BAB II
KAJIAN LITERATUR

2.1 Tinjauan Pustaka

Bagian ini berisi landasan teori terkait teori-teori yang digunakan dan pekerjaan yang sudah dilakukan oleh penelitian sebelumnya untuk mendukung penyelesaian penelitian yang akan dilakukan.

2.2 Data Mining

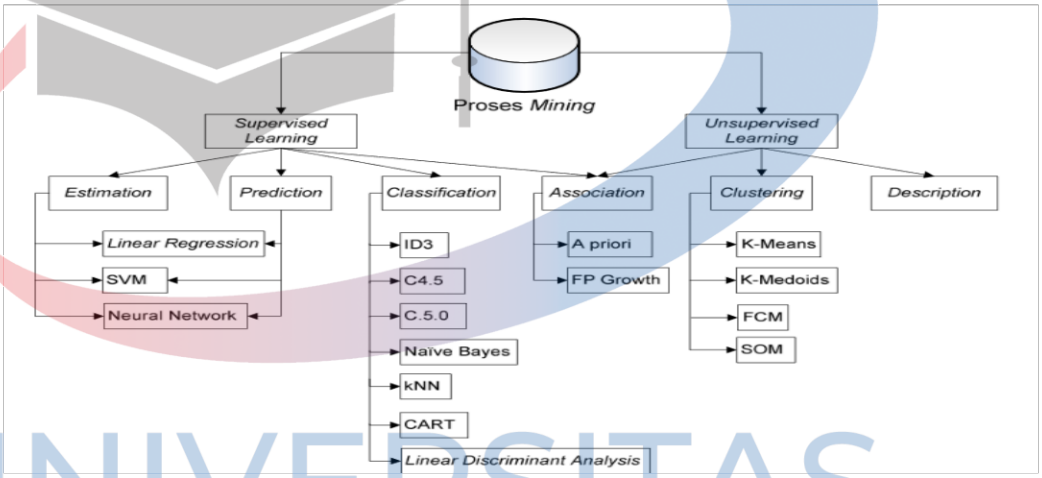
Data mining berisi pencarian *trend* atau pola yang diinginkan dalam Database yang besar untuk membantu pengambilan keputusan di waktu yang akan datang. Harapannya, *Data Mining* mampu mengenali pola-pola ini dalam data dengan masukan minimal (Hermawati:2013). *Data Mining* adalah proses yang memperkerjakan satu atau lebih teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstraksi pengetahuan (*knowledge*) secara otomatis. Definisi lain diantaranya adalah pembelajaran berbasis induksi (*induction based learning*) adalah proses pembentukan definisi-definisi konsep umum yang dilakukan dengan cara mengobservasi contoh-contoh spesifik dari konsep-konsep yang akan di pelajari. Hermawati (2013:3) menyimpulkan bahwa "Data Mining adalah proses iterative dan interaktif untuk menemukan pola atau model baru yang sempurna, bermanfaat dan dapat dimengerti dalam suatu database yang sangat besar (*Massive Database*)".

Analisis yang dilakukan oleh *data mining* mengungguli sistem pendukung keputusan tradisional yang sudah banyak digunakan. *Data mining* dapat menjawab pertanyaan-pertanyaan bisnis yang jika dibandingkan dengan cara tradisional memerlukan banyak waktu dan biaya yang besar. *Data mining* mengeksplorasi basis data untuk menemukan pola-pola yang tersembunyi, mencari informasi untuk memprediksi yang mungkin saja terlupakan oleh para analis karena terletak di luar ekspektasi mereka (Alexander, 2009).

Berdasarkan beberapa pengertian di atas, dapat disimpulkan bahwa *data mining* adalah suatu proses analisis untuk menggali informasi yang tersembunyi

dengan menggunakan statistik dan *artificial intelligence* di dalam suatu database dengan ukuran sangat besar, sehingga ditemukan suatu pola dari data yang sebelumnya tidak diketahui, dan pola tersebut direpresentasikan dengan grafik komputer agar mudah dimengerti. *Data mining* juga dapat memanfaatkan pengalaman atau bahkan kesalahan di masa lalu untuk meningkatkan kualitas dari model maupun hasil analisisnya, salah satunya dengan kemampuan pembelajaran yang dimiliki beberapa teknik *data mining* seperti klasifikasi dan *clustering* (Kusnawi, 2007).

Beberapa metode yang dapat digunakan berdasarkan pengelompokan data mining dapat dilihat pada Gambar 2.1:



Gambar 2.1 Beberapa metode *data mining*
(Ridwan *et al.*, 2013)

2.3 Metode Data Mining

Konsep pembelajaran dalam *data mining* terbagi menjadi 2 macam konsep pembelajaran (Maimon dan Lior, 2010), yaitu pertama *Supervised Learning* yang merupakan algoritma yang membuat suatu fungsi berdasarkan data latihan yang sudah ada, dalam hal ini dapat dikatakan untuk algoritma ini sudah tersedia data latihan secara lengkap dan detil dan terklasifikasi dengan baik yang akan dijadikan sebuah model data saat dilakukan proses ujicoba dengan data tes yang baru dan menghasilkan hasil keluaran yang sesuai diharapkan sebelumnya berdasarkan data latihan yang ada. Kedua adalah *Unsupervised Learning* yang

merupakan algoritma yang berusaha untuk melakukan representasi atau mewakili pola sebuah input yang berasal dari data latihan dan yang menjadi salah satu perbedaan dengan *Supervised Learning* adalah tidak adanya pengklasifikasian dari input data.

2.3.1 Supervised Learning

Supervised Learning merupakan teknik pembelajaran mesin yang membuat suatu fungsi berdasarkan data latihan yang sudah ada, dalam hal ini dapat dikatakan untuk teknik ini sudah tersedia data latihan secara detil dan terklasifikasi dengan baik yang akan dijadikan sebuah model data saat dilakukan proses ujicoba dengan data tes yang baru dan menghasilkan hasil keluaran yang sesuai diharapkan sebelumnya berdasarkan data latihan yang ada (Shwartz dan David, 2014). Beberapa algoritma Supervised Learning diantaranya adalah *Decision tree*, *K- Nearest Neighbor Classifier*, *Naive Bayes Classifier*, *Artificial Neural Network*, dan *Support Vector Machine*.

2.3.1.1 Decision tree

Decision Tree adalah pohon yang digunakan sebagai prosedur penalaran untuk mendapatkan jawaban dari masalah yang dimasukkan. Pohon yang dibentuk tidak selalu berupa pohon biner. Jika semua fitur dalam data set menggunakan 2 macam nilai kategorikal maka bentuk pohon yang didapatkan berupa pohon biner. Jika dalam fitur berisi lebih dari 2 macam nilai kategorikal atau menggunakan tipe numeric maka bentuk pohon yang didapatkan biasanya tidak berupa pohon biner. Kefleksibelan membuat metode ini atraktif, khususnya karena memberikan keuntungan berupa visualisasi sasaran (dalam bentuk decision tree) yang membuat prosedur prediksinya dapat diamati (Gorunescu, 2011). Karakteristik dari decision tree dibentuk sejumlah elemen sebagai berikut (Tan, 2006):

a. Node Akar, tidak mempunyai lengan masukan dan mempunyai nol atau lebih lengan keluaran.

b. Node internal, setiap node yang bukan daun (*nonterminal*) yang mempunyai tepat satu lengan masukan dan dua atau lebih lengan keluaran. Node ini menyatakan pengujian yang didasarkan pada nilai fitur.
c. Lengan, setiap cabang menyatakan nilai hasil pengujian di node bukan daun.
d. Node daun (*terminal*), node yang mempunyai tepat satu lengan masukan dan tidak mempunyai lengan keluaran. Node ini menyatakan label kelas (keputusan)

Ada banyak pilihan algoritma untuk menginduksi decision tree, seperti: Hunt, CART (C&RT), ID3, C4.5, SLIQ, SPRINT, QUEST, DTREG, THAID, CHAID, dan sebagainya.

2.3.1.2 Artificial Neural Network

Artificial Neuron Network (ANN) atau Jaringan Saraf Tiruan adalah sebuah konsep rekayasa pengetahuan yang mengadopsi sistem kerja saraf manusia. Metode ini dapat digunakan untuk pengenalan pola, klasifikasi dan peramalan. Dalam desainnya, ANN memiliki 3 bagian yaitu bagian input, bagian pemrosesan dan bagian output (Prasetyo, 2012). Inputan pada ANN ini dapat berupa vector sehingga perhitungan dalam ANN dapat dilakukan untuk masalah yang kompleks dengan mudah. Dalam prosesnya, metode ANN ini digunakan untuk melakukan peramalan dan pengenalan pola dalam *data mining*. Untuk melakukannya, ANN memerlukan proses pelatihan agar dapat melakukan prediksi kelas dari suatu data uji coba. Dalam proses penambangan data, ANN menggunakan fungsi aktivasi yang digunakan untuk membatasi keluaran dari bagian pemrosesan atau neuron agar sesuai dengan batasan yang diinginkan. Terdapat berbagai algoritma yang dapat digunakan untuk menggunakan metode ini. Salah satunya adalah algoritma *Backpropagation*.

Algoritma *Backpropagation* adalah salah satu algoritma yang digunakan untuk melakukan pelatihan pada metode ANN. Algoritma ini bersifat nonlinear yang dapat mengatasi berbagai masalah yang rumit. Algoritma ini memiliki dasar matematis yang tinggi dan dilatih menggunakan metode belajar terbimbing dimana hasil atau tujuannya sudah diketahui sebelumnya. Pada algoritma ini,

jaringan akan diberikan sepasang pola yang merupakan masukan dan pola yang diinginkan. Ketika pola dimasukan ke dalam jaringan maka bobot-bobot akan diubah untuk memperkecil perbedaan pola keluaran dengan pola yang diinginkan. Pelatihan ini dilakukan berulang-ulang sehingga memenuhi pola yang diinginkan, Algoritma ini mendukung jenis ANN yang bersifat multi layer atau biasa disebut Multi Layer Precepton (MLP). Pada algoritma ini terdiri dari 3 *layer* yaitu *layer* input, *layer* tersembunyi dan *layer output*.

Backpropagation klasifikasi memiliki metode yang digunakan untuk melakukan pembelajaran terhadap kumpulan data dan kemudian memetakan masing-masing data yang terpilih ke dalam salah satu dari kelompok kelas yang telah ditetapkan sebelumya. Tujuan dari klasifikasi yaitu memperkirakan kelas yang dimiliki dari suatu objek dimana objek tersebut belum diketahui labelnya. Proses klasifikasi ini melakukan proses pencarian model atau fungsi yang dapat menjelaskan atau membedakan kelas dari data tertentu (Han, *et al.*, 2012)

2.3.1.3 Klasifikasi *Support Vector Machine*

Support Vector Machine (SVM) adalah sebuah algoritma yang bekerja dengan nonlinear mapping yang berfungsi untuk mentransformasikan data training awal ke dimensi baru yang lebih tinggi. Pada dimensi yang baru ini, SVM akan menemukan *hyperplane* linear yang optimum. Dengan melakukan nonlinear mapping ke dimensi yang lebih tinggi, data dari dua kelas pasti akan selalu dapat dipisahkan oleh sebuah *hyperplane*. Metode ini akan menemukan *hyperplane* dengan menggunakan *support vectors* dan *margins* (Han *et al.*, 2011).

Metode ini pertama kali dipresentasikan pada tahun 1992 oleh Vapnik, Boser, dan Guyon pada *Workshop on Computational Learning*. Teori SVM memperkenalkan strategi baru dengan mencari hyperplane terbaik pada ruang input. Prinsip SVM mula-mula adalah linear *classifier*, tetapi SVM kemudian dikembangkan agar mampu bekerja pada masalah non-linear dengan memasukkan kernel. Perkembangan SVM ini menstimulasi minat penelitian di bidang *pattern recognition* dalam mengembangkan potensi kemampuan metode SVM baik dari segi teoretis maupun dari segi aplikasi. Dewasa ini SVM telah berhasil

diaplikasikan dalam menyelesaikan masalah praktis. Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada ruang input.

Masalah klasifikasi dapat diartikan sebagai usaha menemukan garis yang memisahkan antara kedua kelompok tersebut. *Hyperplane* pemisah terbaik antara kedua kelas dapat ditemukan dengan mengukur margin hyperplane dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* dengan *pattern* terdekat dari masing-masing kelas. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Usaha untuk mencari lokasi hyperplane ini merupakan inti dari proses pembelajaran SVM.

2.3.1.4 Naive Bayes Classifier

Algoritma Naive Bayes adalah salah satu algoritma yang terdapat pada teknik *data mining* klasifikasi. Naive bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuan Inggris yaitu Thomas bayes, Naive Bayes memprediksi peluang dimasa depan berdasarkan pengalaman dimasa sebelumnya, sehingga dikenal dengan Teorema Bayes. Teorema tersebut dikombinasikan dengan Naive dimana diasumsikan kondisi antar atribut saling bebas. Klasifikasi Naive Bayes diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya (Taheri dan Mammadov 2015).

Persamaan dari teorema Bayes adalah:

$$P(H|X) = \frac{P(X|H)\times P(H)}{P(X)} \quad \text{.....(2.1)}$$

Keterangan:

X : Data dengan class yang belum diketahui

H : Hipotesis data X merupakan suatu class spesifik

$P(H|X)$: Probabilitas hipotesis H berdasar kondisi X (posteriori probability)

$P(H)$: Probabilitas hipotesis H (prior probability)

$P(X|H)$: Probabilitas X berdasarkan kondisi pada hipotesis H

$P(X)$: Probabilitas X

Adapun alur dari metode Naive Bayes adalah sebagai berikut:

1. Baca data training
2. Hitung Jumlah dan probabilitas, namun apabila data numerik maka:
    a. Cari nilai mean dan standar deviasi dari masing-masing parameter yang merupakan data numerik.
    b. Cari nilai probabilistik dengan cara menghitung jumlah data yang sesuai dari kategori yang sama dibagi dengan jumlah data pada kategori tersebut.
3. Mendapatkan nilai dalam tabel mean, standar deviasi dan probabilitas.

2.3.1.5 K Nearest-Neighbor

Metode k-Nearest Neighbor (k-NN) pertama kali diperkenalkan pada awal 1950-an. Metode ini belum mendapatkan perhatian sampai tahun 1960-an, mengingat sifat metode ini yang labor intensive ketika diberikan data training yang sangat besar. Baru pada saat teknologi komputasi semakin maju, metode ini mulai banyak digunakan terutama pada bidang *pattern recognition* (Han et al., 2011)

Klasifikasi berbasis Nearest Neighbor (NN) menjadi salah satu metode dalam top sepuluh metode *data mining* yang paling popular digunakan (Wu dan Kumar, 2009). Metode NN yang murni termasuk dalam klasifikasi yang *lazy learner* karena menunda proses pelatihan (atau bahkan tidak melakukan sama sekali) sampai ada data uji yang ingin diketahui label kelasnya, maka metode baru akan menjalankan algoritmanya. Algoritma Nearest Neighbor melakukan klasifikasi berdasarkan kemiripan suatu data dengan data yang lain (Tan *et al.*, 2005).

Metode K-Nearest Neighbor (K-NN) menjadi salah satu metode berbasis NN yang paling tua dan popular. Nilai K yang digunakan di sini menyatakan jumlah tetangga terdekat yang dilibatkan dalam penentuan prediksi label kelas

pada data uji. Dari K tetangga terdekat yang terpilih kemudian dilakukan voting kelas dari K tetangga terdekat tersebut. Kelas dengan jumlah suara tetangga terbanyaklah yang diberikan sebagai label kelas hasil prediksi pada data uji tersebut (Tan *et al*, 2005). Berikut algoritma prediksi dengan K-NN:

1. $Z = (x', y')$ adalah data uji dengan data x' dan label kelas y' yang belum diketahui
2. C adalah himpunan label kelas data.
3. Hitung jarak $d(x', x)$ jarak di antara data uji z ke setiap vector data latih, simpan dalam D.
4. Pilih $D_z \in D$, yaitu K tetangga terdekat dari z

Dalam algoritma ini, tidak ada rumus baku untuk menentukan nilai k yang paling optimal. Beberapa penelitian merekomendasikan beberapa cara untuk menentukannya. Ada penelitian yang menyarankan nilai k sama dengan akar dari jumlah data training (*neighbors*) (Jonsson & Wohlin, 2004). Ada yang menyarankan dengan metode eksperimen, dengan memulai eksperimen dari nilai $k = 1,2,3$,dst., lalu dipilih nilai k yang menghasilkan akurasi terbaik (Han et al., 2011). Ada pula penelitian yang memberikan rekomendasi nilai k sebaiknya bernilai ganjil dan bukan merupakan kelipatan dari jumlah kelas, yang dimaksudkan supaya proses algoritma berjalan lebih cepat, dengan menghindari kesempatan dua atau lebih kelas mendapatkan votes yang sama (Hassanat, *et al.*, 2014).

Ada dua kekurangan k-NN, yaitu (Hassanat *et al.*, 2014):

1. Karena metode ini menggunakan seluruh data training dalam setiap test, tidak ada model output yang dihasilkan.
2. Performa klasifikasi bergantung pada nilai jumlah neighbor (k) yang membedakan data sampel satu dan yang lainnya.

2.3.1.6 *Fuzzy K Nearest-Neighbor*

Fuzzy K-Nearest Neighbor (FK-NN) merupakan salah satu metode klasifikasi dengan menggabungkan teknik *Fuzzy* dan K-NN. Metode ini tidak

seperti metode lain yang mana pada metode ini akan secara tegas memprediksi kelas yang diikuti oleh data uji berdasarkan perbandingan *K* terdekat. Dasar dari algoritma FK-NN adalah untuk menetapkan nilai keanggotaan sebagai fungsi jarak vektor dari KNN dan keanggotaan tetangga mereka di kelas-kelas yang memungkinkan. Metode ini berperan penting dalam menghilangkan ambiguitas dalam klasifikasi. Selain itu, sebuah instance akan memiliki derajat nilai keanggotaan pada setiap kelas sehingga akan lebih memberikan kekuatan atau kepercayaan suatu instance berada pada suatu kelas (Nugraha *et al.*, 2017).

Algortime FK-NN memberikan nilai keanggotaan kelas pada data uji bukan menempatkan data uji pada kelas tertentu. FK-NN merupakan metode klasifikasi yang digunakan untuk memprediksi data uji menggunakan nilai derajat keanggotaan data uji pada setiap kelas (Anugerah *et al.*, 2018).

Sebelum menghitung nilai keanggotaan pada *Fuzzy* K-NN, terlebih dahulu dilakukan proses menggunakan pada persamaan 2.2 berikut ini.

$$u_{ij} = \begin{cases} 0{,}51 + \left(\frac{n_j}{n}\right) * 0{,}49, jika\ j = 1 \\ \left(\frac{n_j}{n}\right) * 0{,}49, jika\ j \neq 1 \end{cases} \quad \text{.....(2.2)}$$

Keterangan:

n_j = Jumlah anggota kelas j pada suatu data latih n.

n = Jumlah data latih yang digunakan

j= kelas data

Selanjutnya menghitung nilai keanggotaan masing-masing kelas dengan persamaan 2.3 berikut:

$$u_i(x) = \frac{\sum_{j=1}^{k} u_{ij}\left(\|x - x_j\|^{-2/(m-1)}\right)}{\sum_{j=1}^{k}\left(\|x - x_j\|^{-2/(m-1)}\right)} \quad \text{.....(2.3)}$$

Keterangan:

u_{ij} = nilai keanggotaan *fuzzy* pada contoh pengujian (x, x_j)

k = nilai tetangga terdekat

j = variable data keanggotaan data uji

m = bobot pangkat yang besarnya $m > 1$.

2.3.2 Unsupervised Learning

Unsupervised Learning merupakan teknik pembelajaran mesin yang berusaha untuk melakukan representasi pola sebuah input yang berasal dari data latihan dan salah satu yang menjadi perbedaan dengan *Supervised Learning* adalah tidak adanya pengklasifikasian dari input data. Dalam *Machine Learning* teknik *Unsupervised Learning* menjadi esensial karena sistem kerja yang diberikan sama dengan cara kerja otak manusia dimana dalam proses pembelajaran tidak ada role model atau informasi dan contoh yang tersedia untuk dijadikan sebagai model dalam melakukan proses ujicoba untuk penyelesaian sebuah masalah dengan data yang baru (Shwartz dan David, 2014). Beberapa algoritma *Unsupervised Learning* diantaranya adalah: K-Means Clustering, Hierarchical Clustering, dan *Fuzzy* C-Means.

2.3.2.1 K-Means Clustering

Algoritma K-means merupakan salah satu algoritma dengan partitional, karena K-Means didasarkan pada penentuan jumlah awal kelompok dengan mendefinisikan nilai *centroid* awalnya (Madhulatha, 2012). Algoritma K-means menggunakan proses secara berulang-ulang untuk mendapatkan basis data *cluster*. Dibutuhkan jumlah *cluster* awal yang diinginkan sebagai masukan dan menghasilkan titik *centroid* akhir sebagai output. Metode K-means akan memilih pola *k* sebagai titik awal *centroid* secara acak atau random. Jumlah iterasi untuk mencapai cluster *centroid* akan dipengaruhi oleh calon *cluster centroid* awal secara random. Sehingga didapat cara dalam pengembangan algoritma dengan menentukan *centroid cluster* yang dilihat dari kepadatan data awal yang tinggi agar mendapatkan kinerja yang lebih tinggi.

Dalam penyelesaiannya, algoritma K-Means akan menghasilkan titik *centroid* yang dijadikan tujuan dari algoritma K-Means. Setelah iterasi K-Means berhenti, setiap objek dalam dataset menjadi anggota dari suatu *cluster*. Nilai *cluster* ditentukan dengan mencari seluruh objek untuk menemukan *cluster* dengan jarak terdekat ke objek. Algoritma K-means akan mengelompokan item data dalam suatu dataset ke suatu cluster berdasarkan jarak terdekat (Bangoria et

al., 2013). Nilai *centroid* awal yang dipilih secara acak yang menjadi titik pusat awal, akan dihitung jarak dengan semua data menggunakan rumus Euclidean Distance. Data yang memiliki jarak pendek terhadap *centroid* akan membuat sebuah cluster. Proses ini berkelanjutan sampai tidak terjadi perubahan pada setiap kelompok

Parameter yang harus dimasukkan ketika menggunakan algoritma K-Means adalah nilai K. Nilai K yang digunakan biasanya didasarkan informasi yang diketahui sebelumnya tentang sebenarnya berapa banyak cluster data yang muncul dalam X, berapa banyak cluster yang dibutuhkan untuk penerapannya, atau jenis cluster dicari dengan mengeksplorasi atau melakukan percobaan dengan beberapa nilai K. Berapa nilai K yang dipilih tidak perlu memahami bagaimana K-Means mempartisi set data X. Dalam K-Means, setiap cluster dari K cluster diwakili oleh titik tunggal dalam $\Re^d$. Set representatif cluster dinyatakan $C = \{cj/ = 1, \dots, K\}$. Sejumlah K representatif cluster tersebut disebut juga sebagai cluster means atau cluster centroid (atau *centroid* saja). Untuk set data dalam X dikelompokkan berdasarkan konsep kedekatan atau kemiripan. Meskipun konsep yang dimaksud untuk data-data yang berkumpul dalam satu cluster adalah data-data yang mirip, tetapi kuantitas yang digunakan untuk mengukurnya adalah ketidakmiripan (dissimilarity). Artinya, data-data dengan ketidakmiripan (jarak) yang kecil/dekat maka lebih besar kemungkinannya untuk bergabung dalam satu cluster. Metrik yang umum dugunakan untuk ketidakmiripan adalah Euclidean (Chandra *et al.*, 2014). Berikut merupakan prosedur algoritma pengelompokan K-Means menurut Prasetyo (2014):

1. Inisialisasi: tentukan nilai K sebagai jumlah cluster yang diinginkan dan metrik ketidakmiripan (jarak) yang diinginkan. Jika perlu, tetapkan ambang batas perubahan fungsi objektif dan ambang batas perubahan posisi *centroid*.
2. Pilih K data dari set data X sebagai *centroid*.
3. Alokasikan semua data ke *centroid* terdekat dengan metrik jarak yang sudah ditetapkan (memperbarui cluster ID setiap data)
4. Hitung kembali *centroid* berdasarkan data yang mengikuti cluster masing-masing.

5. Ulangi langkah 3 dan 4 hingga kondisi konvergen tercapai, yaitu (a) perubahan fungsi objektif sudah di bawah ambang batas yang diinginkan; atau (b) tidak ada data yang berpindah cluster; atau (c) perubahan posisi *centroid* sudah di bawah ambang batas yang ditetapkan. Menurut Tutik (2014) untuk menentukan nilai pusat (*centroid*) pada tahap *iterasi* digunakan rumus persamaan 2.4 sebagai berikut:

$$v_{ij} = \frac{1}{N_i} = \sum_{k=0}^{N_i} x_{ki} \quad \text{.....(2.4)}$$

Dimana:

$\bar{V}ij$ = *centroid* rata-rata *cluster* ke-i untuk variable *k-j*

Ni = jumlah anggota *cluster* ke-*i*

i, k = indeks dari *cluster*

j = indeks dari variable

Xkj = nilai data ke-*k* variable ke-*j* dalam *cluster* tersebut

Menurut Afrisawati (2013) untuk menentukan korelasi antar dua obyek yaitu dengan menggunakan rumus *Euclidean Distance* seperti persamaan 2.5 berikut:

$$d_{Euclidean}(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad \text{.....(2.5)}$$

Dimana:

d (xy) = jarak data ke *x* ke pusat *cluster y*

x_i = data ke-*i* pada atribut data ke *n*

y_i = data ke-*j* pada atribut data ke *n*

2.3.2.2 Hierarchical Clustering

Dalam statistik, pengelompokan berbasis hirarki adalah metode analisis cluster yang berusaha untuk membangun sebuah hieark cluster. Strategi untuk pengelompokan berbasis hierarki umumnya jatuh ke dalam dua jenis, yaitu *aglomeratif* dan *divisive*. Pembahasan di bab ini dibatasi hanya pada *Agglomerative Hierarchical Clustering* (AHC). Aglomeratif merupakan metode pengelompokan berbasis hierarki dengan pendekatan buttom up, yaitu proses pengelompokan dimulai dari masing-masing data sebagai satu buah cluster,

kemudian secara rekursif mencari cluster terdekat sebagai pasangan untuk bergabung sebagai satu cluster yang lebih besar (Presetyo, 2013). Proses tersebut diulang terus sehingga tampak bergerak ke atas membentuk hierarki. Cara ini membutuhkan suatu parameter kedekatan cluster (*cluster proximity*).

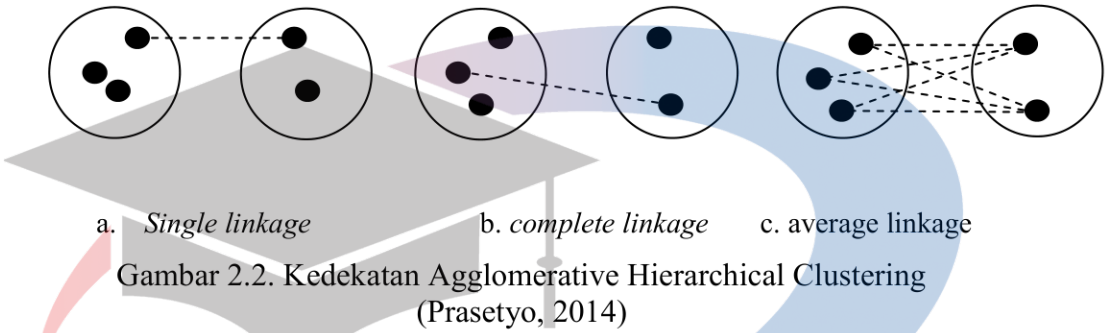
Divisif merupakan metode pengelompokan berbasis hierarki dengan pendekatan *top down*, yaitu proses pengelompokan dimulai dari satu cluster yang berisi semua data, kemudian secara rekursif memecah cluster menjadi dua cluster sampai setiap cluster hanya berisi satu data tunggal (data itu sendiri). Untuk cara ini, yang dibutuhkan adalah keputusan cluster yang manakah yang akan dipecah pada setiap langkah dan bagaimana cara memecahkannya.

Pengelompokan berbasis hierarki sering ditampilkan dalam bentuk grafis menggunakan diagram yang mirip pohon (*tree*) yang disebut dengan *dendrogram*. *Dendrogram* merupakan diagram yang menampilkan hubungan cluster dan subcluster-nya dalam urutan yang mana cluster yang digabung (*agglomerative view*) atau dipecah (*divisive view*). Algoritma AHC dijabarkan dalam Algoritma berikut (Prasetyo, 2014), berikut langkah Algoritma Agglomerative Hierarchical Clustering:

1. Hitung jarak dari semua objek. Nyatakan hasil perhitungan jarak ke dalam matriks jarak.
2. Lakukan pencarian disemua sel matriks jarak untuk menemukan dua cluster/objek yang paling mirip/serupa.
3. Gabungkan dua cluster/ objek terdekat berdasarkan parameter kedekatan yang ditentukan untuk menghasilkan sebuah cluster yang memiliki minimal 2 objek.
4. Perbarui matriks jarak dengan menghitung jarak antara cluster baru dan semua cluster yang lain. Ulangi langkah 2 sampai semua objek masuk ke dalam satu cluster.

Kunci operasi metode AHC adalah penggunaan ukuran kedekatan di antara dua cluster (Hartini, 2012). Ada tiga teknik kedekatan yang digunakan AHC dalam pembahasan di sini yaitu *Single linkage* (jarak terdekat), *complete*

linkage (jarak terjauh), dan *average linkage* (jarak rerata), seperti yang diilustrasikan pada Gambar 2.2.



a. *Single linkage* b. *complete linkage* c. average linkage

Gambar 2.2. Kedekatan Agglomerative Hierarchical Clustering (Prasetyo, 2014)

Single linkage memberikan hasil bila clusters digabungkan menurut jarak antara anggota-anggota yang paling dekat di antara dua cluster. *Complete linkage* terjadi bila kelompok-kelompok digabungkan menurut jarak antara anggota-anggota yang paling jauh di antara dua cluster. Untuk *average linkage* digabungkan menurut jarak rata-rata antara pasangan-pasangan anggota masing-masing pada himpunan di antara dua cluster. Hasil-hasil dari clustering kedekatan tersebut dapat disajikan secara grafik dalam bentuk dendrogram. Cabang-cabang dalam pohon menyajikan cluster. Kemudian, cabang-cabang bergabung pada node yang posisinya sepanjang sumbu jarak (similaritas) menyatakan tingkat di mana penggabungan terjadi.

Pada metode *Single linkage* (MIN), kedekatan di antara dua cluster ditentukan dari jarak terdekat (terkecil) di antara pasangan dua data dari dua cluster berbeda (satu dari cluster pertama satu dari cluster yang lain) atau disebut juga nilai kemiripan yang paling maksimal. Maka, dengan cara ini kita memulainya dari masing-masing data sebagai cluster, kemudian mencari tetangga terdekat dan menggunakan *Single linkage* untuk menggabungkan dua cluster berikutnya sampai semuanya bergabung menjadi satu cluster. Metode ini bagus

untuk menangani set data yang bentuk distribusi datanya non-elips (*non-elliptical shapes*), tapi sangat sensitive terhadap *noise* dan *outlier* (Prasetyo, 2014).

Pada metode *complete linkage* (MAX), kedekatan di antara dua cluster ditentukan dari jarak terjauh (terbesar) di antara pasangan dua data dari dua cluster berbeda (satu dari cluster pertama satu dari cluster yang lain) atau disebut juga nilai kemiripan yang paling minimal. Maka dengan cara ini kita memulainya dari masing-masing data sebagai cluster, kemudian mencari tetangga terdekat dan menggunakan *complete linkage* untuk menggabungkan dua cluster berikutnya sampai semuanya bergabung menjadi satu cluster. Metode ini kurang peka terhadap *noise* dan *outlier*, tetapi bagus untuk data yang mempunyai distribusi bentuk bulat.

Pada metode *average linkage* (*average*), kedekatan di antara dua cluster ditentukan dari jarak rata-rata di antara pasangan di antara dua data dari dua cluster berbeda (satu dari cluster pertama satu dari cluster yang lain) atau disebut juga nilai rata-rata di antara *Single linkage* dan *complete linkage*. Maka, dengan cara ini kita memulainya dari masing-masing data sebagai cluster, kemudian mencari tetangga terdekat dan menggunakan *average linkage* untuk menggabungkan dua cluster berikutnya sampai semuanya bergabung menjadi satu cluster. Metode ini merupakan pendekatan yang mengambil pertengahan di antara *Single linkage* dan *complete linkage*.

2.3.2.3 *Fuzzy C-Means*

Fuzzy C-means Clustering (FCM), atau dikenal juga sebagai *Fuzzy* ISODATA, merupakan salah satu metode clustering yang merupakan bagian dari metode Hard K-Means. FCM menggunakan model pengelompokan fuzzy sehingga data dapat menjadi anggota dari semua kelas atau cluster terbentuk dengan derajat atau tingkat keanggotaan yang berbeda antara 0 hingga 1. (TL emha, 2007).

Konsep dasar FCM, pertama kali adalah menentukan pusat cluster, yang akan menandai lokasi rata-rata untuk tiap-tiap *cluster*. Pada kondisi awal, pusat cluster ini masih belum akurat. Tiap-tiap titik data mewakili derajat keanggotaan

untuk tiap-tiap cluster. Dengan cara memperbaiki pusat cluster dan derajat keanggotaan tiap-tiap titik data secara berulang, maka akan dapat dilihat bahwa pusat cluster akan bergerak menuju lokasi yang tepat. Perulangan ini didasarkan pada minimisasi fungsi obyektif yang menggambarkan jarak dari titik data yang diberikan ke pusat cluster yang terbobot oleh derajat keanggotaan titik data tersebut. (Kusumadewi S & Purnomo H, 2004)

Fuzzy C-Means berhubungan dengan konsep kesamaan fungsi objek yang berdekatan dan menemukan titik pusat *cluster* sebagai *prototype*. Untuk beberapa objek data tidak memiliki batasan pada salah satu kelas saja tetapi data tersebut dapat dikelompokan berdasarkan derajat keanggotaan yaitu antara 0 dan 1 yang menunjukkan keanggotaan parsial dari data tersebut (Phukon dan Baruah, 2013). Beberapa contoh dalam penerapan *Fuzzy C-Means* adalah masalah pengelompokan data nyata yang telah dibuktikan dengan menghasilkan karateristik data yang baik (Phukon dan Baruah, 2013). Algoritma ini dimulai dengan menentukan jumlah *cluster* yang diinginkan serta menginisialisasikan nilai keanggotaan yang berisikan semua data kemudian akan dikelompokan berdasarkan *cluster*nya. Pusat pusat *cluster* dihitung dari jarak terdekat ke titik-titik yang memiliki nilai keanggotaan lebih besar. Dengan kata lain, nilai-nilai keanggotaan tersebut akan bertindak sebagai nilai bobot sementara pada suatu *cluster* (K.G dan Patnaik, 2006).

Algoritma *Fuzzy C-Means* memiliki keuntungan yaitu:

1. Dalam implementasi menyelesaikan masalah algoritma *Fuzzy C-Means* dapat memahami karakteristik data yang kabur atau data yang tidak terdefinisikan.
2. Memiliki kemampuan dalam mengelompokan data yang besar
3. Lebih kokoh terhadap data *outlier*/data dengan karakter yang berbeda atau *value* yang berbeda dalam satu atau beberapa variabel
4. Penentuan titik *cluster* yang optimal (Ali et al., 2008; Suganya dan Shanthi, 2012; Martino dan Sessa, 2009)
5. Dapat melakukan *clustering* lebih dari satu variabel secara sekaligus (Simbolon et al., 2013).

Beberapa kelemahan yang dimiliki oleh algoritma *Fuzzy C-Means* yaitu:

1. Pada algoritma *Fuzzy C-Means* user memerlukan lebih banyak waktu untuk proses perhitungan komputasinya dalam menentukan *cluster* pada setiap anggota di suatu dataset (Bora dan Gupta, 2014)
2. Masih terpengaruh terhadap cara pembagian data yang sering dipergunakan pada data yang sama dan sangat sensitif terhadap kondisi awal seperti jumlah cluster dan titik pusat clusterpada pengelompokan data (Lu et al., 2013).

2.4 PSO

2.4.1 Pengenalan PSO

Particle swarm optimization (PSO) merupakan salah satu metode untuk menyelesaikan masalah-masalah optimisasi yang termasuk dalam metode metaheuristik, artinya PSO berhubungan dengan sesuatu yang random (stokastik) dalam memecahkan masalah optimisasi yang dihadapi, dan PSO bisa digunakan pada masalah optimisasi yang oleh metode-metode klasik seperti metode Newton, Quasi-Newton, dan Gradient Descent tidak dapat diselesaikan. PSO dikembangkan berdasarkan pengamatan akan perilaku kawanan hewan dalam mencari makanan atau menghindari predator yang kemudian dimodelkan secara matematis untuk dapat dikerjakan pada komputer.

Pada awalnya, PSO diciptakan untuk memecahkan masalah-masalah yang ruang pencarian solusinya kontinu. PSO juga dapat digunakan untuk mencari solusi pada ruang pencarian yang diskret. Hal itu dapat dilakukan cara multiple PSO kontinu secara sederhana dengan membulatkan bilangan kontinu ke bilangan bulat terdekat. Ada pula cara yang lebih rumit untuk memultiple PSO kontinu ke PSO diskret misalnya dengan menggunakan fungsi sigmoid (Khanesar et al., 2007). Selain kontinu dan diskret, PSO juga dapat digunakan untuk memecahkan masalah-masalah yang bersifat *kombinatorial*, misalnya untuk memecahkan masalah *travelling salesman* (Zhong et al., 2007). Tasgetiren et al. (2004) mengusulkan metode untuk memecahkan permasalahan-permasalahan kombinatorial dengan menggunakan metode *smallest position value* (SPV).

Komputasi paralel telah digunakan untuk mengerjakan implementasi dari algoritma PSO. Zhou et al., (2009) dan Mussi et al. (2009), mengimplementasikan PSO yang telah dimultiple, yaitu Standard PSO (SPSO) dengan menggunakan

CUDA. Liera et al., (2011) mengimplementasikan PSO untuk menguji fungsi *Rastrigin* dan *Ackley* pada ruang pencarian 30 dimensi. Nashed et al., (2011) mengimplementasikan PSO pada GPU dengan menggunakan OpenCL.

Beberapa istilah umum yang biasa digunakan dalam *Optimisasi Particle Swarm* dapat didefinisikan sebagai berikut (Tuegeh, et al., 2009):

1. *Swarm*: populasi dari suatu algoritma.
2. *Particle*: anggota (individu) pada suatu *swarm*.

   Setiap *particle* merepresentasikan suatu solusi yang potensial pada permasalahan yang diselesaikan. Posisi dari suatu *particle* adalah ditentukan oleh representasi solusi saat itu.
3. Pbest (*Personal best*): posisi Pbest suatu *particle* yang menunjukkan posisi *particle* yang dipersiapkan untuk mendapatkan suatu solusi yang terbaik.
4. Gbest (*Global best*): posisi terbaik *particle* pada *swarm*.
5. Velocity (vektor): vektor yang menggerakkan proses optimisasi yang menentukan arah di mana suatu *particle* diperlukan untuk berpindah (*move*) untuk memperbaiki posisinya semula.
6. *Inertia weight*: *inertia weight* disimbolkan w, parameter ini digunakan untuk mengontrol dampak dari adanya velocity yang diberikan oleh suatu *particle*.

Penelitian ini dilakukan dalam beberapa tahap, meliputi tahap pra-proses data, pemilihan parameter k, m yang optimal menggunakan MPSO dan FKNN. Prosedur peyelesaian masalah menggunakan MPSO dan FKNN dapat dijelaskan dalam bentuk langkah-langkah berikut ini:

Langkah 1: Inisialisasi nilai k dan m. Dalam penerapan FK-NN nilai parameter k dan m harus diinisialisasi terlebih dahulu.

Langkah 2: Bangkitkan Partikel awal secara *random*. Nilai k dan m dibangkitkan secara parallel, dalam penelitian ini partikel yang digunakan sebanyak 50 partikel.

Langkah 3: Nilai k dan m di atas digunakan untuk melatih (*training*) data latih.

Langkah 4: Setiap partikel dievaluasi berdasarkan nilai fitness. Fungsi fitness (fitness *function*) merupakan fungsi yang digunakan untuk menghitung fitness atau tingkat kebaikan suatu individu untuk bertahan hidup.

Langkah 5: Update posisi dan kecepatan partikel. Prosedur PSO mengharuskan posisi dan kecepatan setiap partikel.

Langkah 6: Pada tahap ini kembali dilakukan pengujian model FK-NN untuk menemukan partikel terbaik. Partikel terbaik yang dievaluasi berdasarkan nilai fitness pada langkah 4 diterapkan untuk melatih data latih kembali dan dihitung nilai fitness nya.

Langkah 7: Perbaharui personal optimal fitness (pfit) dan personal optimal position (pbest). Sejauh ini semua partikel pada iterasi awal (pertama) telah dievaluasi, iterasi pertama menghasilkan sebuah partikel terbaiknya (pbest).

Langkah 8: Pada tahap ini dievaluasi apakah seluruh iterasi telah selesai dikerjakan. Jika belum akan dilanjutkan pada langkah 3, jika telah selesai maka dilanjutkan pada langkah 10.

Langkah 9: Setiap iterasi memiliki partikel terbaik. Pada tahap ini partikel terbaik dari semua iterasi akan dievaluasi untuk menentukan partikel terbaik global optimal position (gbest).

Langkah 10: Jika nilai gbest pada langkah 9 merupakan nilai yang diharapkan maka pada saat tersebut telah diperoleh nilai k dan m yang optimal dan dilanjutkan pada langkah 11. Jika nilai gbest belum sesuai dengan kriteria yang diharapkan maka kembali dibangkitkan populasi baru dengan kembali pada langkah 4.

Langkah 12: Lakukan pengujian klasifikasi data dengan memasukkan data uji

Langkah 13: Lakukan *cross validation* untuk mengetahui apakah hasil klasifikasi pada langkah 11 telah memiliki akurasi paling baik. Jika ya maka seluruh proses selesai, jika tidak maka lakukan evaluasi nilai K. Sampai diperoleh nilai akurasi paling baik.

2.5 Ukuran Kemiripan (Jarak Euclidean)

Ukuran kesamaan $D(T,U)$ antara deret berkala T dan U adalah fungsi yang mengambil dua seri waktu sebagai input dan menghasilkan jarak d antara seri ini. Jarak ini harus tidak negatif, yaitu $D(T,U) \geq 0$. Jika ukuran ini memenuhi tambahan properti simetri $D(T,U) = D(U,T)$ dan subadditivitas $D(T,V) \leq D(T,U) + D(U,V)$ (juga dikenal sebagai segitiga ketidaksetaraan), jarak dikatakan sebagai metrik. Metrik merupakan ukuran yang sangat efisien untuk pengindeksan. (Prasetyo, 2014).

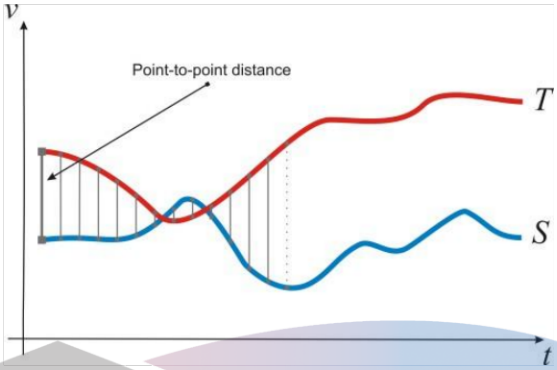
Ukuran kemiripan yang paling umum digunakan adalah jarak Euclidean yang diformulasikan oleh persamaan 2.6 berikut:

$$D(T,U) = \|T - U\| = \sqrt{\sum_{i=1}^{r} (T_i - U_i)^2} \qquad (2.6)$$

$T, U \epsilon X$dan Ti, Ui adalah nilai fitur ke-i dari T dan U, sedangkan r adalah jumlah fitur dalam vector. Ukuran kemiripan Euclidean akan memberikan $d0 = 0$, maka jarak yang mungkin di antara dua vector data juga 0. Selain itu, jarak dari T ke U akan sama dengan U ke $T, D(T,U) = D(U,T)$.

Jarak Euclidean dikatalogkan sebagai fungsi jarak metrik, karena itu mematuhi properti metrik: tidak negatif, identitas, simetri dan segitiga ketidaksamaan. Gambar 2.3 menunjukkan model perhitungan menggunakan jarak Euclidean yaitu hasil akar dari penjumlahan jarak poin ke poin (garis abu-abu), di sepanjang deret berkala. Secara kompetitif dengan pendekatan lain yang lebih kompleks, terutama ketika ukuran dataset semakin besar. Dalam segala hal, jarak Euclidean dan variannya menghadirkan beberapa kekurangan, yang membuat penggunaannya tidak sesuai aplikasi:

1. Hanya membandingkan deret berkala yang sama panjangnya.
2. Tidak menangani pencilan atau keterasingan.
3. Hal ini sangat sensitif terhadap enam transformasi sinyal: pergeseran, skala amplitudo seragam, skala waktu seragam, skala bi skala seragam, *time warping* dan skala amplitudo non-seragam.



Gambar 2.3 *T* dan *S* adalah dua deret berkala dari Sebuah variabel tertentu yaitu *v,* di sepanjang waktu di sumbu *t*
(Rakthanmanon, et al., 2012)

2.6 Evaluasi Klasifikasi

Pengukuran terhadap kinerja suatu sistem klasifikasi merupakan hal yang penting dalam mengevaluasi kinerja dari suatu model klasifikasi. Berikut akan dijelaskan terkait pengukuran kinerja klasifikasi.

2.6.1 Pengukuran Kinerja

Kinerja sistem klasifikasi menggambarkan seberapa baik sistem dalam mengklasifikasikan data. Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi.

2.6.1.1 Akurasi

Sebuah system yang melakukan klasifikasi diharapkan dapat melakukan klasifikasi semua set data dengan benar. Akan tetapi, tidak dapat dipungkiri bahwa kinerja suatu sistem tidak bisa bekerja 100% benar. Oleh karena itu, sebuah system klasifikasi juga harus diukur kinerjanya. Umumnya cara mengukur kinerja klasifikasi menggunakan matriks confusion. Matriks confusion merupakan tael yang mencatat hasil kerja klasifikasi. Tabel 2.1 merupakan contoh matriks confusion yang melakukan klasifikasi masalah biner (dua kelas), misalnya kelas 0 dan 1. Setiap sel fij dalam matriks menyatakan jumlah record/data kelas i yang hasil prediksinya masuk ke kelas j. misalnya sel f_{11} adalah jumlah data dalam

kelas 1 yang secara benar dipetakan ke kelas 1, dan f_{10} adalah data dalam kelas 1 yang dipetakan secara salah ke kelas 0 (Prasetyo, 2014).

Berdasarkan isi matriks confusion, maka dapat diketahui jumlah data dari masing-masing kelas yang diprediksi secara benar yaitu (f_{11} + f_{00}) dan data yang diklasifikasikan secara salah yaitu (f_{10} + f_{01}). Kuantitas matriks confusion dapat diringkas menjadi dua nilai, yaitu akurasi dan laju error. Dengan mengetahui jumlah data yang diklasifikasikan secara benar maka dapat diketahui akurasi hasil prediksi, dan dengan mengetahui jumlah data yang diklasifikasikan secara salah maka dapat diketahui laju error dari prediksi yang dilakukan. Dua kuantitas ini digunakan sebagai metric kinerja klasifikasi (Prasetyo, 2014).

Untuk menghitung akurasi digunakan formula sebagai berikut:

$$Akurasi = \frac{Jumlah\ data\ yang\ diprediksi\ secara\ benar}{jumlah\ prediksi\ yang\ dilakukan} \quad \text{.....(2.7)}$$

$$Akurasi = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}} \quad \text{.....(2.8)}$$

Untuk menghitung laju error (kesalahan prediksi) digunakan formula sebagai berikut:

$$Laju\ error = \frac{Jumlah\ data\ yang\ diprediksi\ secara\ salah}{jumlah\ prediksi\ yang\ dilakukan} \quad \text{.....(2.9)}$$

$$Laju\ error = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}} \quad \text{.....(2.10)}$$

Semua algoritma klasifikasi berusaha untuk membentuk model yang mempunyai akurasi yang tinggi (laju error yang rendah). Umumnya model yang dibangun dapat memprediksi dengan benar pada semua data yang menjadi data latihnya, tetapi ketika model berhadapan dengan data uji barulah kinerja model dari sebuah algoritma klasifikasi ditentukan.

2.6.1.2 Sensitivitas dan Spesifisitas

Pengukuran lain yang bisa digunakan sebagai metric kinerja klasifikasi adalah sensitivitas dan spesifisitas. Kedua kuantitas ini memberikan nilai kinerja

yang lebih relevan yang dimaksud. Sensitivitas (atau disebut juga *true positive rate* atau *recall* dalam bidang ilmu pencarian informasi) mengukur proporsi positif asli yang dikenali diprediksi) secara benar sebagai positif (misalnya persentase orang sakit yang diidentifikasi sakit). Sementara spesifisitas (atau disebut juga *true negative rate*) mengukur proporsi negative asli yang dikenali (diprediksi) secara benar sebagai negative (misalnya persentase orang tidak sakit yang diidentifikasi tidak sakit). Dua metric tersebut berhubungan erat dengan konsep error tipe I dan tipe II dalam bidang statistik. Prediksi yang baik memberikan nilai sensitivitas sebesar 100% (misalnya semua orang sakit diidentifikasi sebagai sakit) dan nilai spesifisitas sebesar 100% (misalnya semua orang yang tidak sakit tidak diidentifikasi sebagai sakit) (Prasetyo, 2014).

Pada evaluasi system diagnosis penyakit pasien, setiap orang akan didiagnosis untuk mengetahui apakah menderita penyakit atau tidak. Hasil tes bisa positif (prediksi bahwa orang tersebut sakit) atau *negative* (prediksi bahwa orang tersebut tidak sakit). Hasil pengujian untuk setiap subjek bisa cocok atau tidak cocok dengan status kesehatan yang sebenarnya pada orang tersebut. Artinya ada 4 kelompok prediksi (Prasetyo, 2014).:

True *positive* : Orang sakit yang didiagnosis secara benar sebagai sakit.
False *positive* : Orang sehat yang didiagnosis secara salah sebagai sakit.
True *Negative* : Orang sehat yang didiagnosis secara benar sebagai sehat.
False *Negative*: Orang sakit yang didiagnosis secara salah sebagai sehat.

Tabel 2.1 Matriks confusion sensitivitas dan spesifisitas

		Kelas hasil prediksi	
		Positif	Negatif
Kelas Asli	Positif	True Positive (TP)	False Negative (FN)
	Negatif	False Positif (FP)	True Negative (TN)

Sensitivitas berhubungan dengan kemampuan pengujian untuk mengidentifikasi hasil yang positif dari sejumlah data yang sebenarnya positif. Persamaan yang digunakan untuk menghitungnya disajikan sebagai berikut:

$$Sensitivitas = \frac{TP}{TP+FN} \quad \text{.....(2.11)}$$

Spesifisitas berhubungan dengan kemampuan pengujian untuk mengidentifikasi hasil yang negative dari sejumlah data yang sebenarnya negatif. Persamaan yang digunakan untuk menghitungnya disajikan sebagai berikut:

$$Spesifisitas = \frac{TN}{FP+TN} \quad \text{.....(2.12)}$$

2.6.1.3 Precision dan Recall

Precision (presisi) merupakan metode pengujian dengan melakukan perbandingan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi yang terambil oleh sistem baik yang relevan maupun tidak. *Recall* merupakan metode pengujian yang membandingkan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi relevan yang ada dalam koleksi informasi (baik yang terambil atau tidak terambil oleh sistem) (Prasetyo, 2014).

Persamaan presisi ditunjukkan pada persamaan berikut.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \text{.....(2.13)}$$

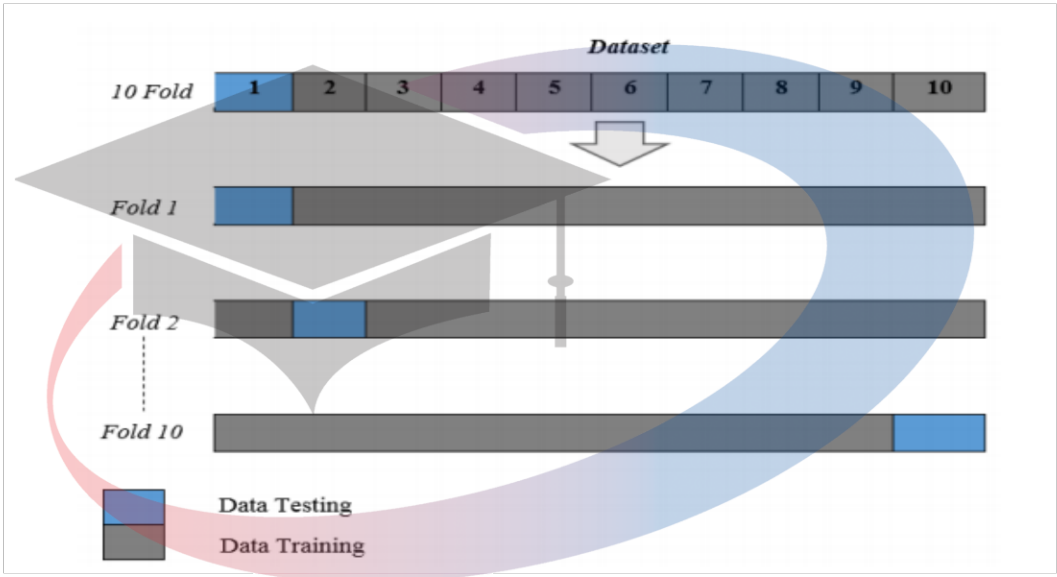
Persamaan recall ditunjukkan pada persamaan berikut.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \text{.....(2.14)}$$

2.6.2 K-Fold Cross Validation (Metode Evaluasi Klasifikator)

Jumlah data memegang peranan penting di dalam algoritma machine learning. Jumlah data yang sedikit (1000 *instance*) namun data itu sendiri tidak mudah untuk diperoleh. Metode yang digunakan untuk mengevaluasi kinerja *classifier* pada penelitian ini adalah k-fold cross validation. K-fold cross validation adalah teknik evaluasi yang dapat digunakan apabila memiliki jumlah data yang terbatas (jumlah instance tidak banyak). K-fold cross validation merupakan salah satu metode yang digunakan untuk mengetahui rata-rata keberhasilan dari suatu sistem dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga sistem tersebut teruji untuk beberapa atribut input yang acak. K-fold cross validation diawali dengan membagi data sejumlah

n-fold yang diinginkan. Dalam proses cross validation data akan dibagi ke dalam n buah partisi dengan ukuran yang sama (D1, D2, D3, ...Dn), selanjutnya proses testing dan training dilakukan sebanyak n kali. Dalam iterasi ke-i, partisi Di akan menjadi data testing dan sisanya akan menjadi data training. Penggunaan jumlah fold terbaik untuk uji validitas, dianjurkan menggunakan10-fold cross validation dalam model (Prasetyo, 2014).



Gambar 2.4 Iterasi pada 10-fold cross validation

Contoh pembagian dataset dalam proses 10-fold cross validation terlihat pada Gambar 2.4. Cara kerja K-fold cross validation adalah sebagai berikut:

1. Total instance dibagi menjadi N bagian.
2. Fold ke-1 adalah ketika bagian ke-1 menjadi data uji (testing data) dan sisanya menjadi data latih (training data). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut. Perhitungan akurasi tersebut dengan menggunakan persamaan sebagai berikut:

$$\text{Akurasi} = \frac{\sum \text{data uji benar klasifikasi}}{\sum \text{total data uji}} \times 100\% \ldots\ldots\ldots\ldots\ldots\ldots (2.15)$$

3. Fold ke-2 adalah ketika bagian ke-2 menjadi data uji (testing data) dan sisanya menjadi data latih (training data). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut. Demikian seterusnya hingga mencapai fold

ke-K. Kemudian, hitung rata-rata akurasi dari K buah akurasi di atas. Rata-rata akurasi tersebut menjadi akurasi final.

2.7 Kredit

2.7.1 Pengertian Kredit

Istilah kredit berasal dari bahasa Yunani "Credere" yang berarti kepercayaan, oleh karena itu dasar dari kredit adalah kepercayaan. Seseorang atau semua badan yang memberikan kredit (kreditur) percaya bahwa penerima kredit (debitur) di masa mendatang akan sanggup memenuhi segala sesuatu yang telah dijanjikan itu dapat berupa barang, uang atau jasa (Thomas, et al., 1998). Kredit yang diberikan oleh bank dapat didefinisikan sebagai penyediaan uang atau tagihan yang dapat dipersamakan dengan itu, berdasarkan persetujuan atau kesepakatan pinjam-meminjam antara bank dengan pihak lain yang mewajibkan pihak peminjam untuk melunasi hutangnya setelah jangka waktu tertentu dengan jumlah bunga, imbalan atau pembagian hasil keuntungan (Taswan, 2003).

Berdasarkan Undang-undang Nomor 10 tahun 1998 tentang Perubahan atas Undang-undang Nomor 7 tahun 1992 tentang Perbankan, yang dimaksud dengan kredit adalah sebagai berikut: "penyediaan uang atau tagihan yang dapat dipersamakan dengan itu, berdasarkan persetujuan atau kesepakatan pinjam-meminjam antara bank dengan pihak lain yang mewajibkan pihak peminjam untuk melunasi utangnya setelah jangka waktu tertentu dengan pemberian bunga" (Sigit Triandaru dan Totok Budisantoso, 2006). Dari beberapa pengertian tentang kredit yang telah dikemukakan oleh para ahli di atas, maka dapat disimpulkan bahwa kredit adalah penyediaan uang atau tagihan yang dapat dipersamakan dengan itu, berdasarkan persetujuan antara pihak bank dengan pihak peminjam dengan suatu janji bahwa pembayarannya akan dilunasi oleh pihak peminjam

sesuai dengan jangka waktu yang telah disepakati beserta besarnya bunga yang telah ditetapkan.

2.7.2 Unsur-unsur Kredit

Kredit yang diberikan oleh suatu lembaga kredit merupakan pemberian kepercayaan. Adapun unsur-unsur yang terkandung dalam pemberian fasilitas kredit menurut Martono (2002:52) adalah sebagai berikut:

1) Kepercayaan

   Kepercayaan merupakan suatu keyakinan pemberi kredit (bank) bahwa kredit yang diberikan berupa uang atau jasa akan benar-benar diterima kembali di masa tertentu di masa mendatang.

2) Kesepakatan

   Kesepakatan dituangkan dalam suatu perjanjian di mana masing-masing pihak menandatangani hak dan kewajiban masing-masing.

3) Jangka waktu

   Setiap kredit yang diberikan pasti memiliki jangka waktu tertentu yang mencakup masa pengembalian kredit yang disepakati.

4) Risiko

   Faktor risiko dapat disebabkan oleh dua hal:

   a. Faktor kerugian yang diakibatkan adanya unsur kesengajaan nasabah untuk tidak membayar kreditnya padahal mampu.
   b. Faktor kerugian yang ditimbulkan oleh unsur ketidaksengajaan nasabah sehingga mereka tidak mampu membayar kreditnya, misalnya akibat terjadi musibah bencana alam.

2.7.3 Manfaat Kredit

Manfaat kredit bagi pihak bank menurut Pudjo Mulyono pada bukunya "Bank Budgeting" (1996: 207) adalah:

1. Sebagai sumber pendapatan yang terbesar berupa bunga. Dengan adanya pendapatan bunga ini memungkinkan setiap bank untuk dapat mengembangkan usahanya, apabila kredit yang diberikan dapat berjalan lancar.
2. Untuk menjaga solvabilitasnya, sebab kredit merupakan salah satu bentuk penyaluran dana bank terbesar. Dengan demikian yang diharapkan dari kredit yang lancar tersebut dapat dipakai sebagai sarana untuk pembayaran kembali dana dan bunga yang dipinjamkan dari masyarakat.
3. Kredit dapat dipakai sebagai alat baik untuk memasarkan produk dan jasa bank yang lain, bahkan saat ini suatu opini (pendapat) yang mengatakan pemberian kredit semata-mata hanya untuk mendapatkan bunga sudah mubadhir.
4. Dengan menyalurkan dana akan mampu mengembangkan para stafnya untuk mengenal dunia bisnis yang lain.

2.7.4 Prinsip-prinsip Perkreditan

Prinsip perkreditan disebut juga sebagai konsep 6C (Martono, 2002). Pada dasarnya konsep 6C ini akan dapat memberikan informasi mengenai tekad baik dan kemampuan membayar nasabah untuk melunasi kembali pinjaman beserta bunganya. Prinsip 6C tersebut antara lain adalah:

1. *Character* Penilaian character ini dapat mengetahui sejauh mana tingkat kejujuran dan tekad baik calon debitur yaitu kemauan untuk memenuhi kewajiban-kewajiban dari calon debitur.
2. *Capacity* Penilaian *capacity* untuk melihat kemampuan dalam melunasi kewajibannya dari kegiatan usaha yang dilakukan atau kegiatan usaha yang akan dilakukan yang dibiayai dengan kredit dari bank.
3. *Capital* Penilaian terhadap prinsip capital tidak hanya melihat besar kecilnya modal yang dimiliki oleh calon debitur tetapi juga bagaimana distribusi modal itu ditempatkan.
4. *Collateral* diartikan sebagai jaminan fisik harta benda yang bernilai uang dan mempunyai harga stabil dan mudah dijual. Jika pada dari peminjam

terkena kecelakaan atau hal-hal lain yang mengakibatkan peminjam tidak mampu membayar hutangnya, maka tindakan akhir yang dilakukan oleh bank adalah melaksanakan haknya atas *collateral* yang diikat secara yuridis untuk menjamin hutangnya pada bank.

5. *Condition of Economy* Pada prinsip condition (kondisi), dinilai situasi dan kondisi politik, sosial, ekonomi, dan kondisi pada sektor usaha calon debitur. Maksudnya agar bank dapat memperkecil risiko yang mungkin timbul oleh kondisi ekonomi, keadaan perdagangan dan persaingan di lingkungan sektor usaha calon debitur dapat diketahui.
6. *Constraint* untuk menilai budaya atau kebiasaan yang tidak memungkinkan seseorang melakukan bisnis di suatu tempat. Masalah constraint ini agak sukar dirumuskan karena tidak ada peraturan tertulis mengenai hal tersebut, dan juga tidak dapat selalu didefinisikan secara fisik permasalahannya.

2.7.5 Kebijaksanaan Perkreditan

Dalam menetapkan kebijaksanaan perkreditan tersebut harus diperhatikan 3 (tiga) asas pokok yaitu (Muljono, 2007):

1. Asas likuiditas, Asas likuiditas adalah suatu asas yang mengharuskan bank untuk tetap dapat menjaga tingkat likuiditasnya, karena suatu bank yang tidak likuid akibatnya akan sangat parah yaitu hilangnya kepercayaan dari para nasabahnya atau dari masyarakat luas. Suatu bank dikatakan likuid apabila memenuhi kreteria antara lain:
    a. Bank tersebut memiliki cash assets sebesar kebutuhan yang akan digunakan untuk memenuhi likuiditasnya.
    b. Bank tersebut memiliki assets lainnya yang dapat dicairkan sewaktu-waktu tanpa mengalami penurunan nilai pasarnya.
    c. Bank tersebut mempunyai kemampuan untuk menciptakan *cash assets* baru melalui berbagai bentuk utang.

2. Asas solvabilitas, Asas solvabilitas usaha pokok perbankan yaitu menerima simpanan dana dari masyarakat dan disalurkan dalam bentuk kredit.
3. Asas rentabilitas, Asas rentabiltas sebagaimana halnya pada setiap kegiatan usaha akan selalu mengharapkan untuk memperoleh laba, baik untuk mempertahankan eksistensinya maupun untuk keperluan mengembangkan dirinya.

2.7.6 Penggolongan Kolektibilitas Kredit

Dalam kenyataan tidak semua kredit yang telah diberikan dapat berjalan lancar, sebagian ada yang kurang lancar dan sebagian menuju kemacetan. Demi amannya suatu kredit, maka perlu diambil langkah-langkah untuk mengklasifikasikan kredit berdasarkan kelancarannya. Hal ini sangat diperlukan untuk melakukan tugas-tugas pengendalian kredit agar dapat berjalan dengan lancar. Keadaan pembayaran pokok atau angsuran pokok dan bunga pinjaman oleh nasabah, terlihat pada tata usaha bank dan hal ini merupakan kolektibilitas dari kredit. Informasi dari tingkat kolektibilitas akan sangat bergantung bagi bank untuk kegiatan pengawasan terhadap masing-masing nasabah secara individu maupun secara keseluruhan. Kolektibilitas adalah suatu pembayaran pokok atau bunga pinjaman oleh nasabah sebagaimana terlihat tata usaha bank berdasarkan Surat Keputusan Direksi Bank Indonesia (BI) No. 32/268/KEP/DIR tanggal 27 Pebruari 1998, maka kredit dapat dibedakan menjadi:

1. Kredit lancar

   Kredit lancar yaitu kredit yang pengembalian pokok pinjaman dan pembayaran bunganya tepat waktu, perkembangan rekening baik dan tidak ada tunggakan serta sesuai dengan persyaratan kredit. Kredit lancar mempunyai kriteria sebagai berikut:

   1) Pembayaran angsuran pokok dan bunga tepat waktu.
   2) Memiliki mutasi rekening yang aktif.
   3) Bagian dari kredit yang dijamin dengan uang tunai.
2. Kredit kurang lancar

Yaitu kredit yang pengembalian pokok pinjaman atau pembayaran bunganya terdapat tunggakan telah melampaui 90 hari sampai 180 hari dari waktu yang telah disepakati. Kredit kurang lancar mempunyai kriteria sebagai berikut:

1) Terdapat tunggakan angsuran pokok dan bunga yang telah melampaui 90 hari.
2) Frekuensi mutasi rendah.
3) Terjadi pelnggaran terhadap kontrak yang telah dijanjikan lebih dari 90 hari.
4) Terjadi mutasi masalah keuangan yang dihadapi debitur.
5) Dokumentasi pinjaman lemah.

3. Kredit diragukan yaitu kredit yang pengembalian pokok pinjaman dan pembayaran bunganya terdapat tunggakan yang telah melampaui 180 hari sampai 270 hari dari waktu yang disepakati. Kredit diragukan memiliki kriteria sebagai berikut:
   1) Terdapat tunggakan angusuran pokok atau bunga yang telah melampaui 180 hari.
   2) Terjadinya wanprestasi lebih dari 180 hari.
   3) Terjadi cerukan yang bersifat permanen.
   4) Terjadi kapitalisasi bunga.
   5) Dokumentasi hukum yang lemah baik untuk perjanjian maupun pengikat pinjaman.
4. Kredit macet yaitu kredit yang pengembalian pokok pinjaman dan pembayaran bunganya terdapat tunggakan telah melampaui 270 hari. Kredit macet mempunyai kriteria sebagai berikut:
   1) Terdapat tunggakan angsuran pokok yang telah melampaui 270 hari.
   2) Kerugian operasional dituntut dengan pinjaman baru.
   3) Jaminan tidak dapat dicairkan pada nilai wajar, baik dari segi hukum maupun dari segi kondisi pasar.

Faktor-faktor penyebab kredit macet adalah sebagai berikut (Kuncoro dan Suhardjono, 2002):

a. Faktor eksternal bank
    1) Adanya maksud tidak baik dari para debitur yang diragukan.
    2) Adanya kesulitan atau kegagalan dalam proses likuiditas dari perjanjian kredit yang telah disepakati antara debitur dengan bank.
    3) Kondisi manajemen dan lingkungan usaha debitur.
    4) Musibah (misalnya: kebakaran, bencana alam) atau kegagalan usaha.
b. Faktor internal bank
    1) Kurang adanya pengetahuan dan keterampilan para pengelola kredit.
    2) Tidak adanya kebijakan perkreditan pada bank yang bersangkutan.
    3) Pemberian dan pengawasan kredit yang dilakukan oleh bank menyimpang dari prosedur yang telah ditetapkan.
    4) Lemahnya organisasi dan manajemen dari bank yang bersangkutan.

2.8 Penelitian Sebelumnya

Salah satu penelitian mengenai penerapan seleksi fitur menggunakan PSO-KNN adalah penelitian pada masalah diagnosis pengelompokan mikro kalsifikasi dalam mamografi (Zyout, 2011). Dalam penelitian tersebut menggunakan dataset yang terdiri dari 34 fitur. Dari penelitian yang telah dilakukan, hasil akurasi KNN yang dioptimasi lebih tinggi dari pada tanpa optimasi dengan peningkatan akurasi tertinggi mencapai 38%, yaitu dari 56% menjadi 94%.

Danenas & Garsva (2012) melakukan studi tentang pemodelan evaluasi risiko kredit menggunakan pengklasifikasi linear Vector Support Machines (SVM), dikombinasikan dengan pemilihan parameter evolusi menggunakan Genetic Algorithms dan Particle Swarm Optimization, dan pendekatan *sliding window*. Analisis diskriminan diterapkan untuk evaluasi contoh keuangan dan pembentukan dinamis kelas kebangkrutan. Kemungkinan aplikasi pemilihan fitur juga diteliti dengan menerapkan evaluator subset fitur berbasis korelasi. Hasil penelitian menunjukkan kemungkinan untuk mengembangkan dan menerapkan pengklasifikasi cerdas berdasarkan metode analisis diskriminan asli evaluasi dan menunjukkan bahwa PSO sangat cocok dikombinasikan dengan SVM.

Li, et al., (2013) mengusulkan metode untuk evaluasi kredit pribadi berdasarkan PSO-RBF neural jaringan, yang menggunakan algoritma PSO untuk mengoptimalkan parameter jaringan saraf RBF, kemudian menerapkan RBF yang dioptimalkan jaringan saraf dalam evaluasi kredit pribadi. Metode ini menggabungkan kemampuan pencarian global algoritma PSO dan efektivitas tinggi optimalisasi RBF lokal bersama-sama, mengatasi algoritme PSO yang tidak stabil dan kelemahan RBF yang dengan mudah mengarah ke minimum lokal. Hasilnya menunjukkan bahwa metode penilaian kredit pribadi didasarkan pada jaringan saraf PSO-RBF sangat akurat dalam klasifikasi dan prediksi, dan cocok untuk penilaian dan prediksi kredit pribadi, yang berarti bahwa PSO juga cocok dikombinasikan dengan Jaringan Syaraf Tiruan. Hasil yang diperoleh pada model-model tersebut mengindikasikan perbaikan akurasi melalui penerapan bahwa PSO dalam mengoptimasi parameter.

2.9 Kerangka Pikir Pemecahan Masalah

Berdasarkan konsep dasar Fk-NN yaitu memberikan teknik klasifikasi yang sederhana tetapi memiliki hasil kerja yang cukup bagus sehingga diharapkan metode ini dapat memberikan klasifikasi yang sederhana, mudah, cepat dan optimal (Dyah, 2015). Model klasifikasi risiko kredit adalah menyediakan pemisahan antara peminjam yang berpotensi gagal dengan yang tidak gagal dalam hal pembayaran kredit dengan jangka waktu yang telah disepakati. Identifikasi dini dan klasifikasi risiko kredit merupakan hal yang sangat penting agar dapat menghindari kondisi yang memungkinkan proses pembayaran menjadi tidak lancar atau disebut juga dengan istilah kredit macet (Ivandari, 2017).

Pada MPSO (*Modified Particle Swarm Optimization*) digunakan untuk meningkatkan akurasi dan fungsionalitas dalam optimasi paramater Fk-NN. Untuk mempelajari model klasifikasi berdasarkan data yang dikumpulkan secara terpusat kepada server, namun pengumpulan data secara terpusat dapat menimbulkan masalah hal kebocoran privasi data para peserta peminjam kredit karena dapat terjadi kesalahan selama proses klasifikasi. Pada penelitian ini, kami fokus untuk mengatasi masalah risiko kredit untuk klasifikasi Fk-NN dan MPSO

dengan mengoptimalkan proses klasifikasi risiko kredit, dengan menggunakan dataset kredit German yang diterbitkan oleh UCI repository.

Solusi yang ditawarkan pada penelitian ini adalah membangun model untuk klasifikasi risiko kredit menggunakan Fk-NN dengan MPSO dan fungsi K-fold cross validation. Fungsi K-fold cross validation yaitu melakukan validasi atau mengevaluasi kinerja *classifier*, hasil klasifikasi untuk pemilihan nilai *m* dan *k*. Berdasarkan solusi yang ditawarkan, maka dilakukan pengukuran kesamaan antara data dengan menggunakan euclidean distance, fungsi dari euclidean distance adalah untuk mendapatkan hasil perhitungan jarak yang maksimal, dengan mengurangi nilai error dan waktu komputasi, sehingga model yang diusulkan dapat digunakan dalam mengoptimalkan akurasi model klasifikasi pada risiko kredit yang lebih baik.