong rule*. *Association Rule* dalam algoritma *Apriori* biasanya dinyatakan dalam bentuk :

Tabel 2.1 *Association Rule*

Product	Support	Confidence
{Diaper, Milk} → Baby Podwder	40%	50%

Maksud dari bentuk diatas adalah 50% dari transaksi di *database* yang memuat *item* diaper dan milk juga memuat *item* baby powder. Sedangkan 40% dari seluruh transaksi yang ada di *database* memuat ketiga *item* tersebut. Dapat juga diartikan : Seorang konsumen yang membeli diaper dan milk punya kemungkinan 50% untuk juga membeli baby powder. Aturan tersebut cukup akurat karena mewakili 40% dari catatan transaksi yang ada (Kusrini, 2009).

2.2.1. Metodologi Dasar Asosiasi pada Algoritma Apriori

Metodologi dasar Asosiasi pada algoritma *Apriori* terbagi menjadi dua tahap yaitu sebagai berikut (Kusrini, 2009).

1. Analisa Pola Frekuensi Tinggi

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

$$Support(A) = \frac{\sum \text{transaksi berisi } A}{\sum \text{transaksi}} \dots\dots\dots \text{Rumus 2.1}$$

Pada rumus 2.1 menjelaskan bahwa nilai *support* diperoleh dengan cara mencari jumlah transaksi yang mengandung nilai A (satu *item*) dibagi dengan jumlah keseluruhan transaksi.

2. Pembentukan Aturan Asosiatif

Setelah semua pola frekuensi tinggi ditemukan, kemudian mencari aturan asosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiatif $A \rightarrow B$ dari *support* pola frekuensi tinggi A dan B, menggunakan rumus berikut :

$$Confidence = p(A|B) = \frac{\sum \text{transaksi berisi } A \text{ dan } B}{\sum \text{transaksi yang berisi } A} \dots\dots\dots \text{Rumus 2.2}$$

Pada rumus 2.2 menjelaskan bahwa nilai *confidence* diperoleh dengan cara mencari jumlah transaksi yang mengandung nilai A dan B (*item* pertama bersamaan dengan *item* yang lain) dibagi dengan jumlah transaksi yang mengandung A (*item* pertama).

Proses utama yang dilakukan dalam algoritma *Apriori* untuk mendapat *frequent itemset* yaitu sebagai berikut (Erwin, 2009).

1. *Join* (penggabungan)

Proses ini dilakukan dengan cara pengkombinasian *item* dengan *item* yang lainnya hingga tidak dapat terbentuk kombinasi lagi.

2. *Prune* (pemangkasan)

Proses pemangkasan yaitu hasil dari *item* yang telah dikombinasikan kemudian dipangkas dengan menggunakan *minimum support* yang telah ditentukan oleh *user*.

Prinsip dari Algoritma *Apriori* adalah sebagai berikut.

1. Mengumpulkan *item* yang tunggal kemudian mencari *item* yang terbesar. *Item* tunggal ini biasa juga disebut dengan *1-itemset* karena hanya terdiri dari *item* tunggal.
2. Dapatkan *candidate pairs* kemudian hitung *large pairs* dari masing-masing *item*. Yang dimaksud dengan *candidate pairs* adalah dua kombinasi *item* barang yang ada dan *large pairs* adalah jumlah kemunculan dua kombinasi *item* tersebut. Dua kombinasi *item* barang ini dapat ditulis juga dengan *2-itemset* karena terdiri dari dua jenis barang yang dikombinasikan menjadi pasangan.
3. Temukan *candidate triplets* dari setiap *item*. Yang dimaksud dengan *candidate triplets* adalah tiga kombinasi *item* barang yang ada. *Candidate triplets* dapat dicari menggunakan *pairs* sebelumnya yang sudah lolos

minimum support. Tiga kombinasi *item* barang ini dapat ditulis juga dengan *3-itemset* karena terdiri dari tiga jenis barang yang dikombinasikan menjadi pasangan.

4. Setiap *subset* dari sebuah *frequent itemset* harus menjadi *frequent*. Yang artinya adalah setiap anggota dalam *frequent itemset* yang terpilih karena memenuhi *minimum support* dan *confidence* akan dijadikan sebuah *frequent* baru yang memiliki *strong rule*.

Berikut adalah contoh gambaran cara kerja algoritma *Apriori* dengan ditentukan *minimum support* = 40% dan *minimum confidence* = 70% (Bahadi, 2011).

Tabel 2.2 Daftar Transaksi (Bahadi, 2011)

ID Transaksi	Daftar Belanja
1	Beer, diaper, baby powder, bread, umbrella
2	Diaper, baby powder
3	Beer, diaper, milk
4	Diaper, beer, detergent
5	Beer, milk, coca cola

Dari tabel 2.1 menunjukkan daftar transaksi yang setiap transaksi memiliki daftar barang yang sudah dibeli. Setiap barang yang ada akan dibatasi dengan *minimum support* = 40% (2/5).

Tabel 2.3 Tabel C1 (Bahadi, 2011)

Barang	Support
Beer	4/5
Diaper	4/5
Baby powder	2/5
Bread	1/5
Umbrella	1/5
Milk	2/5
Detergent	1/5
Coca-cola	1/5

Tabel 2.2 menunjukkan *item* tunggal atau 1-*itemset* dari setiap barang yang berada dalam daftar transaksi. Barang tersebut dicari nilai *support*-nya dengan menggunakan rumus 2.1. Sebagai contoh Beer memiliki *support* bernilai 4/5, nilai ini dihasilkan dengan melihat daftar transaksi. Dari 5 (lima) transaksi yang dilakukan, Beer dibeli sebanyak 4 (empat) kali atau Beer masuk ke dalam 4 (empat) transaksi dari 5 (lima) transaksi.

Sebelumnya setiap barang sudah dibatasi dengan *minimum support* = 40% (2/5). Dengan melihat nilai *support* yang ada pada tabel 2.2 maka dapat dilihat barang yang tidak memenuhi *minimum support*. Untuk barang yang tidak memenuhi *minimum support* akan dihilangkan dan tidak dimasukkan ke tabel berikutnya yaitu L1.

Tabel 2.4 Tabel L1 (Bahadi, 2011)

Barang	Support
Beer	4/5
Diaper	4/5
Baby powder	2/5
Milk	2/5

Barang yang berada pada tabel C1 (tabel 2.2) yang memiliki *support* kurang dari 40% dihilangkan sedangkan barang yang memiliki *support* lebih dari 40% akan dimasukkan ke dalam tabel L1 (tabel 2.3).

Setelah menemukan barang-barang yang memenuhi *minimum support*, selanjutnya barang-barang tersebut akan dikombinasikan dan dimasukkan ke dalam tabel C2 (tabel 2.4) atau mencari *candidate pairs* atau 2-itemset. Barang-barang ini dikombinasikan sesuai dengan daftar transaksi yang ada.

Tabel 2.5 Tabel C2 (Bahadi, 2011)

Barang	Support
Beer, diaper	3/5
Beer, baby powder	1/5
Beer, milk	2/5
Diaper, baby powder	2/5
Diaper, milk	1/5
Baby powder, milk	0

Barang pada tabel C2 (tabel 2.4) dihapus dari tabel jika memiliki *support* kurang dari 40%. Jika barang memiliki *support* lebih dari 40% akan dimasukkan ke tabel L2 (tabel 2.5).

Tabel 2.6 Tabel L2 (Bahadi, 2011)

Barang	Support
Beer, diaper	3/5
Beer, milk	2/5
Diaper, baby powder	2/5

Barang yang ada pada tabel 2.5 kemudian akan dikombinasikan dan dimasukkan ke dalam tabel C3 (tabel 2.6) untuk mencari *candidate triplets*.

Tabel 2.7 Tabel C3 (Bahadi, 2011)

Barang	Support
Beer, diaper, milk	1/5
Beer, diaper, baby powder	1/5
Beer, milk, baby powder	0/5
Milk, diaper, baby powder	0/5

Kombinasi barang yang ada pada tabel C3 tidak memenuhi *minimum support*, sehingga tabel L2 yang akan diberikan *minimum confidence* = 70%. Barang yang tidak memenuhi *minimum confidence* akan dihilangkan dan yang memenuhi *minimum confidence* akan masuk ke daftar *strong rule* atau hasil akhir kombinasi barang dari transaksi.

Tabel 2.8 L2 dengan nilai Confidence (Bahadi, 2011)

Barang	Support(A,B)	Confidence
Beer, diaper	60%	75%
Beer, milk	40%	50%
Diaper, baby powder	40%	50%
Diaper, beer	60%	75%
Milk, beer	40%	100%
Baby powder, diaper	40%	100%

Pada tabel L2 seharusnya hanya memiliki 3 (tiga) kombinasi barang. Akan tetapi, dari 3 (tiga) kombinasi tersebut kemudian ditambahkan dengan cara membalik posisi barang yang tadinya Beer & Diaper menjadi Diaper & Beer. Hal ini dilakukan untuk mendapatkan nilai propabilitas yang disesuaikan dengan rumus

2.2. Tabel 2.9 berisi *strong rules*, yaitu kombinasi barang yang memenuhi kriteria *minimum confidence*.

Tabel 2.9 Daftar Strong Rules (Bahadi, 2011)

<i>Strong Rules</i>	<i>Support</i>	<i>Confidence</i>
Beer → diaper	60%	75%
Diaper → beer	60%	75%
Milk → beer	40%	100%
Baby powder → diaper	40%	100%

2.3 Market Basket Analysis

Menurut Budhi dan Seodjianto, *Market Basket Analysis* merupakan sebuah analisis terhadap kebiasaan belanja konsumen, dengan cara menemukan asosiasi di antara berbagai macam *item* yang dimasukkan konsumen di dalam keranjang belanja mereka (Budhi dan Soedjianto, 2007:121). Analisis ini sering diterapkan pada swalayan atau supermarket. Dalam keranjang belanja tersebut menginformasikan tentang *item-item* apa saja yang dibeli konsumen secara bersamaan yang akhirnya menuju kasir untuk membayarnya dan tersimpan dalam sistem komputerisasi yang terdapat di swalayan, secara sekilas tidak ada hal istimewa yang terjadi pada transaksi tersebut. Namun dari sisi swalayan tersebut

sebenarnya telah memperoleh sebuah pengetahuan yang jarang diketahui sebelumnya yaitu tentang pola pembelian item yang dibeli oleh konsumen, yang dapat dilihat dari serangkaian *item-item* yang dibeli oleh konsumen.

Untuk beberapa kasus, pola dari *item-item* yang dibeli secara bersamaan oleh konsumen mudah untuk ditebak, misalnya susu dibeli bersamaan dengan roti. Namun, mungkin saja terdapat suatu pola pembelian *item* yang tidak pernah terpikirkan sebelumnya. Misalnya, pembelian minyak goreng dengan deterjen. Mungkin saja pola seperti ini tidak pernah terpikirkan sebelumnya karena minyak goreng dan deterjen tidak mempunyai hubungan, baik sebagai barang pelengkap maupun barang pengganti. Hal ini mungkin tidak pernah terpikirkan sebelumnya sehingga tidak dapat diantisipasi jika terjadi sesuatu, seperti kekurangan stok deterjen. Inilah salah satu manfaat yang dapat diperoleh dari *Market Basket Analysis*. Dengan kata lain tujuan *Market Basket Analysis* adalah untuk merancang strategi penjualan atau pemasaran dengan memanfaatkan data transaksi penjualan yang telah tersedia di perusahaan, diantaranya :

- Dengan mengubah *layout* toko, mendekatkan kategori-kategori *item* yang sering dibeli bersamaan pada satu waktu.
- Memberikan diskon pada pembelian paket *item* yang jarang dibeli.

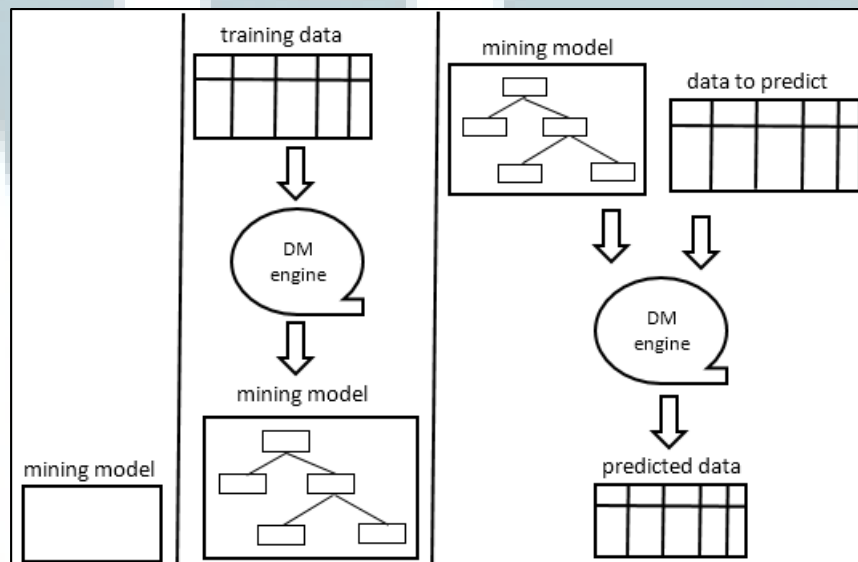
2.4 Data Mining

Menurut Davies, *data mining* adalah penambangan atau penemuan informasi baru dengan mencari pola atau aturan tertentu dari sejumlah data yang sangat besar (Davies, 2004). Santoso menambahkan, *data mining* sering juga disebut sebagai *knowledge discovery in database* (KDD). KDD adalah kegiatan yang meliputi pengumpulan, pemakaian data, historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar (Santoso, 2007).

Menurut Han, *Data mining* adalah kegiatan menemukan pola yang menarik dari data dalam jumlah besar, data dapat disimpan dalam *database*, *data warehouse*, atau penyimpanan informasi lainnya. *Data mining* berkaitan dengan bidang ilmu – ilmu lain, seperti *database system*, *data warehousing*, statistik, *machine learning*, *information retrieval*, komputasi tingkat tinggi, dan *decision support system*. Selain itu, *data mining* didukung oleh ilmu lain seperti *neural network*, pengenalan pola, *spatial data analysis*, *image database*, *signal processing* (Han, 2006). Witten mengatakan *data mining* didefinisikan sebagai proses menemukan pola-pola dalam data. Proses ini otomatis atau seringkali semiotomatis. Pola yang ditemukan harus penuh arti dan pola tersebut memberikan keuntungan, biasanya keuntungan secara ekonomi. Data yang dibutuhkan dalam jumlah besar (Witten, 2005).

2.3.1. Proses Data Mining

Jamie MacLennan, ZhaoHui T., dan Bogdan C. menjelaskan alasan dasar dari *data mining* atau *machine learning* adalah untuk menunjukkan beberapa contoh algoritma, dan dari contoh tersebut, sebuah algoritma dapat mengekstrak pola yang kemudian bisa digunakan untuk inspeksi atau untuk menyimpulkan informasi tentang contoh-contoh baru. Trik untuk *data mining* adalah membentuk contoh-contoh tersebut sedemikian rupa sehingga pola yang diekstrak berguna, informatif, dan akurat.



Gambar 2.2 Proses *Data Mining* (*The Data Mining Process*, 2009)

Gambar 2.2 menunjukkan proses *data mining* secara singkat. Pertama, Anda harus mendefinisikan masalah dan merumuskan suatu objek, yang disebut *mining model*. *Model mining* ini menjelaskan seperti apa contoh data yang ada dan bagaimana algoritma *data mining* harus menginterpretasikan data. Kedua, Anda harus memberikan contoh data untuk algoritma.

Algoritma ini menggunakan definisi masalah Anda (*mining model*) sebagai semacam lensa yang akan digunakan untuk memeriksa data dan mengekstrak pola. Ini disebut *training model*. Bila sudah selesai, tergantung pada apa yang Anda coba lakukan, Anda bisa berhenti dan menganalisa pola yang ditemukan oleh algoritma dalam contoh Anda.

Akhirnya, Anda dapat memberikan data baru, dirumuskan dalam cara yang sama seperti contoh pelatihan, untuk melakukan *predictions* atau *deductions* informasi tentang contoh menggunakan pola yang ditemukan oleh algoritma (Jamie M., ZhaoHui T., Bogdan C., 2009).

UMN