

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian ini dilakukan tidak terlepas dari hasil penelitian-penelitian terdahulu yang pernah dilakukan sebagai bahan perbandingan dan kajian. Diantaranya adalah Penelitian yang dilakukan oleh Deni Malik Irawan (2022) dengan judul penelitian “Penerapan *Data Mining* Untuk menilai Kinerja Guru Menggunakan Metode *K-Means Clustering* (Studi Kasus: SD Negeri Jatimulyo)”. Hasil dari penelitian ini digunakan sebagai salah satu dasar pengambilan keputusan untuk menentukan kinerja guru SD Negeri Jatimulyo berdasarkan *cluster* yang terbentuk oleh hasil perhitungan kuesioner tiap guru di SD Jatimulyo.

Penelitian Kedua dilakukan oleh Nurul Kasanah (2023) dengan judul penelitian “Implementasi *Data Mining* Dalam Mengelompokan Tingkat Kemiskinan Penduduk Di Provinsi Jawa Tengah Menggunakan Algoritma *K-Means Clustering*”. Hasil dari penelitian ini didapatkan bahwa dengan adanya 3 *cluster* kemiskinan dari tahun 2011 hingga tahun 2022, terdapat 166 data Kabupaten/Kota berada pada *cluster* 0, 92 data Kabupaten/Kota berada pada *cluster* 1 dan 162 data Kabupaten/Kota berada pada *cluster* 2.

Penelitian ketiga dilakukan oleh Nisriina Nuur Hasanah, dkk.(2022), dengan judul penelitian “Implementasi *Data Mining* Untuk Pengelompokan Buku Menggunakan Algoritma *K-Means Clustering* (Studi Kasus: Perpustakaan

Politeknik LPP Yogyakarta)”. Hasil akhir dari penelitian berupa data buku paling diminati, cukup diminati dan sedikit diminati.

Penelitian keempat dilakukan oleh Hesti Astuti(2019) dengan judul penelitian “Penerapan *Data Mining* Menggunakan Metode *K-Means Clustering* Untuk Pengelompokan Data Pelanggan (Studi Kasus : PT. Pinus Merah Abadi)”. Hasil dari penelitian ini adalah menghasilkan tiga *cluster* pelanggan dengan karakteristik pelanggan dari tiap-tiap *cluster*. Hasil dari *clustering* ini dapat membantu PT.Pinus Merah Abadi untuk mengambil kebijakan dan strategi dalam meningkatkan atau mempertahankan pelanggan yang sesuai dengan *cluster* pelanggan tersebut.

Penelitian kelima dilakukan oleh Idham Kholid Ramadhani (2023) dengan judul penelitian “Implementasi *Data Mining* Penentuan Kelompok Pelanggan Listrik Subsidi Atau *Non Subsidi*”. Hasil dari penelitian ini adalah Akurasi prediksi metode *KNN* pada penelitian ini dihasilkan 99% pada k bernilai 3, nilai tersebut sudah diuji dengan melakukan perbandingan akurasi pada k=3 dengan *accuracy score* 0.9995, k=5 dengan *accuracy score* 0.9990 dan k=7 dengan *accuracy score* 0.9990, sehingga disimpulkan bahwa nilai k yang diimplementasikan pada prediksi status pelanggan PLN menghasilkan nilai mendekati akurat dan tepat.

Perbandingan penelitian yang digunakan dalam penelitian ini dapat dilihat pada tabel perbandingan 2.1

Tabel 2 1 Perbandingan Penelitian

Peneliti	Judul Penelitian	Metode	Objek	Hasil
Deni Malik Irawan (2022)	Penerapan <i>Data Mining</i> Untuk menilai Kinerja Guru Menggunakan Metode <i>K-Means Clustering</i> (“Studi Kasus: SD Negeri Jatimulyo”)	<i>K-Means Clustering</i>	Kinerja Guru SD Negeri Jatimulyo	Hasil dari penelitian ini digunakan sebagai salah satu dasar pengambilan keputusan untuk menentukan kinerja guru SD Negeri Jatimulyo berdasarkan <i>cluster</i> yang terbentuk oleh hasil perhitungan kuesioner tiap guru di SD Jatimulyo.
Nurul Kasanah (2023)	Implementasi <i>Data Mining</i> Dalam Mengelompokkan Tingkat Kemiskinan Penduduk Di Provinsi Jawa Tengah Menggunakan Algoritma <i>K-Means Clustering</i>	<i>K-Means Clustering</i>	Tingkat Kemiskinan Di Provinsi Jawa Tengah	Hasil dari penelitian ini didapatkan bahwa dengan adanya 3 <i>cluster</i> kemiskinan dari tahun 2011 hingga tahun 2022, terdapat 166 data Kabupaten/Kota berada pada <i>cluster</i> 0, 92 data Kabupaten/Kota berada pada <i>cluster</i> 1 dan 162 data Kabupaten/Kota berada pada <i>cluster</i> 2.
Nisriina Nur Hasanah, dkk. (2022)	Implementasi <i>Data Mining</i> Untuk Pengelompokan Buku Menggunakan Algoritma <i>K-Means Clustering</i> (Studi Kasus: Perpustakaan Politeknik LPP Yogyakarta)	<i>K-Means Clustering</i>	Perpustakaan Politeknik LPP Yogyakarta	Hasil akhir dari penelitian berupa data buku paling diminati, cukup diminati dan sedikit diminati.
Hesti Astuti (2021)	Penerapan <i>Data Mining</i> Menggunakan Metode <i>K-Means Clustering</i> Untuk Pengelompokkan Data Pelanggan (Studi Kasus : PT. Pinus Merah Abadi)	<i>K-Means Clustering</i>	PT. Pinus Merah Abadi	Hasil dari <i>clustering</i> ini dapat membantu PT.Pinus Merah Abadi untuk mengambil kebijakan dan strategi dalam meningkatkan atau mempertahankan pelanggan yang sesuai dengan <i>cluster</i> pelanggan tersebut.

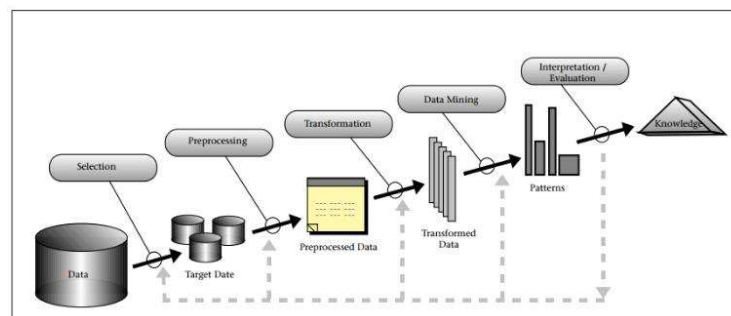
Peneliti	Judul Penelitian	Metode	Objek	Hasil
Idham Kholid Ramadhani (2023)	Implementasi <i>Data Mining</i> Penentuan Kelompok Pelanggan Listrik Subsidi Atau <i>Non Subsidi</i>	K-Medoids & K-Nearest Neighbor (K-NN)	PT. PLN Kebumen	Hasil dari penelitian ini adalah Akurasi prediksi metode <i>KNN</i> pada penelitian ini dihasilkan 99% pada <i>k</i> bernilai 3, nilai tersebut sudah diuji dengan melakukan perbandingan akurasi pada <i>k</i> =3 dengan <i>accuracy score</i> 0.9995, <i>k</i> =5 dengan <i>accuracy score</i> 0.9990 dan <i>k</i> =7 dengan <i>accuracy score</i> 0.9990, sehingga disimpulkan bahwa nilai <i>k</i> yang diimplementasikan pada prediksi status pelanggan PLN menghasilkan nilai mendekati akurat dan tepat.
Angelina Nope (2023)	Implementasi <i>Data Mining</i> Untuk Pengelompokan Buku menggunakan Algoritma <i>K-Means Clustering</i>	<i>K-Means Clustering</i>	Situs <i>GoodReads</i> Indonesia	Penelitian ini menghasilkan pengelompokan data buku dengan tiga cluster yaitu <i>cluster 0</i> , <i>cluster 1</i> dan <i>cluster 2</i> . Implementasi <i>Data Mining</i> menggunakan algoritma <i>K-Means clustering</i> untuk pengelompokan data buku menghasilkan hasil akhir dari pengelompokan data sebanyak 2123 data buku dengan 1567 data <i>cluster 0</i> , 487 data <i>cluster 1</i> dan 69 data <i>cluster 2</i> .

## 2.2 Dasar Teori

### 2.2.1 Data Mining

*Data mining* merupakan serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini tidak diketahui secara manual dari suatu basis data. *Data mining* mulai ada sejak 1990-an sebagai cara yang benar dan tepat untuk mengambil pola dan informasi yang digunakan untuk menemukan hubungan antara data untuk melakukan pengelompokan ke dalam satu atau lebih *cluster* sehingga objek-objek yang berada dalam satu *cluster* akan mempunyai kesamaan yang tinggi antara satu dengan lainnya. *Data mining* merupakan bagian dari proses penemuan pengetahuan dari basis data *Knowledge Discovery in Databases* (Alkhairi & Windarto, 2019).

Dalam penerapannya, data mining merupakan salah satu bagian dari sebuah proses yang dinamakan *Knowledge Discovery in Database* (KDD) yaitu proses ekstraksi *non trivial* dari *implicit* suatu informasi yang sebelumnya tidak diketahui tetapi terdapat potensi informasi yang dihasilkan dari data yang ada (Ependi & Putra, 2019). Grafik KDD ditunjukkan pada Gambar 2.1



Gambar 2. 1 Tahapan *Knowledge Discovery Database* (Ependi & Putra, 2019)

Proses *KDD* bersifat interaktif dan berulang, melibatkan banyak langkah dengan banyak keputusan yang dibuat oleh pengguna. Brachman dan Anand (1996) memberikan pandangan praktis tentang proses *KDD*, dengan menekankan sifat interaktif dari proses tersebut (*traduccionesbigdata.blogspot*).

Adapun proses *Knowledge Discovery Database* adalah sebagai berikut:

1. *Data Selection*: proses pengambilan data - data yang relevan untuk kemudian dimasukkan ke proses analisis.
2. *Preprocessing*: proses *cleaning* data yang mencakup beberapa proses seperti membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data.
3. *Data transformation*: proses transformasi dan konsolidasi data ke dalam bentuk yang sesuai untuk ditambang supaya bisa menghasilkan sebuah kesimpulan atau penggabungan.
4. *Data mining*: proses awal yaitu penerapan metode pengkajian untuk mengekstraksi pola data.
5. *Pattern evaluation*: proses mengidentifikasi pola unik yang mewakili basis pengetahuan berdasarkan ukuran tertentu.
6. *Knowledge Presentation*: proses teknik visualisasi dan presentasi yang digunakan untuk menampilkan pengetahuan atau hasil kepada pengguna.

### **2.2.2 Clustering**

*Clustering* adalah suatu proses pengelompokan kumpulan data ke dalam kelompok-kelompok sehingga objek-objek yang terdapat dalam suatu kelompok

memiliki kesamaan yang tinggi dan berbeda dengan objek yang berada dalam kelompok lain (Ghuman, 2016).

*Clustering* adalah bagian penting dari analisis data, yang bertujuan untuk mempartisi *dataset* yang diberikan ke dalam beberapa *cluster* yang mana *dataset* yang berada dalam suatu *cluster* memiliki kemiripan dan berbeda dari *dataset* yang berada di *cluster* lain (Raval dan Jani, 2016; Y. Rani dan Rohil, 2013).

### 2.2.3 *Elbow Method*

Metode *Elbow* merupakan sebuah metode optimasi *cluster* yang dihasilkan melalui perbandingan persentase antar *cluster* yang membentuk siku pada suatu titik. Hasil persentase yang berbeda dari setiap nilai *cluster* dapat ditunjukkan dengan menggunakan grafik sebagai sumber informasinya. Jika nilai *cluster* pertama dengan nilai *cluster* kedua memberikan sudut dalam grafik dan nilainya mengalami penurunan paling besar maka nilai *cluster* tersebut yang terbaik. Untuk mendapatkan perbandingannya adalah dengan menghitung nilai *Sum of Square Error (SSE)* dari masing-masing nilai *cluster*. Semakin jauh jarak yang membentuk titik siku, maka jumlah *cluster* tersebut menjadi yang paling optimal (Madhulatha (2012) dalam (Merliana et al., 2015)).

$$SSE = \sum_{i=1}^k \sum_{X_i \in C_i} D(X_i, C_i)^2 \dots \dots \dots (1)$$

Dimana:

$k$  = Jumlah *Cluster*

$X_i \in C_i$  = Nilai keanggotaan titik data  $X_i$  ke pusat kelompok  $C_i$

$C_i$  = Pusat *Cluster* ke- $i$

$D(X_i, C_i)$  = Jarak dari titik  $x_i$  ke kelompok  $C_i$  yang diikuti

#### 2.2.4 *K-Means Clustering*

*K-Means Clustering* merupakan salah satu metode *cluster* analisis *non* hirarki yang berusaha untuk mempartisi objek yang ada kedalam satu atau lebih *cluster* atau kelompok objek berdasarkan karakteristiknya, sehingga objek yang mempunyai karakteristik yang sama dikelompokkan dalam satu *cluster* yang sama dan objek yang mempunyai karakteristik yang berbeda dikelompokkan kedalam *cluster* yang lain (Syafnidawaty,2020).

Prinsip utama dari algoritma *K-Means* adalah menyusun  $K$  buah partisi dari sekumpulan data. Tujuan *K-Means* untuk meminimalkan fungsi objektif pada *dataset* kelompok dalam proses pengelompokan. Pada umumnya teknik ini berusaha meminimalkan variasi antar kelompok lainnya (Randi, Rony, Budiman, dkk (2023).

Terdapat langkah-langkah dalam penggunaan algoritma *K-Means Clustering*, yaitu sebagai berikut :

1. Pilih nilai  $K$  untuk menentukan jumlah *cluster* yang akan dibentuk.
2. Menetapkan titik  $K$  secara acak yang akan bertindak sebagai pusat *cluster* (*centroid*). Menentukan nilai *centroid* dengan cara mengambil dari nilai rata-rata (*mean*) semua nilai dari data pada setiap fiturnya. Jika  $M$  menyatakan



jumlah data pada suatu kelompok. maka  $i$  menyatakan fitur ke- $i$  dalam sebuah kelompok.

3. Perhitungan jarak dari *centroid cluster*. *Euclidean distance* adalah perhitungan untuk mengukur jarak dua titik dalam *euclidean space* yang mempelajari hubungan antara sudut dan jarak (Derisma, Firdaus, & Yusya, 2016). Dalam matematika *euclidean distance* digunakan untuk mengukur dua titik dalam satu dimensi yang memberikan hasil seperti perhitungan *pythagoras* (Mustofa & Suasana, 2018).

Berikut adalah persamaan *euclidean distance*:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_1)^2} \quad \dots\dots\dots(2)$$

Keterangan :

$d$  = Jarak

$x_1$  = Koordinat *latitude* 1

$x_2$  = Koordinat *latitude* 2

$y_1$  = Koordinat *Longitude* 1

$y_2$  = Koordinat *Longitude* 2

4. Tetapkan setiap titik data, berdasarkan jaraknya dari titik yang dipilih secara acak (*centroid*), ke *centroid* terdekat, yang akan membentuk *cluster* yang telah ditentukan sebelumnya.

$$n_k = \frac{1}{n_k} \sum d_i \quad \dots\dots(3)$$

Dimana :

$n_k$  = jumlah data dalam cluster

$d_i$  = jumlah dari nilai jarak yang masuk dalam masing-masing cluster

5. Tempatkan *centroid* baru dari setiap *cluster*.
6. Ulangi Langkah 3 sampai langkah 5 hingga nilai titik *centroid* tidak berubah (*konvergen*).

Jika telah *konvergen*, maka model baru siap digunakan.

Berikut contoh penerapan Algoritma *K-Means* menggunakan *study* kasus pada data yang akan digunakan dalam penelitian ini. Pada contoh penerapan ini hanya akan digunakan enam data yang diacak dari *dataset*. Berikut langkah penerapan metode *clustering K-Means* pada *dataset*.

Tabel 2 2 Contoh Perhitungan *K-Means*

No	Original_Rating	Adjusted_Rating	Rating_Count
1	3.00	3.0	34.0
2	2.21	3.0	14.0
3	3.70	4.0	33.0
4	4.20	4.0	20.0
5	5.0	5.0	3.0
6	5.00	5.0	1.0

Data untuk perhitungan manual menggunakan *K-Means clustering* dapat dilihat pada tabel 2.2

1. Menetapkan jumlah *cluster*. Jumlah *cluster* ditetapkan berdasarkan hasil penentuan nilai  $K$  dengan menggunakan metode *elbow*. Nilai  $K$  berdasarkan implementasi metode *elbow* adalah 3 *cluster*.

2. Inisialisasi jumlah *cluster* (*centroid*).

Centroid 1 = 3.00, 3.0, 34.0

Centroid 2 = 2.21, 3.0, 14.0

Centroid 3 = 5.00, 5.0, 1.0

3. Hitung jarak setiap data terhadap pusat *cluster*

Iterasi 1

*Centroid 1*

$$K1 = \sqrt{(3.00 - 3.00)^2 + (3.0 - 3.0)^2 + (34.0 - 34.0)^2} = 0$$

$$K2 = \sqrt{(2.21 - 3.00)^2 + (2.0 - 3.0)^2 + (14.0 - 34.0)^2} = 20,4$$

$$K3 = \sqrt{(3.70 - 3.00)^2 + (5.0 - 3.0)^2 + (33 - 34.0)^2} = 1.57$$

$$K4 = \sqrt{(4.20 - 3.00)^2 + (4.0 - 3.0)^2 + (20.0 - 34.0)^2} = 14.03$$

$$K5 = \sqrt{(5.00 - 3.00)^2 + (5.0 - 3.0)^2 + (1.0 - 34.0)^2} = 31,12$$

$$K6 = \sqrt{(5.00 - 3.00)^2 + (5.0 - 3.0)^2 + (1.0 - 34.0)^2} = 33.12$$

*Centroid 2*

$$K1 = \sqrt{(3.00 - 2.21)^2 + (3.0 - 2.0)^2 + (34.0 - 14.0)^2} = 20.4$$

$$K2 = \sqrt{(2.21 - 2.21)^2 + (2.0 - 2.0)^2 + (14.0 - 14.0)^2} = 0$$

$$K3 = \sqrt{(3.70 - 2.21)^2 + (5.0 - 2.0)^2 + (33 - 14.0)^2} = 19,16$$

$$K4 = \sqrt{(4.20 - 2.21)^2 + (4.0 - 2.0)^2 + (20.0 - 14.0)^2} = 6.63$$

$$K5 = \sqrt{(5.00 - 2.21)^2 + (5.0 - 2.0)^2 + (1.0 - 14.0)^2} = 11.73$$

$$K6 = \sqrt{(5.00 - 2.21)^2 + (5.0 - 2.0)^2 + (1.0 - 14.0)^2} = 13.63$$

*Centroid 3*

$$K1 = \sqrt{(3.00 - 5.00)^2 + (3.0 - 5.0)^2 + (34.0 - 1.0)^2} = 33.12$$

$$K2 = \sqrt{(2.21 - 5.00)^2 + (2.0 - 5.0)^2 + (14.0 - 1.0)^2} = 13.63$$

$$K3 = \sqrt{(3.70 - 5.00)^2 + (5.0 - 5.0)^2 + (33 - 1.0)^2} = 32.4$$

$$K4 = \sqrt{(4.20 - 5.00)^2 + (4.0 - 5.0)^2 + (20.0 - 1.0)^2} = 19.04$$

$$K5 = \sqrt{(5.00 - 5.00)^2 + (5.0 - 5.0)^2 + (3.0 - 1.0)^2} = 2$$

$$K6 = \sqrt{(5.00 - 5.00)^2 + (5.0 - 5.0)^2 + (1.0 - 1.0)^2} = 0$$

Pada Tabel 2.3 dapat dilihat hasil perhitungan *K-Means Clustering* dari perhitungan Iterasi 1

Tabel 2.3 Hasil Perhitungan *K-Means* Iterasi 1

No	A1	A2	A3	C1	C2	C3	Cluster
1	3.00	3.0	34.0	0	20.04	33,12	C1
2	2.21	3.0	14.0	20,04	0	13.63	C2
3	3.70	4.0	33.0	1.57	19.16	32.04	C1
4	4.20	4.0	20.0	14.03	6.63	19.04	C2
5	5.0	5.0	3.0	31,12	11.73	2	C3
6	5.00	5.0	1.0	33.12	13.63	0	C3

4. Menentukan kembali titik pusat *cluster* yang baru berdasarkan rata-rata *Cluster* baru tersebut didapat dari rumus = nilai hasil/banyak hasil

$$\text{Centroid 1} = 3.35, 3.5, 24$$

*Centroid 2 = 3.20, 3, 2*

*Centroid 3 = 5, 5, 2*

Iterasi 2

*Centroid 1*

$$K1 = \sqrt{(3.00 - 3.35)^2 + (3.0 - 3.5)^2 + (34.0 - 24)^2} = 10.01$$

$$K2 = \sqrt{(2.21 - 3.35)^2 + (2.0 - 3.5)^2 + (14.0 - 24)^2} = 10.17$$

$$K3 = \sqrt{(3.70 - 3.35)^2 + (5.0 - 3.5)^2 + (33 - 24)^2} = 9.02$$

$$K4 = \sqrt{(4.20 - 3.35)^2 + (4.0 - 3.5)^2 + (20.0 - 24)^2} = 4.11$$

$$K5 = \sqrt{(5.00 - 3.35)^2 + (5.0 - 3.5)^2 + (3.0 - 24)^2} = 21.11$$

$$K6 = \sqrt{(5.00 - 3.35)^2 + (5.0 - 3.5)^2 + (1.0 - 24)^2} = 23.10$$

*Centroid 2*

$$K1 = \sqrt{(3.00 - 3.20)^2 + (3.0 - 3)^2 + (34.0 - 17)^2} = 17$$

$$K2 = \sqrt{(2.21 - 3.20)^2 + (2.0 - 3)^2 + (14.0 - 17)^2} = 3.31$$

$$K3 = \sqrt{(3.70 - 3.20)^2 + (5.0 - 3)^2 + (33 - 17)^2} = 16.03$$

$$K4 = \sqrt{(4.20 - 3.20)^2 + (4.0 - 3)^2 + (20.0 - 17)^2} = 3.31$$

$$K5 = \sqrt{(5.00 - 3.20)^2 + (5.0 - 3)^2 + (3.0 - 17)^2} = 14.25$$

$$K6 = \sqrt{(5.00 - 3.20)^2 + (5.0 - 3)^2 + (1.0 - 17)^2} = 16.22$$

*Centroid 3*

$$K1 = \sqrt{(3.00 - 5)^2 + (3.0 - 5)^2 + (34.0 - 2)^2} = 32.12$$

$$K2 = \sqrt{(2.21 - 5)^2 + (2.0 - 5)^2 + (14.0 - 2)^2} = 12.67$$

$$K3 = \sqrt{(3.70 - 5)^2 + (5.0 - 5)^2 + (33 - 2)^2} = 31.04$$

$$K4 = \sqrt{(4.20 - 5)^2 + (4.0 - 5)^2 + (20.0 - 2)^2} = 18.04$$

$$K5 = \sqrt{(5.00 - 5)^2 + (5.0 - 5)^2 + (3.0 - 2)^2} = 1$$

$$K6 = \sqrt{(5.00 - 5)^2 + (5.0 - 5)^2 + (1.0 - 2)^2} = 1$$

Pada Tabel 2.4 dapat dilihat hasil perhitungan *K-Means Clustering* dari perhitungan Iterasi 2

Tabel 2 4 Hasil Perhitungan *K-Means* Iterasi 2

No	A1	A2	A3	C1	C2	C3	Cluster
1	3.00	3.0	34.0	10.1	17	32.12	C1
2	2.21	3.0	14.0	10.17	3.31	12.67	C2
3	3.70	4.0	33.0	9.02	16.03	31.04	C1
4	4.20	4.0	20.0	4.11	3.31	18.04	C2
5	5.0	5.0	3.0	21.11	14.25	1	C3
6	5.00	5.0	1.0	22.11	16.22	1	C3

Hasil dari Iterasi pertama dan kedua tidak berubah, maka hasil sudah sesuai dengan pengelompokan *cluster*.

### 2.2.5 Python

*Python* adalah bahasa pemrograman *high-level*, interpretatif, multiguna, berorientasi objek dengan semantik dinamis. Sintaks *Python* yang sederhana dan mudah dipelajari menekankan keterbacaan dan karenanya mengurangi biaya pemeliharaan program. *Python* mendukung modul dan paket, yang mendorong modularitas program dan *code reuse*. Interpreter *Python* dan pustaka standar yang

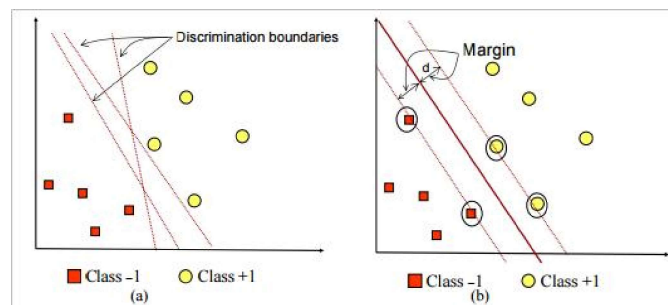
luas tersedia dalam bentuk *source* atau *biner* tanpa biaya untuk semua *platform* dan dapat didistribusikan secara bebas. (*Python Software Foundation*, 2022)

*Python* membuat penulisan program menjadi padat (*compact*) dan mudah dibaca. Program yang ditulis di *Python* pada dasarnya memerlukan kode yang lebih sedikit daripada program yang ditulis dengan bahasa *C*, *C++*, atau *Java* karena:

1. Tipe data *high-level* yang dapat melakukan operasi kompleks dalam satu *statement*.
2. Pengelompokan *statement* dilakukan dengan *indentation* (tulisan sedikit menjorok ke kanan), bukan dengan *brackets*.
3. Tidak memerlukan deklarasi variabel atau argument

### **2.2.6 Support Vector Machine (SVM)**

*Support Vector Machine (SVM)* adalah salah satu metode *machine learning* yang bekerja dengan prinsip *Structural Risk Minimization (SRM)* yang termasuk dalam kategori *supervised learning*. Dalam prosesnya, metode *SVM* memiliki tujuan yaitu untuk menemukan *hyperplane* paling optimal yang berfungsi untuk memisahkan dua buah kelas. Tingkat akurasi pada model yang dihasilkan oleh proses klasifikasi dengan *SVM* sangat bergantung terhadap fungsi kernel dan parameter yang digunakan. (Parapat, et al., 2018).



Gambar 2. 2 *Support Vector Machine* (Parapat, et al., 2018)

Pada Ilustrasi gambar 2.2, ada dua kelas yang dipisahkan oleh garis *hyperplane* yaitu kelas positif yang berniali +1 (lingkaran kuning) dan kelas negatif yang bernilai -1 (kotak merah). Garis solid yang terdapat pada tengah-tengah kedua kelas adalah *hyperplane* terbaik, dan objek merah dan kuning yang berada dalam lingkaran hitam disebut *support vector*.

Pada algoritma *Support Vector Machine*, data ke- $i$  pada *dataset* diwakilkan dengan variabel  $x$ , sementara kelas pada *dataset* diwakilkan dengan variabel  $y_i$ . Data  $x_i$  yang termasuk dalam kelas +1 dirumuskan dengan persamaan (1), sedangkan data  $x_i$  yang termasuk dalam kelas -1 dirumuskan dengan persamaan (2) (Parapat et al., 2018).

$$x_i \cdot w + b \geq y_i = 1 \quad \dots\dots(4)$$

$$x_i \cdot w + b \leq y_i = -1 \quad \dots\dots(5)$$

*Keterangan :*

$x_i$  = data ke - $i$



$w$  = nilai bobot *support vector* yang tegak lurus dengan *hyperplane*

$b$  = nilai bias

$y_i$  = kelas data ke- $I$

Berikut tahap – tahap perhitungan klasifikasi menggunakan *SVM*:

1. Meminimalkan nilai margin (Zalyhaty et al., 2020). Tahap ini dapat dilakukan dengan menggunakan persamaan berikut:

$$\frac{1}{2} \|w\|^2 = \frac{1}{2} \left( w \frac{1}{2} + w \frac{2}{2} \right) \dots\dots(6)$$

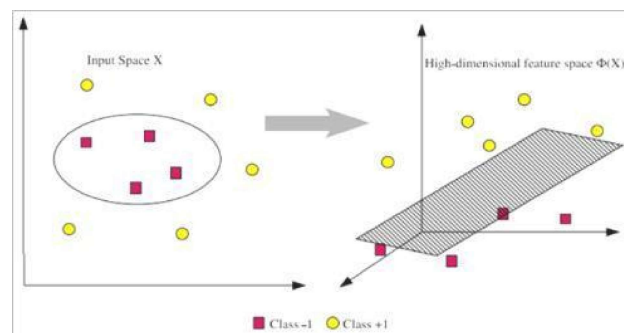
dengan syarat:  $y_i (w_i \cdot x_i + b) \geq 1, i = 1,2,3, \dots, n$  .....(7)

2. Setelah meminimalkan nilai margin, maka dapat ditemukan nilai  $w$  (bobot) dan nilai  $b$  (bias), lalu bisa dicari persamaan *hyperplane*.
3. Menghitung *margin hyperplane* dalam proses menemukan titik maksimal. Persamaan (5) adalah rumus untuk memperoleh garis *hyperplane* pada *SVM* (Husada & Paramita, 2021).

$$w_i \cdot x_i + b = 0 \dots\dots(8)$$

Prinsip kerja algoritma *Support Vector Machine* pada dasarnya adalah suatu algoritma yang digunakan untuk klasifikasi data *linear*, sehingga dalam proses klasifikasi seringkali ditemukan kondisi dimana *SVM* tidak bekerja dengan baik dalam melakukan klasifikasi pada data *non-linear*. Masalah tersebut bisa diatasi dengan menggunakan *kernel trick*. *Kernel trick* digunakan untuk memetakan data

*non-linear* berdimensi ke dalam ruang dimensi yang lebih tinggi sehingga membuat data terpisah secara *linear* lalu dapat terbentuk *hyperplane* yang optimal. Proses klasifikasi dengan SVM dapat dilakukan dengan memilih salah satu di antara 4 kernel yang tersedia yaitu *linear*, *polynomial*, *RBF*, dan *sigmoid* (Husada & Paramita, 2021). Ilustrasi *kernel trick* dapat dilihat di Gambar 2.3



Gambar 2. 3 Pemetaan *Input Space* Berdimensi Dua dengan Pemetaan ke Dimensi Tinggi (Rahutomo et al.,2018)

### 2.2.7 Figma

Figma adalah sebuah *tools* desain yang digunakan untuk membuat desain *user interface*, website, dan aplikasi *mobile*. Aplikasi ini juga dilengkapi fitur yang mendukung ilustrasi vektor, gambar *bitmap*, *typography*, dan juga animasi. Figma digunakan oleh developer untuk mendemokan secara langsung hasil akhir dari kode yang sudah dibuat sebelumnya untuk mengetahui bagaimana kode-kode tersebut akan bekerja ketika dieksekusi (academy.alterra.id).