**Review Data Mining  
Arianto Wibowo – 5113100037**

**Tipe Data :**

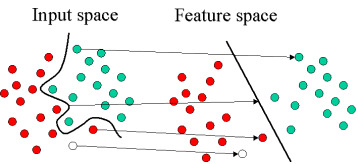
1. **Nominal / Kategorikal** : Data yang membedakan dalam ukuran kualitatif. Data ini tidak memiliki tingkatan (semua data dianggap setara). Contoh : data warna (merah,hijau,kuning). Tidak ada data yang tingkatnya lebih tinggi ataupun lebih rendah.
2. **Ordinal** : Data yang memiliki tingkatan. Terdapat klasifikasi berdasarkan tingkatannya. Contoh : data ukuran baju (S, M, L, XL, XXL). Disini, tingkatan XL lebih tinggi tingkatnya daripada L, M lebih tinggi tingkatnya dari S, dll.
3. **Scale** : Data yang memiliki tingkatan dan terdapat klasifikasi berdasarkan tingkatannya. Yang membedakannya dengan data ordinal adalah adanya ukuran yang tetap dan skala yang tetap. Contoh : berat badan (50 kg, 70 kg, 75 kg). Disini, jelas dinyatakan bahwa beda berat badan 1 dan 2 adalah 20 kg, dan berat badan 2 dengan 3 adalah 5 kg. Jika dalam data ordinal, tidak diketahui dengan jelas berapa perbedaan dari ukuran baju S dengan M.
4. **Kontinu** : Dalam data numerik, terdapat data kontinu. Data ini merupakan data kuantitatif yang nilainya dapat menempati nilai semua interval pengukuran. Data kontinu dapat berupa pecahan / bilangan real. Contoh, tinggi badan seseorang (162.5 cm, 170.2 cm)
5. **Diskrit** : Dalam data numeri, terdapat data diskrit. Data ini sifatnya terputus-putus karena tidak memiliki nilai pecahan. Oleh karena itu, scope dari data ini adalah bilangan bulat / integer. Contoh, banyaknya mobil di parkiran (10 mobil, 15 mobil, 23 mobil). Tidak mungkin jumlah mobil di parkiran dapat dinyatakan dengan bilangan pecahan.

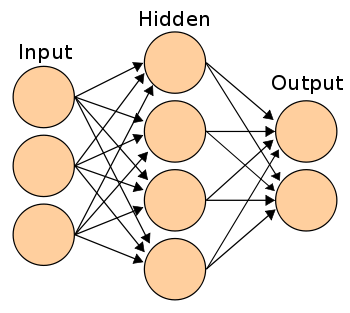
**Preprocessing :**

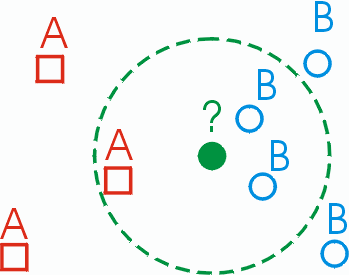
Data preprocessing dilakukan terlebih dahulu sebelum melakukan proses data mining selanjutnya. Hal ini digunakan untuk memperbaiki akurasi pada tahap selanjutnya. Berikut adalah beberapa macam data preprocessing yang dapat dilakukan terlebih dahulu :

1. **Sampling :** Jika data yang ada terlalu besar dan terlalu sulit untuk diproses secara keseluruhan, maka dapat dilakukan sampling terlebih dahulu. Tujuan hal ini adalah untuk memperkecil kompleksitas program sehingga tidak memakan memori dan waktu yang terlalu besar untuk melakukan tahap selanjutnya.
2. **Data Cleaning :** Dapat digunakan untuk menghilangkan data-data yang nilainya salah, memperbaiki kekacauan data dan memeriksa data yang tidak konsisten (noise / outlier).
3. **Normalisasi :** Jika data memiliki satuan ukuran yang beragam, maka dapat dilakukan normalisasi data terlebih dahulu. Tujuan normalisasi adalah agar seluruh data memiliki satuan yang homogen. Sebagai contoh, jika salah satu atribut bernilai 1-10, maka jika atribut lain ada yang memiliki range 1-1.000.000.000 maka dapat dilakukan normalisasi terlebih dahulu agar rangenya homogen.
4. **Reduksi :** Mungkin tidak seluruh atribut data dalam dataset perlu diproses. Dapat dilakukan reduksi terhadap data ini agar dapat mengurangi kompleksitas dan mendapat akurasi yang lebih baik. Dapat dilakukan menggunakan seleksi fitur (memilih fitur tertentu yang akan diproses) ataupun reduksi dimensi (membuat atribut baru yang merupakan campuran dari atribut sebelumnya yang sudah ada).
5. **Missing Value :** Terkadang data yang didapat memiliki missing value (yaitu sel yang tidak memiliki nilai). Maka sebelum diproses dapat dilakukan upaya untuk menghapus missing value ini. Bisa dengan menghapus record tersebut atau mengisi dengan ukuran pemusatan tertentu (mean, median, atau modus).
6. **Integrasi :** Data dapat diperoleh dari beberapa sumber, database, atau beberapa file. Untuk mendapat data yang berkualitas, data tersebut harus diintegrasi agar menjadi koheren. Dapat juga digunakan untuk menyelesaikan masalah konflik data.

**Classification**

1. **Support Vector Machine :** SVM merupakan adalah salah satu algoritma yang dapat digunakan untuk klasifikasi dan regresi. Diberikan data dengan kelas yang berbeda, SVM akan mencoba memisahkan kedua data tersebut. Pemisahan data dengan cara memberi pembatas antar kedua data sehingga ketika prediksi, data akan dilihat berada pada bagian mana dari garis tersebut. Oleh karena itu, data training yang dimiliki harus bersifat linearly separable, yang berarti dapat dipisahkan oleh fungsi linear. Jika tidak bersifat linear separable, maka SVM membutuhkan kernel yang dapat digunakan untuk memanipulasi data. Data dimanipulasi dengan dibawa ke dimensi yang lebih besar dengan harapan pada dimensi tersebut data akan menjadi linearly separable. Contoh kernel yang dapat digunakan untuk melakukan mapping adalah kernel polynomial dan RBF (Gaussian radial basis function).   
   
2. **Neural Network :** Neural network dapat digunakan untuk klasifikasi dan regresi. Neural network merupakan kumpulan dari neuron yang masing-masing memiliki fungsi didalamnya. Tiap neuron menerima input dari input data atau neuron pada layer sebelumnya. Lalu, neuron tersebut akan menembakkan suatu bilangan untuk dijadikan output / input pada layer selanjutnya. Pada kasus klasifikasi, input layer terdiri dari neuron sebanyak jumlah atribut. Neuron ini menerima input langsung dari data. Sementara itu, output layer memiliki neuron sebanyak jumlah kelas yang diklasifikasi. Neuron ini akan memberikan output berupa angka hasil klasifikasi untuk tiap kelas. Pengambilan kesimpulan prediksi diambil dari angka kelas yang paling besar. Selain input dan output layer, terdapat hidden layer yang berfungsi untuk membuat fungsi ke derajat yang lebih tinggi.



1. **K-Nearest Neighbour :** K-nearest neighbour merupakan metode klasifikasi yang cukup simple. Metode ini mencari persamaan antara data yang akan diprediksi dengan keseluruhan data training. Lalu, diambil k data yang paling mirip dengan data yang akan diprediksi. Dari k data ini, dicari data dengan kelas yang paling banyak muncul. Kelas yang paling banyak muncul inilah yang akan dijadikan sebagai hasil prediksi.   
   

**Metode Evaluasi**

Untuk menguji akurasi dari prediksi klasifikasi, harus digunakan testing. Oleh karena itu, dataset awal yang memiliki banyak data dipecah menjadi 2. Sebesar kurang lebih 80% dijadikan data training, yaitu data yang akan digunakan untuk pembelajaran metode klasifikasi. 20% data sisanya dijadikan data testing, yaitu data yang digunakan untuk mengetes akurasi dari klasifikasi. Data testing ini tidak boleh digunakan juga untuk training, agar algoritma klasifikasinya tidak “mencontek”. Setelah proses pembelajaran selesai, maka proses testing dapat dimulai. Masukkan data testing (tanpa kelasnya) ke algoritma klasifikasi yang dipilih, lalu bandingkan hasil prediksi dengan kelas aslinya. Apabila hasil prediksi sama dengan kelas asli, maka dapat dinyatakan bahwa prediksi data tersebut benar. Lalu, dapat dilakukan penghitungan akurasi dengan rumus jumlah prediksi benar / jumlah data testing.