BAB 2

KAJIAN LITERATUR

2.1 Tinjauan Pustaka

Pada subbab ini, akan dijelaskan tinjauan pustaka yang berkaitan dengan penelitian yang akan dilakukan.

2.1.1 Data Mining

Dalam proses pengolahan data menjadi informasi, *data mining* sangat diperlukan yang kemudian akan menghasilkan suatu pengetahuan baru yang bersumber dari data yang lama, dimana hasil keputusan tersebut dapat digunakan sebagai acuan dalam pengambilan keputusan dimasa yang akan datang. *Data mining* melibatkan penemuan pola baru, menarik, dan berpotensi berguna dari kumpulan data besar dan menerapkan algoritma untuk ekstraksi informasi tersembunyi. Banyak istilah lain yang digunakan untuk *data mining*, misalnya, penemuan pengetahuan (mining) dalam *database* (KDD), ekstraksi pengetahuan, analisis data/pola, arkeologi data, pengerukan data, dan pengumpulan informasi(Han et al., 2011).

Tujuan dari setiap proses *data mining* adalah untuk membangun model prediktif atau deskriptif yang efisien dari sejumlah besar data yang tidak hanya paling cocok atau menjelaskannya, tetapi juga dapat menggeneralisasi ke data baru (Mukhopadhyay et al., 2013). Berdasarkan tampilan luas dari fungsionalitas *data mining*, *data mining* adalah proses menemukan pengetahuan yang menarik dari sejumlah besar data yang disimpan baik di *database*, gudang data, atau repositori informasi lainnya.

Berdasarkan definisi *data mining* dan definisi fungsi *data mining*, umumnya proses *data mining* mencakup langkah-langkah berikut(Chen et al., 2015):

1. Persiapan data: siapkan data untuk penambangan. Ini mencakup 3 langkah berikut: mengintegrasikan data di berbagai sumber data dan membersihkan

gangguan dari data; mengekstrak beberapa bagian data ke dalam sistem *data mining*; praproses data untuk memfasilitasi *data mining*.

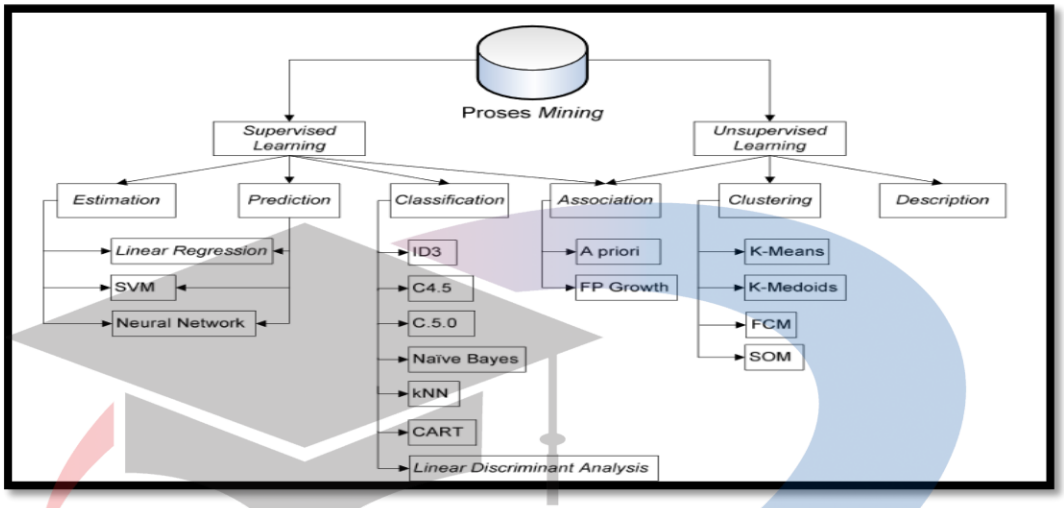
2. Penambangan data: menerapkan algoritma pada data untuk menemukan pola dan mengevaluasi pola pengetahuan yang ditemukan.
3. Penyajian data: memvisualisasikan data dan merepresentasikan pengetahuan yang ditambang kepada pengguna.

Kita bisa melihat data mining dalam tampilan multidimensi(Han et al., 2011), yaitu:

1. Dalam tampilan pengetahuan atau tampilan fungsi data mining, ini mencakup karakterisasi, diskriminasi, klasifikasi, pengelompokan, analisis asosiasi, analisis deret waktu, dan analisis pencilan.
2. Dalam tampilan teknik yang digunakan, itu termasuk pembelajaran mesin, statistik, pengenalan pola, data besar, mesin vektor pendukung, himpunan kasar, jaringan saraf, dan algoritma evolusioner.
3. Dalam tampilan aplikasi, ini mencakup industri, telekomunikasi, perbankan, analisis penipuan, penambangan biodata, analisis pasar saham, penambangan teks, penambangan web, jaringan sosial, dan *e-commerce*.

Konsep pembelajaran dalam *Data mining* terbagi menjadi 2 macam konsep pembelajaran(Oded & 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.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)

2.1.2 Unsupervised Learning

Unsupervised Learning merupakan teknik pembelajaran mesin yang berusaha untuk melakukan representasi pola sebuah input yang berasal dari datalatihan 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(Shalev-Shwartz & Ben-David, 2014). Beberapa algoritma *Unsupervised Learning* diantaranya adalah: *K-Means Clustering*, *Hierarchical Clustering*, dan *Fuzzy* C-Means.

2.1.2.1 K-Means Clustering

K-Means merupakan salah satu metode pengelompokan data *nonhierarki* (sekatan) yang berusaha mempartisi data yang ada ke dalam bentuk dua atau lebih kelompok. Adapun tujuan pengelompokkan data ini adalah untuk meminimalkan fungsi objektif yang diatur dalam proses pengelompokan, yang pada umumnya berusaha meminimalkan variasi di dalam suatu kelompok dan memaksimalkan variasi antar kelompok. *K-Means* adalah salah satu algoritma yang menggunakan metode partisi. *K-Means* adalah algoritma *clustering* yang membagi masing-masing item data ke dalam satu *cluster*. *K-Means* adalah suatu teknik pengelompokkan data yang mana keberadaan tiap-tiap titik data dalam suatu klaster ditentukan oleh derajat keanggotaan(Fränti & Sieranoja, 2018).Dari beberapa definisi di atas dapat disimpulkan bahwa *K-Means Clustering* adalah metode pengelompokan yang mempartisi data kedalam kelompok yang memiliki karakteristik sama dimasukkan ke dalam kelompok yang sama dimana set data yang dimasukkan ke dalam kelompok tidak tumpang tindih.

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/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 R^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 dataset 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 digunakan untuk ketidakmiripan adalah *Euclidean*(CHANDRA &

Hermadi, 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.Untuk menentukan nilai pusat (*centroid*) pada tahap iterasidigunakan rumus persamaan (1) sebagai berikut:

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

Dimana:

Vij : *centroid* rata-rata *cluster* ke-i untuk variable *k-j*

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

i, k :indeks dari *cluster*

j : indeks dari variabel

X_{kj} : nilai data ke-*k* variable ke-*j* dalam *cluster* tersebut

Menurut Prasetyo (2014) untuk menentukan korelasi antar dua obyek yaitu dengan menggunakan rumus *Euclidean Distance*seperti persamaan (2) berikut:

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

Dimana:

d(x,y) :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*

2.1.2.2 Hierarchical Clustering

Dalam statistik, pengelompokan berbasis hirarki adalah metode analisis *cluster* yang berusaha untuk membangun sebuah *hiearki cluster*. Strategi untuk pengelompokan berbasis hirarki umumnya jatuh ke dalam dua jenis, yaitu *aglomeratif* dan *divisive*. *Aglomeratif* merupakan metode pengelompokan berbasis hirarki 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, 2014). Proses tersebut diulang terus sehingga tampak bergerak ke atas membentuk hirarki. Cara ini membutuhkan suatu parameter kedekatan *cluster* (*cluster proximity*).

Divisif merupakan metode pengelompokan berbasis hirarki 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):

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 minimal2 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.

2.1.2.3 Fuzzy C-Means

Teknik ini pertama kali diperkenalkan oleh Jim Bezdek pada tahun 1981. *Fuzzy Cluster Means* (FCM) merupakan algoritma yang digunakan untuk melakukan clustering data sesuai berdasarkan keberadaan tiap-tiap titik data sesuai dengan derajat keanggotaannya. Algoritma ini merupakan salah satu teknik *soft clustering* yang paling popular dengan menggunakan pendekatan data *point* dimana titik pusat*cluster* akan selalu diperbaharui sesuai dengan nilai keanggotaan dari datayang ada dan selain itu algoritma *Fuzzy C-Means* juga merupakan algoritmayang bekerja dengan menggunakan model *fuzzy* sehingga memungkinkansemua data dari semua anggota kelompok terbentuk dengan derajat keanggotaan yang berbeda antara 0 dan 1 Metode *Fuzzy C-Means* pada dasarnya memiliki tujuan meminimalisasikan fungsi serta mendapatkan pusat *cluster* yang nantinyaakan digunakan untuk mengetahui data yang masuk ke dalam sebuah*cluster*(Bora & Gupta, 2014).

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. Beberapa contoh dalam penerapan *Fuzzy C-Means* adalah masalah pengelompokan data nyata yang telah dibuktikan dengan menghasilkan karateristik data yang baik(Kaushik & Hemanta, 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*.

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 berbedaatau *value* yang berbeda dalam satu atau beberapa variabel
4. Penentuan titik *cluster* yang optimal
5. Dapat melakukan *clustering* lebih dari satu variabel secara sekaligus.

Beberapa kelemahan yang dimiliki oleh algoritma *Fuzzy C-Means* yaitu(Bora & Gupta, 2014):

1. Pada algoritma *Fuzzy C-Means* user memerlukan lebih banyak waktu untuk proses perhitungan komputasinya dalam menentukan *cluster*pada setiap anggota di suatu dataset.
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 *cluster* pada pengelompokan data.

2.1.3 *Supervised Learning*

Supervised Learning merupakan teknik pembelajaran mesin yang membuat suatu fungsi berdasarkan data latihan yang sudah ada, dalam hal

inidapat 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(Shalev-Shwartz & Ben-David, 2014). Beberapa algoritma *Supervised Learning* diantaranya adalah *Decision tree*, *K- Nearest Neighbor Classifier*, *Naive Bayes Classifier*, *Artificial Neural Network*, dan *Support Vector Machine*.

2.1.3.1 Decision Tree

Decision Tree merupakan salah satu algoritma yang paling umum digunakan dalam aplikasi penilaian kredit(Shetty & Manoj, 2019). Algoritma *Decision Tree* menggunakan struktur pohon 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.

Karakteristik dari *decision tree* dibentuk sejumlah elemen sebagai berikut(Zhang, 2020):

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.

2.1.3.2 Artificial Neural Network

Neural Network telah berhasil diterapkan pada tugas klasifikasi praktis di berbagai industri, termasuk bidang industri, komersial, dan ilmiah. *Neural Network* terdiri dari neuron yang dapat memproses informasi. Interkoneksi antar neuron merupakan model algoritma yang meliputi input layer, *hidden layer*, dan *output layer*(Hewahi & Hamra, 2017). Saat menggunakan *Artificial Neural Network*(ANN) untuk menyelesaikan masalah klasifikasi, misalnya, kita dapat menerapkan fungsi aktivasi sigmoid, yang sering digunakan sebagai fungsi ambang jaringan saraf untuk memetakan variabel antara 0 dan 1, untuk menyelesaikan tugas klasifikasi.Mhatre et al.(2017) mengusulkan agar jaringan saraf stabil karena ketika elemen jaringan saraf gagal, ia dapat melanjutkan sifat paralelnya tanpa masalah. Namun untuk pemrosesan data yang besar, dibutuhkan waktu pemrosesan yang lebih lama.

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, 2014). 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 digunakanuntuk 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 *layerinput*, *layer* tersembunyi dan *layer output*.

Klasifikasi *Backpropagation* 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., 2011).

2.1.3.3 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 *hyperplanelinear* yang optimum. Dengan melakukan *mappingnonlinear* 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). SVM dapat memecahkan masalah *nonlinier* dan dapat menangani kumpulan data berdimensi tinggi, tetapi interpretasinya tidak kuat. Dengan adanya sekumpulan contoh pelatihan, setiap contoh pelatihan ditandai sebagai milik satu atau yang

lain dari dua kategori, SVM membuat model yang menetapkan contoh baru ke salah satu dari dua kategori, menjadikannya pengklasifikasi linier biner *non-probabilistik*(Zhang, 2020).

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.

2.1.3.4 *Naive Bayes Classifier*

Klasifikasi *naïve bayesian* didasarkan pada teorema *Bayes* dan mengasumsikan bahwa kondisi fitur tidak bergantung satu sama lain(Merikoski et al., 2018). Untuk set pelatihan, mengambil independensi antara fitur sebagai premis, melalui model pembelajaran, input X menemukan output Y yang memaksimalkan *probabilitas posterior*. *Naïve bayesian* dapat menangani beberapa masalah klasifikasi tetapi memiliki tingkat kesalahan tertentu karena menentukan klasifikasi berdasarkan data sebelumnya untuk menentukan probabilitas *posterior*(Zhang, 2020).

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. Teorematersebut 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 & Mammadov, 2015).

Persamaan dari teorema *Bayes* adalah(Taheri & Mammadov, 2015):

$$P(H|X) = \frac{P(X|H) \times P(H)}{P(X)} \quad (3)$$

Keterangan:

X : Data dengan kelas yang belum diketahui

H : Hipotesis data X merupakan suatu kelas 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 (Taheri & Mammadov, 2015):

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.

2.1.3.5 *k–Nearest Neighbor*

Metode *k–Nearest Neighbor* (kNN) 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 25 besar. Baru pada saat teknologi komputasi semakin maju, metode ini mulai banyak digunakan terutama pada bidang pattern recognition(Han et al., 2011). Penggunaan kNN untuk klasifikasi memiliki berbagai keunggulan, yaitu lebih intuitif dan mudah diimplementasikan, apalagi interpretabilitas kNN yang kuat sangat penting untuk membangun model prediksi kredit (Zhang, 2020). Selain sangat sederhana, kuat, mudah diterapkan dan dipahami, kNN sangat berguna karena tidak melibatkan asumsi apa pun tentang data, ditambah lagi ukuran jarak yang dapat dihitung secara konsisten antara dua contoh (Kiran et al., 2018).

Algoritma kNN intuitif dan mudah diterapkan. Langkah pertama adalah memilih fungsi jarak, beberapa pilihan diantaranya yaitu jarak *Euclidean*, jarak *Manhattan*, jarak *Cosine*, jarak *Mahalanobis*, dan seterusnya. Jarak *Euclidean* banyak digunakan dalam tugas klasifikasi berbasis jarak(Sainin & Alfred, 2010). Pada langkah kedua, jarak antara data yang tidak berlabel dan semua tetangga diurutkan dari kecil ke besar, lalu *k* atas dipilih. Langkah ketiga adalah menghitung jumlah kemunculan setiap kategori di *k* tetangga dan mengklasifikasikan data yang tidak berlabel sebagai kategori dengan kemunculan terbanyak.

Metode kNN menjadi salah satu metode berbasis NN yang paling tua dan popular. 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. 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 (Zhang, 2020).

Dalam algoritma ini, tidak ada rumus baku untuk menentukan nilai k yang paling optimal. Beberapa penelitian merekomendasikan beberapa cara untuk menentukannya.Han et al.(2011)menyarankan pemilihan nilai k dengan metode eksperimen, dengan memulai eksperimen dari nilai $k = 1, 2, 3,$ dst., lalu dipilih nilai k yang menghasilkan akurasi terbaik. 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 kNN, 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.

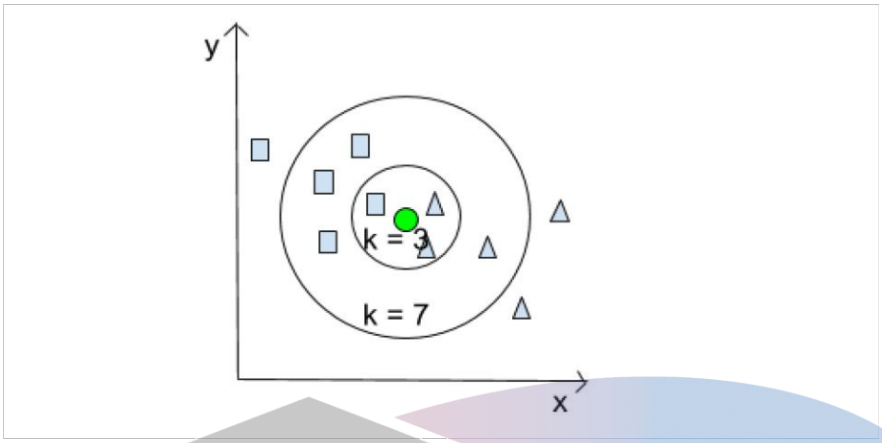
Algoritma KNN mudah diimplementasikan dan dipahami, dan berikut ini adalah proses implementasi spesifiknya:

1. Tetapkan satu set sampel pengujian dan satu set sampel pelatihan.
   Kumpulan sampel pelatihan dinyatakan sebagai:$X = \{(x_i, c_i) \mid i = 1, 2, \dots, n\}$. Pada rumus: $x_i = \left(x_i^1, x_i^2, \dots, x_i^l\right)$ merupakan seuah vector berdimensi l, yang menyatakan banyak fitur yaitu l. x_i^jmewakili nilai komponen fitur ke-j dari sampel data *training* ke-i. c_i mewakili kategori yang sesuai dari sampel data ke-i, c_i termasuk dalam kategori label C, C = $\{1, 2, \dots, m\}$, dimana m adalah banyaknyakategori/kelas.
   Set sampel pengujian dinyatakan sebagai: $Y = \{y_j \mid j = 1, 2, \dots, n\}$.

Pada rumus: $y_j = \left(x_j^1, x_j^2, \ldots, x_j^l\right)$ merupakan seuah vector berdimensi l, yang menyatakan banyak fitur yaitu l. y_j^i mewakili nilai komponen fitur ke-i dari sampel data *training* ke-j.

2. Tetapkan nilai k. Penentuan nilai k diatur berulang-ulang sesuai dengan efek klasifikasi pada percobaan hingga didapatkan nilai k yang optimal. Kami menggunakan validasi silang untuk menentukan nilai k, yang merupakan metode yang mengevaluasi dan memilih parameter.
3. Hitung jarak antara titik sampel pengujian dan semua titik sampel pelatihan dengan menggunakan fungsi jarak *Euclidean*.
4. Pilih k sampel pelatihan tetangga terdekat. Untuk titik sampel pengujian y, k titik sampel pelatihan yang terdekat dengan titik sampel pengujian y dalam set sampel pelatihan ditemukan sesuai dengan fungsi jarak *Euclidean*, fungsi jarak lainnya sebagai alternatif.
5. Aturan diskriminan untuk kategori y sampel pengujian. Yaitu melakukan statistik pada titik sampel pelatihan tetangga terdekat yang diperoleh pada langkah 4, menghitung jumlah masing-masing kategori titik sampel pelatihan k, dan mengelompokkan kategori sampel uji ke dalam kategori sampel pelatihan dengan jumlah terbesar.

Pada ruang dua dimensi, contoh visualnya ditunjukkan pada Gambar 2.2, jika k sama dengan 3, lingkaran hijau diklasifikasikan sebagai segitiga, tetapi jika k sama dengan 7, lingkaran hijau diklasifikasikan sebagai segi empat. Jarak *Euclidean* disiratkan oleh lingkaran.



Gambar 2.2. Contoh 2-dimensi dari algoritma KNN (Zhang, 2020)

Saat menerapkan fungsi jarak *Euclidean*, kita perlu menskalakan data, karena hasilnya akan dipengaruhi oleh unit variabel. Matriks jarak *Euclidean* adalah matriks jarak kuadrat antar titik, yang telah digunakan dalam pembelajaran mesin, jaringan sensor nirkabel, akustik, dan bidang lainnya(Dokmanic et al., 2015). D'Agostino & Dardanoni(2009) menyebutkan rumus jarak Euclidean dalam ruang berdimensi-n yang merepresentasikan jarak sebenarnya antara dua titik dalam ruang berdimensi n, yaitu jarak linier dari titik ke titik, yaitu:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (4)$$

2.1.4 Seleksi Fitur

Pemilihan fitur dapat disebut sebagai proses pemilihan *subset* fitur yang optimal berdasarkan kriteria tertentu. Pemilihan fitur dan pemilihan variabel sangat penting untuk penemuan pengetahuan dari sejumlah besar data. Metode pemilihan fitur secaraklasik melibatkan masalah pengoptimalan kombinatorial yaitu *NP-hard*. Waktu komputasi mereka meningkat seiring dengan dimensinya. Karena biaya komputasi yang tinggi ini, prosedur tradisional ini menjadi tidak layak untuk analisis data berdimensi tinggi. Dengan demikian, prosedur pemilihan

variabel yang lebih inovatif sangat penting untuk menangani data berdimensi tinggi. Pemilihan fitur memiliki peran penting dalam teknik *preprocessing* data yang digunakan untuk data mining (Han et al., 2011).

Metode pemilihan fitur dapat berupa *wrapper-based*, *filter-based*atau juga model disematkan. Model klasifikasi yang mengandalkan pendekatan berbasis filter (*filter-based*) melakukan pemilihan fitur selama tahap *preprocessing* tetapi mengabaikan algoritma pembelajaran. Model *wrapper-based* melakukan proses pemilihan fitur dengan mempertimbangkan setiap subset yang mungkin, dan *subset* fitur diberi peringkat sesuai dengan kekuatan prediktifnya. Sedangkan model yang disematkan, memilih fitur sambil mempertimbangkan desain pengklasifikasinya (Nayar et al., 2019).

2.1.4.1 Seleksi Fitur menggunakan *Swarm Intelegent*

Swarm Intelegence (SI) atau kecerdasan kawanan merupakan bagian integral dalam bidang *Artificial Intelegence* (AI) atau kecerdasan buatan. SI secara bertahap menjadi terkenal, karena semakin banyak masalah kompleksitas yang tinggi membutuhkan solusi yang mungkin kurang optimal tetapi dapat dicapai dalam waktu yang wajar periode waktu. Sebagian besar terinspirasi oleh sistem biologis, SI mengadopsi perilaku kolektif dari kelompok hewan yang terorganisir, saat mereka berusaha untuk bertahan hidup (Chakraborty & Kar, 2017).

Baru-baru ini, banyak algoritma SI telah diterapkan pada pemilihan fitur, dan tiga algoritma yang paling populer diantaranya yaitu algoritma*Particle Swarm Optimization* (PSO), *Artificial Bee Colony* (ABC), dan *Ant Colony Optimization* (ACO). Untuk setiap algoritma, ada dua cara utama untuk merepresentasikan pemilihan fitur, yaitu representasi standar (seringkali sebagai vektor kontinu) dan representasi biner yang disesuaikan untuk pemilihan fitur, dimana mekanisme pencarian sangat bergantung pada representasi(Nguyen et al., 2020).

2.1.4.2 Seleksi Fitur Menggunakan PSO

PSO adalah sebuah teknik *stochastic optimization* berdasarkan populasi (ikan, lebah, burung dll), dikemukakan oleh Russell C. Eberhart dan James

Kennedy di tahun 1995 yang terinspirasi oleh perilaku sosial dari pergerakan burung atau ikan.PSO telah sukses diterapkan di dalam berbagai bidang penelitian dan banyak aplikasi, termasuk aplikasi yang spesifik dengan kebutuhan yang spesifik pula, seperti: optimasi fungsi, permainan sudoku, pengontrolan sistem fuzzy, termasuk "pelatihan" *Artificial Neural Networks* (ANN), menyelesaikan persoalan rantai suplay (Habibi, 2017)dan banyak aplikasi lainnya. Hal ini disebabkan karena PSO memiliki metode penyelesaiaan masalah dengan cepat dan sederhana serta memberikan hasil yang lebih baik bila dibandingan dengan metode lain.

PSO bersimulasi dengan perilaku dari sekawanan burung. Seperti skenario berikut: ada sekelompok burung yang secara acak mencari makanan di suatu daerah, dimana hanya ada satu potong makanan di daerah yang dicari. Semua burung tidak tahu seberapa jauh keberadaan makanan tersebut. Maka strategi yang paling baik untuk menemukan makanan adalah mengikuti burung yang berada paling dekat dengan makanan. PSO mengadopsi skenario tersebut dan menerapkannya untuk memecahkan masalah optimasi.

Dalam PSO, setiap satu solusi yang dimaksud dengan "burung" dalam pencarian ruang kita sebut dengan "partikel" (atau individu). Setiap partikel "terbang" mengikuti individu-individu yang optimum saat ini (*current optimum particles*). Partikel menyimpan jejak-jejak posisinya dalam *problem space*. Jejak-jejak posisi tersebut diartikan sebagai *best solution*, atau *fitness* dalam GA yang telah diperolehnya sejauh ini. Nilainya, yakni *fitness value*, yang disebut *pbest* juga turut disimpan. Selain *pbest* yang merupakan milik individu yang bersangkutan, turut disimpan pula nilai terbaik milik individu di sekitarnya (*local best*), yang disebut *lbest*. Jika suatu individu memperhitungkan semua individu di dalam populasi dimana dia berada sebagai individu di sekitarnya, maka nilai terbaik yang dimaksud adalah nilai terbaik secara keseluruhan (*global best*) dan disebut *gbest*. Selanjutnya, terjadi akselerasi antara lokasi *pbest* dan lokasi *pbest* dari setiap individu. Akselerasi ini diberi bobot berupa bilangan acak. Diagram alir dari PSO dapat dilihat pada Gambar 2.3.

Setelah menemukan dua nilai terbaik, pembaruan partikel kecepatan dan posisi dengan persamaan berikut (Tuegeh et al., 2009):

$$v_{ij}^{k+1} = \omega_k * v_{ij}^k + c_1 * rand * \left(pbest_{ij}^k - x_{ij}^k\right) + c_2 * rand * \left(gbest_{ij}^k - x_{ij}^k\right) \quad (5)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (6)$$

dimana:

v_{ij}^{k+1} : adalah kecepatan partikel

x_{ij}^k : adalah posisi partikel saat ini (solusi)

$pbest_{ij}^k$: nilai terbaik milik individu di sekitarnya

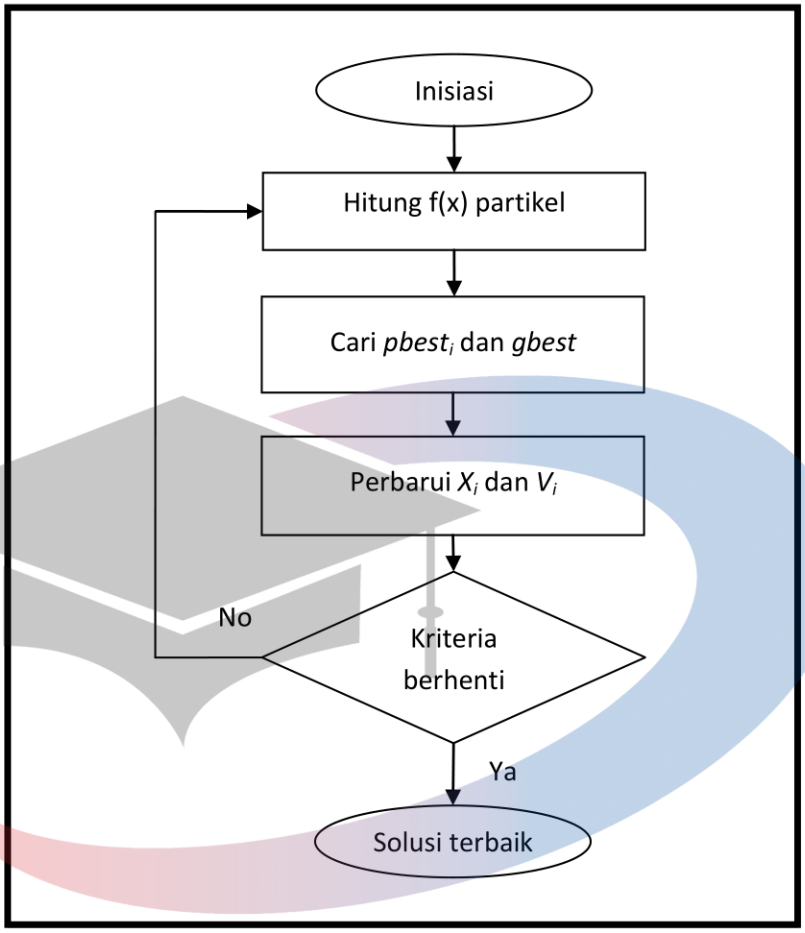
$gbest_{ij}^k$: nilai terbaik secara keseluruhan

$rand( )$: adalah nomor acak antara (0,1)

c_1, c_2 : faktor yang mempengaruhi kecepatan perpindahan partikel, biasanya $c_1 = c_2 = 2$

Diagram alir dari PSO ditunjukkan pada Gambar 2.3. Beberapa istilah umum yang biasa digunakan dalam PSO dapatdidefinisikan sebagai berikut (Tuegeh, et al., 2009):

1. *Swarm* : populasi dari suatu algoritma.
2. Particle: anggota (individu) pada suatu swarm.
   Setiap partikel 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* : bobot *inertia* disimbolkan w, parameter ini digunakan untuk mengontrol dampak dari adanya *velocity* yang diberikan oleh suatu partikel.

Gambar 2.3. Diagram Alir PSO
(Xiao et al., 2018)

Sebagaimana dijelaskan sebelumnya, dalam PSO, sekumpulan partikel bergerak mencari solusi secara bersamaan, di mana setiap partikel mewakili solusi kandidat. Partikel tersebut mencatat solusi terbaiknya (*pbest*) dan solusi terbaik yang ditemukan oleh segerombolan (*gbest*) sejauh ini. Dua posisi terbaik diharapkan memimpin gerombolan untuk menjelajahi wilayah pencarian yang menjanjikan. Posisi partikel diwakili oleh vektor yang merupakan representasi

alami untuk pemilihan fitur. Secara khusus, setiap elemen vektor sesuai dengan fitur asli, dan nilainya menunjukkan apakah fitur terkait dipilih atau tidak. Ada banyak penelitian yang diajukan untuk meningkatkan efektivitas dan efisiensi algoritma pemilihan fitur berbasis PSO, diantaranya pengelompokan teks (Abualigah et al., 2018), masalah aliran data(Fong et al., 2015) dan pada analisis gambar (Silva et al., 2018).

2.1.4.3 Seleksi Fitur menggunakan *Binary*-PSO (BPSO)

PSO awalnya diusulkan dalam bentuk representasi kontiniu. Dalam menerapkan PSO untuk menyelesaikan pengoptimalan biner adalah dengan tetap menggunakan bentuk representasi kontiniu dan mengubah posisi kontinu ke posisi biner. Fungsi sigmoid banyak digunakan untuk tugas ini karena dapat mengubah nilai kontinu apa pun menjadi nilai kontinu dalam rentang [0,1], yang kemudian diubah menjadi nilai biner dengan membandingkannya dengan angka acak atau ambang batas(Nguyen et al., 2020). Nilai acak antara [0,1] digunakan untuk mengubah nilai kontinu yang diperoleh menjadi nilai biner. Pendekatan di atas telah diterapkan untuk mencapai pemilihan fitur(Qasim & Algamal, 2018) dan (Yadav et al., 2018). Namun, karena mekanisme pencarian kontinu masih diterapkan, pendekatan di atas juga mengalami konvergensi dini seperti pada PSO kontinu.

Tran et al.(2017) mengusulkan representasi yang dapat mencapai pemilihan fitur dan diskritisasi fitur. Dalam representasi yang diusulkan, setiap elemen dalam vektor posisi digunakan sebagai titik potong untuk membedakan fitur nilai nyata asli. Jika nilai elemen berada di luar rentang yang telah ditentukan, fitur terkait akan dibuang. Representasi yang diusulkan membantu PSO untuk memilih sejumlah kecil fitur dan mencapai kinerja klasifikasi yang lebih baik daripada menggunakan representasi standar. Namun, algoritma yang diusulkan diubah dari nilai kontinu menjadi nilai biner, yang mungkin kehilangan informasi diskriminatif. Selain itu, daftar titik potong perlu dibuattelah ditentukan sebelumnya untuk setiap fitur.Representasi PSO standar terdiri dari elemen nilai

nyata, yang dapat disebut representasi kontinu. Jika nilai elemen lebih besar dari ambang θ, fitur terkait dipilih. Jika tidak, fitur tersebut akan dibuang.

Akan tetapi, dalam ruang pencarian biner, tidak ada arah, dan partikel bergerak dengan membalik bitnya. Oleh karena itu, penggunaan konsep kecepatan dan momentum dari PSO kontinu tidak tepat. Untuk menghindari batasan ini, Nguyen et al.(2017) mengusulkan algoritma *binery*-PSO baru di mana konsep kecepatan dan momentum didefinisikan ulang. Dalam algoritma yang diusulkan, kecepatan didefinisikan sebagai probabilitas bit membalik pada posisi. Momentum diartikan sebagai kecenderungan untuk bertahan dengan posisi saat ini. Algoritma BPSO yang diusulkan dapat memilih subset fitur yang lebih baik dengan kinerja klasifikasi yang lebih tinggi daripada algoritma PSO biner standar karena algoritma ini mendeskripsikan pergerakan biner secara lebih akurat.

Dibandingkan dengan PSO kontiniu, ada lebih sedikit penelitian yang menerapkan BPSO dengan representasi biner untuk pemilihan fitur. Padahal, representasi biner lebih cocok untuk pemilihan fitur daripada representasi kontinu(Nguyen et al., 2020). Satu *subset* fitur dapat direpresentasikan oleh satu vektor biner. Sementara itu, satu *subset* fitur dapat direpresentasikan oleh banyak (mungkin tak hingga) jumlah vektor kontinu. Oleh karena itu, menggunakan representasi berkelanjutan secara signifikan memperbesar ruang pencarian(Nguyen et al., 2020).Oleh karena itu, penelitian ini bertujuan untuk mengembangkan algoritma PSO, yang dapat mengatasi ruang pencarian biner dan mempertahankan tiga properti penting algoritma PSO: *momentum*, komponen *kognitif* dan sosial.

BPSO yang akan diterapkan pada penelitian ini menggunakannilai peluang "membalik" yang diperkenalkan oleh (Xue et al., 2014)untuk menggantikan kecepatan memperbarui setiap partikel selama proses evolusi. p menunjukkan nilai peluang "membalik", yang merupakan vektor berdimensi-d (d menyatakan banyak fitur).$p_i = (p_{i1}, p_{i2}, \dots, p_{id})$ menunjukkan nilai peluang "membalik" untuk partikel i. p_{id}menunjukkan kemungkinan nilai x_{id}"dibalik", misalnya $update x_{id}^{t+1} = 1$ jika $x_{id}^{t} = 0$, atau $update x_{id}^{t+1} = 0$ jika $x_{id}^{t} = 1$, dimana t menyatakan iterasi ke-$t$pada proses PSO. Untuk memperbarui posisi partikel

digunakan rumus (3). p dihitung berdasarkan posisi terakhir dari partikel, *pbest* dan *gbest* dengan mempertimbangkan rumus (7) dan (8) berikut (Xue et al., 2014):

$$x_{id}^{t+1} = \begin{cases} 1 - x_{id}^{t}, & if\ random() < p_{id} \\ x_{id}^{t}, & untuk\ yang\ lainnya. \end{cases} \quad (7)$$

$$p_{id} = p_0 + p_{pd} + p_{gd} = 1 \quad (8)$$

Dimana,

$$p_{pd} = \begin{cases} p_1, & jika\ x_{id}^{t} \neq y_{id}^{t} \\ 0, untuk\ yang\ lainnya \end{cases}$$

dan

$$p_{gd} = \begin{cases} p_2, & jika\ x_{id}^{t} \neq \hat{y}_{id}^{t} \\ 0, untuk\ yang\ lainnya \end{cases}$$

Pada rumus (7), $(1 - x_{id}^{t})$digunakan untuk memperbarui x_{id}^{t+1} dari 1 menjadi 0 atau dari 0 menjadi 1.p_{pd}danp_{gd} menyatakan pengaruh *pbest*dan *gbest*.$p_0, p_1,$ dan p_2 merupakan bilangan real (0,1). 0 <$p_0$ditetapkan untuk memastikan bahwa selalu ada peluang untuk memperbarui nilai dari x_{id}^{t}. Nilai dari p_{pd}danp_{gd} dihitung untuk setiap dimensi pada setiap iterasi. Nilai $p_0, p_1,$ dan p_2 ditentukan di awal dengan ketentuan $p_0 + p_1 + p_2 = 1$. ketentuan $p_0 + p_1 + p_2 = 1$ digunakan untuk memastikan bahwa ketika x_{id}^{t} tidak sama dengan *pbest*y_{id}^{t}dang*best*$\hat{y}_{id}^{t}$(Xue et al., 2014). Pada penelitian ini, nilai parameter ditentukan yaitu: $p_0 = 0.3,\ p_1 = 0.3,\ p_2 = 0.4$ untuk menyatakan bahwa *gbest*lebih berpengaruh dari pada *pbest*.

2.1.5 Evaluasi Klasifikasi

Pengukuran terhadap kinerja suatu sistem klasifikasi merupakan hal yang penting dalam mengevaluasi kinerja dari suatu model klasifikasi. 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.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 sistem 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. Pada pengukuran kinerja menggunakan *confusion matrix*, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Nilai *True Negative* (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan *False Positive* (FP) merupakan data negatif namun terdeteksi sebagai data positif. Sementara itu, *True Positive* (TP) merupakan data positif yang terdeteksi benar. *False Negative* (FN) merupakan kebalikan dari *True Positive*, sehingga data posifit, namun terdeteksi sebagai data negatif.

Tabel 2.1 Confusion Matrix

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
Positif	TP (True Positive)	FN (False Negative)
Negatif	FP (False Positive)	TN (True Negative)

Berdasarkan nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP) dapat diperoleh nilai akurasi, presisi,*recall*dan F1-Score. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai akurasi dapat diperoleh dengan Persamaan (9). Nilai presisi

menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. Presisi dapat diperoleh dengan Persamaan (10). Sementara itu, *recall* menunjukkan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh sistem. Nilai *recall* diperoleh dengan Persamaan (11). Nilai F1-Score merupakan rata-rata harmonik precision dan recall, diperoleh dengan menggunakan Persamaan (12).

Akurasi = (TP + TN) / (TP + TN + FP + FN)) * 100% . ……..…… (9)

Precision = (TP / (TP + FP)) * 100%. …… (10)

*Recall = (TP / (TP + FN)) * 100%. ……..… (11)*

F1 Score = 2 * (*Precision * Recall) / (Precision + Recall)*..…… (12)

dimana:

- TP adalah *True Positive*, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem.
- TN adalah *True Negative*, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem.
- FN adalah *False Negative*, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem.
- FP adalah *False Positive*, yaitu jumlah data positif namun terklasifikasi salah oleh sistem

Sementara itu, pada klasifikasi dengan jumlah keluaran kelas yang lebih dari dua (*multi-class*), cara menghitung akurasi, presisi dan recall dapat dilakukan dengan menghitung rata-rata dari nilai akurasi, presisi dan recall pada setiap kelas. Persamaan 4, 5, dan 6 merupakan formula untuk menghitung nilai akurasi, presisi dan recall dari sistem klasifikasi multi-class.

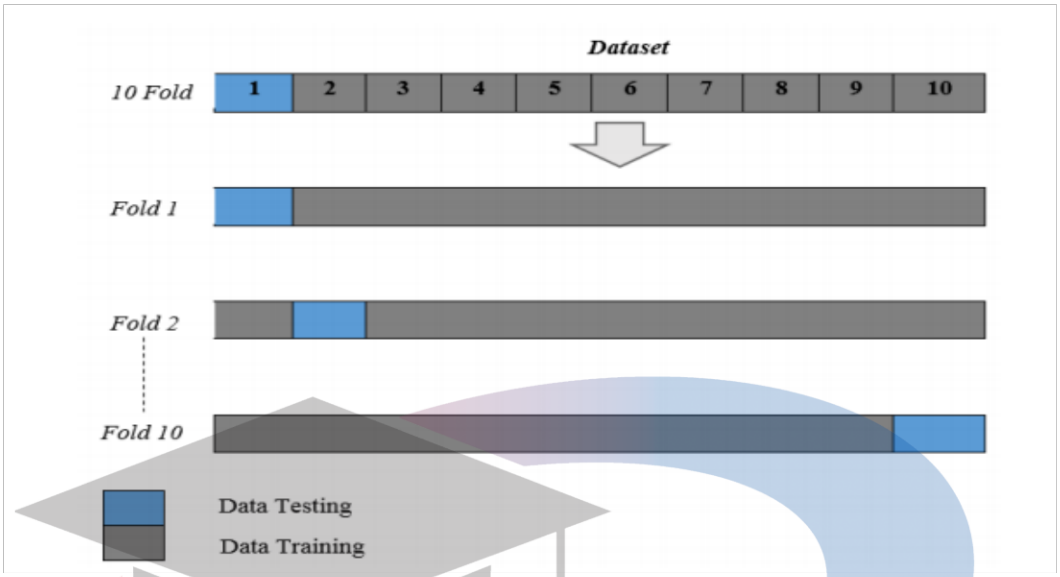
Pemilihan fitur memiliki dua tujuan utama, yaitu memaksimalkan akurasi klasifikasi (meminimalkan tingkat kesalahan/*ErrorRate*) dan meminimalkan jumlah fitur. Oleh karena itu, dalam *Binary*-PSO digunakan fungsi *fitness* yang menggabungkan kedua fungsi tujuan tersebut, yang ditunjukkan oleh rumus (13) sebagai berikut (Xue et al., 2014):

$$fitness = \alpha * ErrorRate + (1 - \alpha) * \frac{\#banyak\ fitur\ terpilih}{\#banyak\ semua\ fitur} \quad (13)$$

Di mana *ErrorRate* mewakili tingkat kesalahan klasifikasi dari fitur yang dipilih. *#banyak fitur terpilih*menunjukkan jumlah fitur yang dipilih dan *#banyak semua fitur* menunjukkan jumlah total fitur dalam dataset. Dan$(1 - \alpha)$mencerminkan prioritas relatif dari kinerja klasifikasi dan jumlah fitur yang dipilih. $\alpha \in (0{,}5; 1]$ karena akurasi dari kinerja klasifikasi dianggap lebih penting daripada pengurangan jumlah fitur(Xue et al., 2014).

2.1.6 *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 (D_1, D_2, D_3, ..., D_n), selanjutnya proses testing dan training dilakukan sebanyak *n* kali. Dalam iterasi ke-*i*, partisi D_i akan menjadi data testing dan sisanya akan menjadi data training. Penggunaan jumlah fold terbaik untuk uji validitas, dianjurkan menggunakan 10 *fold cross validation* dalam model (Prasetyo, 2014).Contoh pembagian *dataset* dalam proses 10-*fold cross validation* terlihat pada Gambar 2.4.



Gambar 2.4. Iterasi pada 10-fold cross validation (Prasetyo, 2014)

Cara kerja *K-fold cross validation* adalah sebagai berikut (Prasetyo, 2014):

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 (Prasetyo, 2014):

$$\text{Akurasi} = \frac{\sum \text{data uji benar klasifikasi}}{\sum \text{total data uji}} \times 100\% \qquad (14)$$

3. *Fold* ke-2 adalah ketika bagian ke-2 menjadi data uji (*testing data*) dan sisanya menjadi data latih (*trainingdata*). 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.

2.1.7 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.

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 benarbenar diterima kembali di masa tertentu di masa mendatang.

2) Kesepakatan

Kesepakatan dituangkan dalam suatu perjanjian di mana masingmasing 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.

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 *solvabilitas*-nya, 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.

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*

   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*

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.

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 angusran 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.

2.1.8 Penelitian Terdahulu

Penelitian terkait klasifikasi risiko kredit telah banyak dilakukan sebelumnya dan berbagai metode atau algoritma klasifikasi telah dipelajari dan diterapkan seperti ditunjukkan pada Tabel 2.2 berikut ini.

Tabel 2.2 Penelitian Terdahulu

Nama	Judul	Teknik Evaluasi	Teknik Seleksi Fitur	Masalah
Gafarova (2017)	Usage of Artificial Neural Network and Support Vector Machine model for classification of Credit Scores	Klasifikasi SVM, ANN	-	Penilaian Kredit
(Ivandari et al., 2017)	Data Attribute Selection with Information Gain to Improve Credit Approval Classification Performance using K-Nearest Neighbor Algorithm	Klasifikasi KNN	Information Gain	Penilaian Kredit

(Kaur & Cheema, 2018)	Selective Feature Processing With K-Nearest Neighbor Classification To Predict Credit Card Frauds	Klasifikasi KNN		Penilaian Kredit (Kartu Kredit)
(Abualigah et al., 2018)	A New Feature Selection Method To Improve The Document Clustering Using Particle Swarm Optimization Algorithm	K-Means Clustering	PSO	Dokumen tulisan (text)
(Ghosh et al., 2019)	A Wrapper-Filter Feature Selection Technique Based On Ant Colony Optimization	Klasifikasi KNN dan Multi-Layer Perceptron.	ACO	A Typical Pattern Recognition Problem
Zhang (2020)	The Impact Of Distance, Feature Weighting And Selection For KNN In Credit Default Prediction	Pembobotan Fitur	KNN	Penilaian Kredit (Kartu Kredit)
Maleki et al. (2020)	A k-NN Method For Lung Cancer Prognosis With The Use Of A Genetic Algorithm For Feature Selection	KNN	GA	Lung cancer prognosis

Gafarova (2017), yang membandingkan kinerja algoritma *Artificial Neural Network* (ANN) dan algoritma *Support Vector Machine* (SVM) dalam menganalisis penilaian kredit. Sedangkan (Ivandari et al., 2017) dan (Kaur & Cheema, 2018) telah mempelajari penerapan metode kNN dalam mengklasifikasi penilaian kredit.

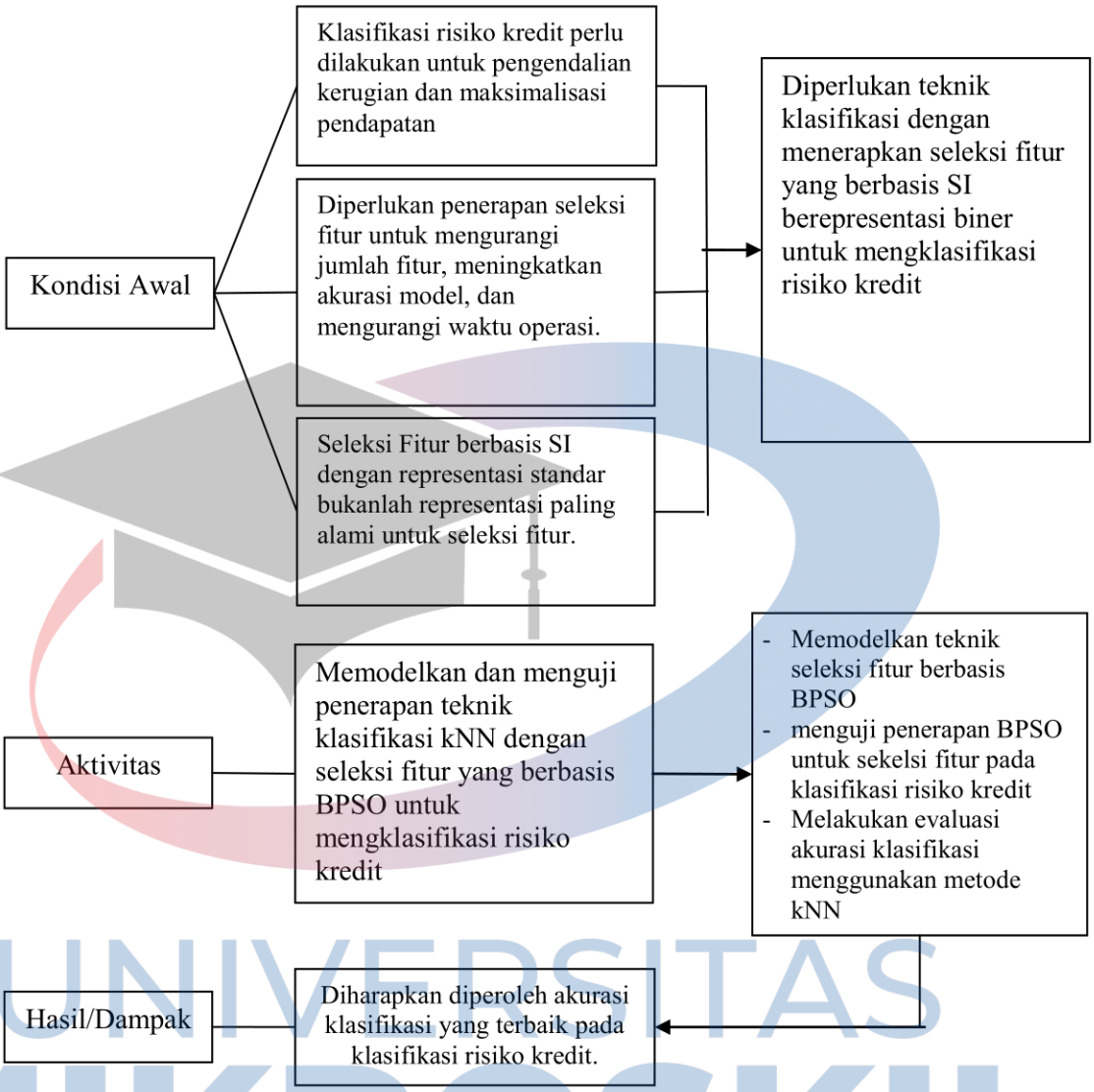
Miao & Niu (2016) dalam penelitiannya mempelajari bahwa pada dasarnya, prinsip dari seleksi fitur adalah untuk menghilangkan fitur yang berlebihan dengan menggunakan algoritma pemilihan fitur, sehingga dapat

mengurangi jumlah fitur, meningkatkan akurasi model, dan mengurangi waktu berjalan. Terkait metode klasifikasi KNN, Zhang (2020) telah mempelajari peningkatan kinerja metode ini menjadi lebih baik dengan menerapkan seleksi (pemilihan) fitur yang mengacu pada pemilihan subset fitur dari semua fitur yang ada. Maleki et al. (2020) juga telah menerapkan seleksi fitur pada klasifikasi kNN dalam mengklasifikasi kanker paru-paru prognosis dengan hasil akurasi 100%.

Untuk itu, diperlukan cara yang efisien dalam menyeleksi sebanyak *n* fitur karena seleksi fitur merupakan persoalan kombinatorial sebanyak 2^n kombinasi. Baru-baru ini, teknik *swarm intelligence* (SI) telah mendapatkan banyak perhatian dari komunitas seleksi fitur karena kesederhanaan dan potensi kemampuan pencarian globalnya (Nguyen et al., 2020). Algoritma SIyang telah dipelajari untuk menyeleksi fitur diantaranya yaitu *Particle Swarm Optimization* (PSO) (Abualigah et al., 2018) dan *Ant Colony Optimization* (ACO) (Ghosh et al., 2019).

2.2 Kerangka Pikir Pemecahan Masalah

Kerangka pikir pemecahan masalah menggambarkan bagaimana masalah penelitian dapat diselesaikan melalui solusi-solusi yang diusulkan serta dari solusi tersebut diharapkan memiliki dampak yang dapat menyelesaikan permasalahan penelitian. Berikut ini pada Gambar 2.5., akan digambarkan kerangka pikir pemecahan masalah dari penelitian yang akan dilakukan.



Gambar 2.5 Kerangka Pikir Pemecahan Masalah

Model klasifikasi risiko kredit diharapkan mampu menyediakan pemisahan antara peminjam yang berpotensi gagal dengan yang tidak gagal dalam hal pembayaran kredit dengan jangka waktu yang telah disepakati. Penggunaan kNN untuk klasifikasi risiko kredit menarik untuk dilakukan, karena berbagai keunggulan yang dimiliki metode kNN. Metode kNN dapat menjadi lebih baik dengan menerapkan seleksi fitur yang mengacu pada pemilihan subset fitur dari semua fitur yang ada. Algoritma PSO telah berhasil diterapkan pada seleksi fitur,

namun kebanyakan menggunakan representasi standar. PadaPSO dengan representasi standar (kontiniu), setiap elemen dalam vektor posisi digunakan sebagai titik potong untuk membedakan fitur nilai nyata asli. Jika nilai elemen berada di luar rentang yang telah ditentukan, fitur terkait akan dibuang. Representasi kontiniu membantu PSO untuk memilih sejumlah kecil fitur dan mencapai kinerja klasifikasi yang lebih baik.

Namun, algoritma PSO dengan representasi standar yang diubah dari nilai kontinu menjadi nilai biner, memungkinkan kehilangan informasi diskriminatif. Representasi biner adalah representasi yang paling natural untuk seleksi fitur dan belum banyak dipelajari. Tantangan dalam ruang pencarian biner yaitu, tidak adanya arah, dan partikel bergerak dengan membalik bitnya. Oleh karena itu, penggunaan konsep kecepatan dan momentum dari PSO kontinu tidak tepat.

Untuk menghindari batasan di atas, penelitian ini mengkontribusikan pengembangan algoritma seleksi fitur berbasis BPSO yang mana konsep kecepatan dan momentum pada algoritma PSO didefinisikan ulang. Dalam algoritma yang diusulkan, kecepatan didefinisikan sebagai probabilitas membalik bit pada posisi partikel. Penerapan BPSO dengan representasi biner diharapkan dapat mengoptimalkan akurasi model klasifikasikNN dengan seleksi fitur padarisiko kredit.