# BAB 2
# LANDASAN TEORI

## 2.1 Diabetes

Diabetes merupakan penyakit yang ditandai dengan terjadinya hiperglikemia dan gangguan metabolisme karbohidrat, lemak, dan protein yang dihubungkan dengan kekurangan secara absolut atau relatif dari kerja dan atau sekresi insulin. Gejala yang dikeluhkan pada penderita Diabetes Melitus yaitu polidipsia, poliuria, polifagia, penurunan berat badan, kesemutan [9]. Secara garis besar diabetes dibagi menjadi beberapa jenis yaitu [10]:

1. Diabetes Mellitus tipe 1

   Dikenal sebagai diabetes remaja, tipe ini terjadi ketika tubuh gagal memproduksi insulin . Orang dengan diabetes tipe 1 tergantung pada insulin, yang berarti mereka harus meminum insulin buatan setiap hari untuk tetap hidup.

2. Diabetes Mellitus tipe 2

   Diabetes tipe 2 mempengaruhi cara tubuh menggunakan insulin. Sementara tubuh masih membuat insulin, tidak seperti pada tipe I, sel-sel dalam tubuh tidak meresponnya seefektif dulu. Ini adalah jenis diabetes yang paling umum, menurut Institut Nasional Diabetes dan Penyakit Pencernaan dan Ginjal, dan memiliki hubungan yang kuat dengan obesitas.

3. Diabetes Gestasional

   Tipe ini terjadi pada wanita selama kehamilan ketika tubuh dapat menjadi kurang sensitif terhadap insulin. Diabetes gestasional tidak terjadi pada semua wanita dan biasanya sembuh setelah melahirkan.

Diabetes Melitus dapat dicegah dengan menerapkan hidup sehat sedini mungkin, yaitu dengan mempertahankan pola makan sehari-hari yang sehat dan seimbang dengan meningkatkan konsumsi sayuran, buah dan serat, membatasi makanan yang tinggi karbohidrat, protein dan lemak, mempertahankan berat badan yang normal sesuai dengan umur dan tinggi badan serta olahraga teratur sesuai

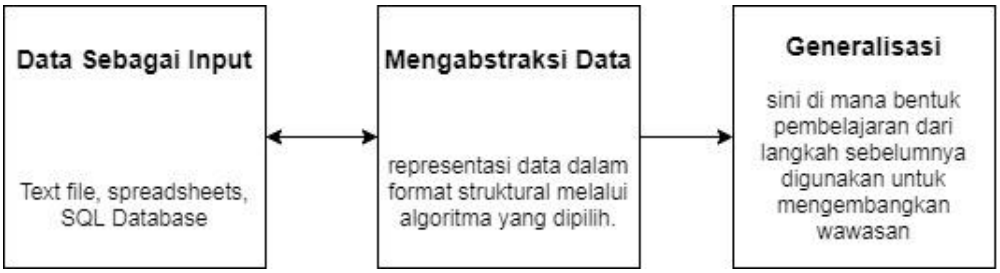
umur dan kemampuan. Tujuan pengobatan penderita Diabetes melitus ialah untuk mengurangi gejala, menurunkan berat badan bagi yang kegemukan dan mencegah terjadinya komplikasi [11].

## 2.2 Machine Learning

Machine learning adalah aplikasi dari disiplin ilmu kecerdasan buatan (*Artificial Intelligence*) yang menggunakan teknik statistika untuk menghasilkan suatu model otomatis dari sekumpulan data, dengan tujuan memberikan komputer kemampuan untuk belajar. Yang dimaksud belajar disini adalah mesin mampu belajar apabila dapat melakukan *update* parameter. Machine learning memungkinkan komputer mempelajari sejumlah data sehingga dapat menghasilkan suatu model untuk melakukan proses input-output tanpa menggunakan kode program yang dibuat secara eksplisit. Proses belajar tersebut menggunakan algoritma khusus yang disebut machine learning algorithms [12].

### 2.2.1 Cara Kerja Machine Learning

Machine Learning melibatkan proses struktural dimana setiap tahap membangun versi mesin yang lebih baik. Untuk penyederhanaan, proses *Machine Learning* dapat dibagi menjadi tiga bagian seperti pada Gambar 2.1 berikut .




**Gambar 2.1** *Cara kerja Machine Learning*

Selain itu, machine learning memiliki langkah-langkah yang dilakukan dalam proses pembelajaran mesin / *Machine Learning* yaitu sebagai berikut [13] :

a. Mengumpulkan data

Data mentah bisa berupa Excel, Ms Access, file teks dan lain-lain. Langkah ini membentuk dasar pembelajaran masa depan. Semakin banyak variasi,

kepadatan dan volume data yang relevan, semakin baik prospek pembelajaran untuk mesin.

b. Mempersiapkan data

Setiap proses analitis berkembang dengan kualitas data yang digunakan. Kita perlu meluangkan waktu untuk menentukan kualitas data dan kemudian mengambil langkah-langkah untuk memperbaiki masalah seperti kehilangan data dan lainnya.

c. Melatih sebuah model

Langkah ini melibatkan pemilihan alrgoritma dan representasi data yang tepat dalam bentuk model. Data yang disiapkan dibagi menjadi dua bagian : train dan test. Bagian pertama (training data) digunakan untuk pengembangan model. Bagian kedua (data test), digunakan sebagai referensi.

d. Mengevaluasi model

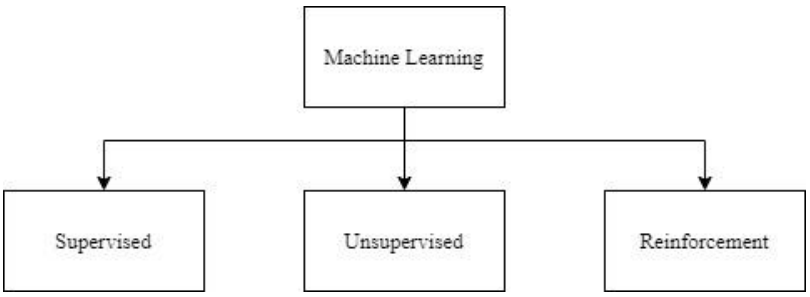
Untuk menguji keakuratan, bagian kedua dari data (data test) digunakan. Langkah ini menentukan ketepatan dalam pemilihan algoritma berdasarkan hasil pengujian. Pengujian yang lebih baik untuk memeriksa ketepatan model adalah dengan melihat kinerjanya pada data yang tidak digunakan sama sekali selama pembuatan model.

e. Meningkatkan kinerja

Langkah ini mungkin melibatkan pemilihan model yang berbeda atau memperkenalkan lebih banyak variabel unntuk meningkatkan efisiensi. Itulah sebabnya dibutuhkan banyak waktu untuk pengumpulan data dan persiapan data.

### 2.2.2 Jenis Algoritma Machine Learning

Pada dasarnya algoritma machine learning memiliki 3 tipe, yaitu Supervised, Unsupervised, dan Reinforcement [14]. Untuk lebih jelasnya akan terlihat pada Gambar 2.2 berikut :




**Gambar 2.2** *Jenis Algoritma Machine Learning*

a. Model Supervised Learning

Model ini digunakan untuk memprediksi hasil masa depan berdasarkan data historis. Model prediktif biasanya diberi instruktsi yang jelas sejak awal seperti apa yang perlu dipelajari dan bagaimana itu perlu dipelajari. Algoritma pembelajaran ini disebut Supervised Learning. Contoh algoritma yang digunakan adalah: K-Nearest Neighbour, Naïve Bayes, Decision Tree, Regression, dan lain-lain.

b. Model Unsupervised Learning

Model ini digunakan untuk melatih dimana tidak ada target yang ditetapkan dan tidak ada faktor yang penting dari yang lainnya. Sebagai contoh penggunaan model unspervised learning ini, bila seorang penjual pengecer ingin mengetahui kombinasi produk apa yang cenderung lebih sering dibeli konsumen. Contoh algoritma yang digunakan di model ini: K-Means Clustering Algorithm.

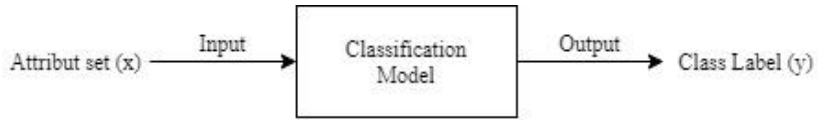
c. Reinforcement Learning (RL)

Model ini adalah contoh pembelajaran mesin dimana mesin dilatih untuk mengambil keputusan spesifik berdasarkan kebutuhan bisnis dengan tujuan utama untuk memaksimalkan efisiensi. Ide dari Reinforcement learning ini adalah perangkat lunak melatih dirinya secara terus menerus berdasarkan lingkungan yang dipengaruhinya, dan menerapkan pengetahuan yang diperkaya unntuk memecahkan masalah bisnis. Proses belajar yang terus-menerus ini

memastikan lebih sedikit keterlibatan manusia sehingga akan banyak menghemat waktu.

## 2.3 Metode Klasifikasi

Klasifikasi merupakan proses untuk menemukan model atau fungsi yang menjelasakan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Dalam mencapai tujuan tersebut, proses klasifikasi membentuk satu model yang mampu membedakan data kedalam kelas – kelas yang berbeda berdasarkan aturan atau fungsi tertentu. Model itu tersendiri bisa berupa pohon keputusan, atau formula matematis [15]. Berikut ini pada Gambar 2.2 merupakan gambaran proses dari metode klasifikasi.

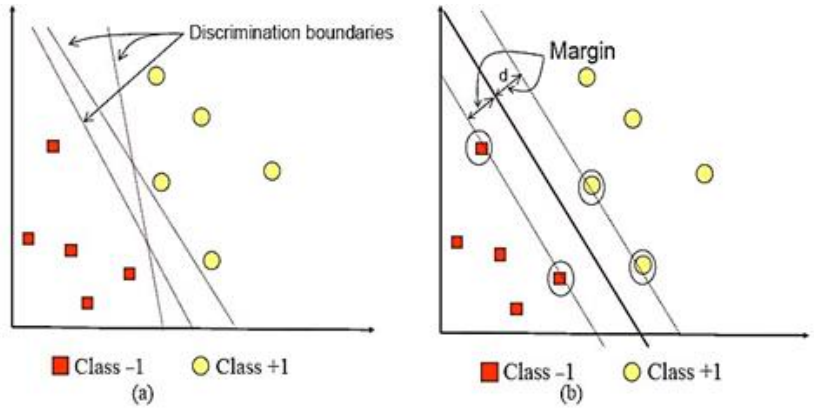



**Gambar 2.3** *Diagram Model Klasifikasi*

### 2.3.1 Support Vector Machine

Support Vector Machine (SVM) dikembangkan oleh Boser, Guyon, dan Vapnik, pertama kali diperkenalkan pada tahun 1992 di Annual Workshop on Computational Learning Theory. Konsep dasar metode SVM sebenarnya merupakan gabungan atau kombinasi dari teori-teori komputasi yang telah ada pada tahun sebelumnya, seperti margin hyperplane. Support Vector Machines (SVM) merupakan metode klasifikasi supervised karena ketika proses pelatihan, diperlukan target pembelajaran tertentu. SVM juga merupakan algoritma yang bekerja menggunakan pemetaan nonlinear untuk mengubah data pelatihan asli ke dimensi yang lebih tinggi. Dalam hal ini dimensi baru, akan mencari hyperplane untuk memisahkan secara linear dan dengan pemetaan nonlinear yang tepat ke dimensi yang cukup tinggi, data dari dua kelas selalu dapat dipisahkan dengan hyperplane tersebut [16]. Teknik SVM digunakan untuk mendapatkan fungsi pemisah (hyperplane) yang optimal untuk memisahkan observasi yang memiliki

nilai variabel target yang berbeda [17]. Berikut ini pada Gambar 2.3 merupakan gambaran dari proses dalam menemukan hyperplane dengan metode SVM.




**Gambar 2.4** *Proses menemukan hyperplane dengan metode SVM*

Hyperplane pemisah terbaik antara kedua class dapat ditemukan dengan mengukur margin hyperplane tersebut dan mencari titik maksimalnya. Margin adalah jarak antara hyperplane tersebut dengan pattern terdekat dari masing-masing class. Pattern yang paling dekat ini disebut sebagai support vector. Garis solid pada gambar 2.3 menunjukkan hyperplane yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua class, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah hitam adalah support vector. Usaha untuk mencari lokasi hyperplane ini merupakan inti dari proses pembelajaran pada SVM [18].

Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari hyperplane terbaik yang berfungsi sebagai pemisah dua buah class pada input space. Gambar 2.4 memperlihatkan beberapa pattern yang merupakan anggota dari dua buah class +1 dan -1. Pattern yang tergabung pada class -1 disimbolkan dengan warna merah (kotak), sedangkan pattern +1 disimbolkan dengan warna kuning (lingkaran).

Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (hyperplane) yang memisahkan antara kedua kelompok tersebut. Dalam linear Support Vector Machine permisah merupakan fungsi linear. Data latih dinyatakan oleh (xi, yi) dan xi = { x1, x2, ..... xiq} merupakan atribut (fitur) set untuk data

latih kelas ke-i. Untuk yi € {-1,1} menyatakan label kelas. Pendefinisian persamaan suatu hyperplane pemisah yang dituliskan dengan formulasi sebagai berikut:

$$w * x_i + b = 0 \quad (2.1)$$

Data $xi$ yang terbagi ke dalam dua kelas, yang termasuk kelas -1 (sampel negatif) didefinisikan sebagai vektor yang memenuhi pertidaksamaan (2.2) berikut.

$$w * xi + b < 0 \, untuk \, yi = -1 \quad (2.2)$$

Sedangkan yang termasuk kelas +1 (sampel positif) memenuhi pertidaksamaan (2.3) berikut.

$$w * x_i + b >= 0 \, untuk \, y_i = +1 \quad (2.3)$$

Dimana:

$x_i$ = data input

$y_i$ = label yang diberikan

w = nilai dari bidang normal

b = posisi bidang relatif terhadap pusat koordinat

Parameter w dan b adalah parameter yang akan dicari nilainya. Bila label data $yi$ = -1, maka pembatas menjadi persamaan (2.4) berikut:

$$w * x_i + b \leq -1 \quad (2.4)$$

Bila label data $yi$ = +1, maka pembatas menjadi persamaan (2.5) berikut:

$$w * x_i + b \geq +1 \quad (2.5)$$

Margin terbesar dapat dicari dengan cara memaksimalkan jarak antar bidang pembatas kedua kelas dan titik terdekatnya, yaitu $\frac{2}{|w|}$. Hal ini dirumuskan sebagai permasalahan *quadratic programming* (QP) *problem* yaitu mencari titik minimal persamaan (2.6) dengan memperhatikan persamaan (2.7) berikut:

$$min \, \tau(w) = \frac{1}{2}||w||^2 \quad (2.6)$$

$$yi(w * xi + b) - 1 \geq 0, (i = 1, .., n) \quad (2.7)$$

Permasalahan ini ini dapat dipecahkan dengan berbagi teknik komputasi. Lebih mudah diselesaikan dengan mengubah persamaan (2.8) berikut:

$$L(w,b,a) = 1/2||w||2 - \sum_{i=1}^{n} a_i(y_i((\mathrm{w}^T\mathrm{x}_i + \mathrm{b}) - 1)) \quad (2.8)$$

Dimana $a_i$ adalah *lagrange multiplier* yang bernilai nol atau positif ($a_i \geq 0$). Nilai optimal dari persamaan (2.9) dapat dihitung dengan meminimalkan L terhadap $w$, $b$ dan $a$. Dapat dilihat pada persamaan (2.10) berikut:

$$\begin{aligned} Min\ L(w,b,a) \quad & (2.9) \\ &= 1/2||w||2 - \sum_{i=1}^{n} a_i y_i((\mathrm{w}^T\mathrm{x}_i + \mathrm{b}) - 1)) \\ &+ \sum_{i=1}^{n} a_i \end{aligned}$$

Model persamaan (2.9) diatas merupakan model primal *Lagrange.* Sedangkan dengan memaksimalkan L terhadap $a_i$, persamannya menjadi persamaan (2.10) berikut [19]:

$$\max_{\alpha} L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j x_i^T x_j \quad (2.10)$$

Dengan memperhatikan persamaan (2.11) dan (2.12)) berikut :

$$0 \leq a_i \leq C, i = 1, \ldots, m \quad (2.11)$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad (2.12)$$

Untuk mencari nilai $x_i$ dan $y_i$ dapat dilakukan ketika sudah didapatkan nilai ekstraksi ciri dari pembobotan dan inisialisasi kelas . Hasil dari pembobotan ekstraksi diubah ke dalam bentuk format data svm, sedangkan data kelas menjadi label data SVM. Setelah parameter $a_i$ didapatakan, kemudian masukkan ke persamaan (2.13) berikut:

$$w = \sum \alpha_i y_i \overline{x_i} \quad (2.13)$$

Hasil yang didapatkan menggunakan persamaan (2.13), selanjutnya digunakan persamaan (2.14) untuk mendapatkan nilai dan *b.*

$$y = wx + b \quad (2.14)$$

Sedemikian sehingga didapatkanlah nilai *w* dan *b* atau nilai *hyperplane* untuk

mengklasifikasikan kedua kelas. Berikut ini adalah beberapa fungsi kernel yang umum digunakan yaitu :

a. Kernel Linear
$K(x_i, x) = x_i^T x$

b. Polynomial
$K(x_i, x) = (-\gamma x_i^T x + r)p, \gamma > 0$

c. Radial Basis Function
$K(x_i, x) = exp(-\gamma \mid x_i - x \mid 2), \gamma > 0$

d. Sigmoid Kernel
$K(x_i, x) = tanh(\gamma x_i^T + r)$.

### 2.3.2 Hyperbolic Support Vector Classification

Pada metode HSVM ini prosesnya akan terjadi di mana titik data terletak pada ruang hiperbolik. Secara khusus proses ini akan mengubah nilai X kedalam bentuk hiperboloid dan membiarkan dimensi menjadi fungsi ruang hiperbolik. Pada metode ini, cara untuk menentukan hyperplane keputusan adalah sebagai berikut :

$$H = \{h(x; w) : w \in R^{n+1}, w * w < 0\} \quad (2.15)$$

Dimana

$$h(x; w) = \{1 \quad w * x > 0 \ atau \ -1 \quad w * x \leq 0\} \quad (2.16)$$

Fungsi keputusan ini memiliki hyperplane keputusan $w * x = 0$, yang merupakan hyperplane n-dimensi dalam $R^{n+1}$. Dengan demikian, keputusan hyperplane yang sesuai dalam $L^n$ yang diperoleh dari persimpangannya setara ruang hiperbolik dari hyperplane linier, yang juga dapat dilihat sebagai gabungan dari kurva geodesi. Metode ini memberikan contoh hyperplanes keputusan linier dalam ruang hiperbolik. Misalnya untuk $L^2$, yang dapat divisualisasikan sebagai hiperboloid atas dalam 3D. Menariknya, pada formulasi metode ini menghulangkan istilah bias (b) karena setiap hiperbola dari codimension satu tercakup oleh parametrization. Masalah klasifikasi margin maksimum dengan ruang fitur hiperbolik X = $L^n$ dengan fungsi jarak (hiperbolik) d, dan fungsi keputusan hiperbolik-linier sebagaimana didefinisikan dengan persamaan (2.24) berikut:

$$Minimasi_{w \in R^{n+1}} \; -\frac{1}{2} w * w \tag{2.17}$$

Dimana

$$y^{(j)}\left(w * x^{(j)}\right) \geq 1 \; dan \; w * w < 0 \tag{2.18}$$

Formulasi soft-margin dari SVM hiperbolik dapat diturunkan dengan melonggarkan batasan pemisahan seperti dalam kasus Euclidean. Daripada memaksakan linier-penalty pada kesalahan klasifikasi. Pada metode ini dilakukan perbaikan pada skala penalti sehingga margin titik terdekat ke batas keputusan (yang diklasifikasikan dengan benar) diatur ke $sinh^{-1}(1)$. Sebagai berikut:

$$Minimasi_{w \in R^{n+1}} \; -\frac{1}{2} w * w + C \sum_{j=1}^{m} \max(0, sinh^{-1}(1) - sinh^{-1}\left(y^{(j)}\left(w * x^{(j)}\right)\right)) \tag{2.19}$$

Dimana

$$w * w < 0 \tag{2.20}$$

Dalam semua percobaan pada bagian tersebut, pada metode ini memilih pendekatan paling sederhana untuk menyelesaikan perumusan SVM hiperbolik di atas melalui proyeksi gradient descent. w awal ditentukan berdasarkan hasil w dari SVM di ruang Euclidean ambien dari model hiperboloid, sehingga $w * x = (w)^T x$ untuk semua x. Hal ini memberikan inisialisasi yang baik untuk optimasi dan memiliki manfaat tambahan untuk meningkatkan stabilitas algoritma tersebut [5].

#### 2.3.2.1 Model Ruang Hyperbolic

Ruang hiperbolik tidak dapat secara isometrik tertanam dalam ruang Euclidean, ada beberapa model geometri hiperbolik yang berguna diformulasikan sebagai bagian dari ruang Euclidean, yang masing-masing memberikan pengartian yang berbeda mengenai sifat-sifat geometri hiperbolik [20]. Standar pada ruang hiperbolik memiliki bebereapa model seperti Hiperboloid dan Poincare. Ruang bernilai real berdimensi $(n + 1)$, $R^{n+1}$, yang dilengkapi dengan produk dalam bentuk sebagai pada persamaan (2.28) berikut.

$$X * y = X_0 y_0 - X_1 y_1 - \cdots - X_n y_n \tag{2.21}$$

Bentuk ini umumnya dikenal sebagai ruang Minkowski. Model hiperboloid n-dimensi $L^n$ berada di dalam $R^{n+1}$,sebagai bagian atas sehubungan dengan produk dalam Minkowski. Pada persamaan (2.29) ini merupakan produk dalam model hiperboloid.

$$L^n = \{x: x = (x_0, \dots, x_n) \in R^{n+1}, x * x = 1, x_0 > 0 \quad (2.22)$$

Jarak antara dua titik di $L^n$ didefinisikan sebagai panjang jalur geodesik pada hiperboloid yang menghubungkan dua titik. Diketahui bahwa setiap kurva geodesik (yaitu, garis hiperbolik) di $L^n$ adalah persimpangan antara $L^n$ dan bidang 2D yang melewati titik asal dalam ruang Euclidean ambient $R^{n+1}$, dan sebaliknya [5]. Proyeksi dari setiap titik $L^n$ ke hyperplane $x_0 = 0$ memberikan model bola Poincaré pada persamaan (2.30).

$$B^n = \{x: x = (x_1; \dots; x_n) \in R^n, \|x\|^2 < 1 \quad (2.23)$$

### 2.3.3 Evaluasi Peforma dengan Confusion Matrix

Akurasi klasifikasi merupakan ukuran ketepatan klasifikasi yang menunjukkan performansi teknik klasifikasi secara keseluruhan. Semakin tinggi akurasi klasifikasi berarti performansi teknik klasifikasi juga semakin baik [21]. Penjelasan mengenai akurasi klasifikasi disajikan pada Tabel 2.1.

**Tabel 2.1** *Confusion Matrix*

| **Aktual** | **Prediksi** | |
|---|---|---|
| | **Positif** | **Negatif** |
| **Positif** | TP | FN |
| **Negatif** | FP | TN |

Keterangan :

TP : (*True Positive*) merupakan jumlah prediksi benar pada kelas positif.

FP : (*False Positive*) merupakan jumlah prediksi salah pada kelas positif.

FN : (*False Nefative*) merupakan jumlah prediksi salah pada kelas negatif.

TN : (*True Negative*) merupakan jumlah prediksi benar pada kelas negatif.

Berdasarkan nilai dari confusion matrix tersebut dapat dihitung nilai akurasi, presisi, dan recall dengan persamaan sebagai berikut :

$$Akurasi = \frac{TP+TN}{TP+FP+FN+TN}\ x\ 100\% \quad (2.30)$$

$$Presisi = \frac{TP}{TP+FP}\ x\ 100\% \quad (2.31)$$

$$Recall = \frac{TP}{TP+TN}\ x\ 100\% \quad (2.32)$$

### 2.3.4 Cross Validation

Cross-validation, atau bisa disebut validasi silang merupakan sebuah teknik validasi model untuk menilai bagaimana hasil statistik analisis akan menggeneralisasi kumpulan data independen. Teknik ini biasa digunakan untuk melakukan prediksi dan memperkirakan seberapa akurat sebuah model ketika dijalankan dalam praktiknya. Dalam sebuah masalah prediksi, sebuah model biasanya diberikan kumpulan data (dataset) yang diketahui untuk digunakan dalam menjalankan pelatihan (dataset pelatihan) terhadap model yang diuji (pengujian dataset) [22].

#### 2.3.4.1 K-folds Cross Validation

N-folds Cross Validation adalah salah satu teknik untuk validasi yang sangat populer digunakan. Metode Validasi dengan n-folds sangat cocok digunakan untuk kasus data yang jumlah sampelnya terbatas. Untuk melakukan proses klasifikasi tentunya data dibagi ke dalam training dan testing. Ketika data yang digunakan untuk training sangat sedikit kemungkinan adalah data yang digunakan sangat representatif. Dalam n-folds cross validation, data (D) dibagi ke dalam n subsets D1, D2, D3, ..., Dn dengan jumlah yang sama. Data yang digunakan untuk training adalah subsets data n-1 yang dikombinasikan secara bersama-sama dan kemudian diaplikasikan untuk sisa satu subsets data sebagai hasil testing. Proses ini diulangi sebanyak n subsets dan hasil akurasi klasifikasi yaitu hasil rata-rata dari setiap data training dan testing. N-folds yang biasa digunakan adalah 3, 5, 10 dan 20 [23].

## 2.4 Python Programming

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Dibuat oleh Guido van Rossum dan dirilis pertama kali pada tahun 1991, filosofi desain Python menekankan keterbacaan kode dengan penggunaan spasi spasi yang signifikan. Konstruk pada bahasanya dan pendekatan berorientasi objek bertujuan untuk membantu programmer menulis kode yang jelas dan logis untuk proyek skala kecil dan besar [24]. Python diketik dan dikumpulkan secara dinamis. Ini mendukung berbagai paradigma pemrograman, termasuk pemrograman terstruktur (khususnya, prosedural), berorientasi objek, dan fungsional. Python juga diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif [25].

Python adalah bahasa pemrograman multi-paradigma. Pemrograman berorientasi objek dan pemrograman terstruktur didukung penuh, dan banyak fitur-fiturnya mendukung pemrograman fungsional dan pemrograman berorientasi aspek. Banyak paradigma lain didukung melalui ekstensi, termasuk desain berdasarkan kontrak dan pemrograman logika. Python menggunakan pengetikan dinamis dan kombinasi penghitungan referensi [26].

### 2.4.1 Kelebihan Bahasa Pemprograman Python

Berikut ini adalah beberapa kelebihan bahasa pemprograman python :

1. Library yang luas dan banyak

Python memiliki library luas dengan beragam modul yang siap untuk Anda gunakan. Di dalamnya terdapat beragam kode untuk beragam keperluan seperti regular expressions, documentation-generation, unit-testing, databases, CGI, email, dan masih banyak lagi.

2. Meningkatkan produktivitas developer

Bahasa yang sederhana serta library yang luas dapat membuat developer menjadi lebih produktif. Selain itu, dengan Python programmer hanya perlu menulis kode

lebih sedikit sehingga dapat mempunyai lebih banyak waktu untuk bisa mengerjakan yang lain.

3. Mendukung IoT

Python mendukung Internet of Things (IoT) dengan sangat baik. IoT merupakan teknologi yang dapat menghubungkan benda-benda di sekitar ke dalam sebuah jaringan yang menghubungkan satu dengan yang lain.

4. Embeddable

Pyhton juga dapat ditanam atau disematkan. programmer dapat meletakkan kode Python ke dalam sumber bahasa lain seperti C++. Kemampuan ini memungkinkan untuk menambahkan kemampuan scripting ke dalam bahasa lain.

### 2.4.2 Kekurangan Bahasa Pemrograman Python

Berikut ini adalah beberapa kekurangan dari bahasa pemprograman python :

1. Eksekusi yang lambat

Python merupakan bahasa interpreter yang bekerja dengan menggunakan kompiler. Ketika dijalankan, Pyhton akan bekerja lebih lambat jika dibandingkan dengan bahasa lain. Namun hal ini juga tergantung dari besar atau kecilnya program yang akan dibuat.

1. Lemah dalam komputasi mobile

Bahasa ini lebih cocok digunakan untuk platform desktop dan server tetapi lemah untuk komputasi mobile. Penggunaan Pyhthon kurang cocok untuk pengembangan ponsel dan pengembangan game.

2. Kesalahan Run Time

Python diketik secara dinamis sehingga tidak perlu mendeklarasikan tipe variabel saat menulis kode. Meskipun ini memudahkan developer selama pengkodean, namun dapat meningkatan terjadinya kesalahan pada saat run-time.

3. Kesulitan dalam bahasa lain

Para pengguna Python biasanya akan sangat terbiasa dengan beragam fitur dan library yang luas. Hal ini akan membuat mereka mengalami sedikit masalah ketika belajar atau bekerja dengan bahasa pemrograman yang lain.

### 2.4.3 Python Scikit-learn

Scikit-learn atau juga dikenal sebagai sklearn adalah library pembelajaran mesin (*machine learning*) untuk bahasa pemrograman Python. Library ini mempunyai berbagai fitur seperti klasifikasi, regresi dan algoritma pengelompokan termasuk Support Vector Machine, Random Forest, Gradien Boosting, K-means, dan DBSCAN. Scikit-learn memanfaatkan lingkungan yang kaya ini untuk menyediakan implementasi canggih dari banyak algoritma pembelajaran mesin yang terkenal, sambil mempertahankan antarmuka yang mudah digunakan yang terintegrasi erat dengan bahasa Python. Hal ini membantu dalam meningkatnya kebutuhan untuk analisis data statistik oleh non-spesialis dalam industri perangkat lunak dan web, serta di bidang di luar ilmu komputer, seperti biologi atau fisika. Library ini dirancang untuk beroperasi dengan library lainnya pada Python seperti numpy dan scipy [27].

### 2.4.4 Project Jupyter

Project Jupyter adalah organisasi yang diciptakan untuk mengembangkan *Open-source software* , *open-standards*, dan layanan untuk komputasi interaktif di berbagai bahasa pemrograman. Project Jupyter mendukung lingkungan eksekusi dalam beberapa bahasa. Nama Proyek Jupyter adalah referensi ke tiga bahasa pemrograman inti yang didukung oleh Jupyter, yaitu Julia, Python dan R, dan juga penghormatan terhadap buku catatan Galileo yang mencatat penemuan bulan-bulan Jupiter. Project Jupyter telah mengembangkan dan mendukung produk-produk komputasi interaktif Jupyter Notebook, JupyterHub, dan JupyterLab, versi generasi selanjutnya dari Jupyter Notebook [28].

#### 2.4.4.1 Jupyter Notebook

Jupyter Notebook merupakan lingkungan komputasi interaktif berbasis web untuk membuat dokumen notebook Jupyter. Istilah "notebook" secara sehari-hari dapat membuat referensi ke banyak entitas yang berbeda, terutama aplikasi web

Jupyter, web server Jupyter Python, atau format dokumen Jupyter tergantung pada konteksnya. Dokumen Jupyter Notebook adalah dokumen JSON, mengikuti skema versi, dan berisi daftar sel input / output yang dapat berisi kode, teks (menggunakan Markdown), matematika, plot, dan media kaya, biasanya diakhiri dengan ekstensi ".ipynb" [29].

Jupyter notebook dapat dikonversi ke sejumlah format output standar terbuka, seperti HTML, slide presentasi, LaTeX, PDF, ReStructuredText, Markdown, atau Python. Untuk menyederhanakan visualisasi dokumen notebook Jupyter di web, perpustakaan nbconvert disediakan sebagai layanan melalui NbViewer yang dapat mengambil URL ke dokumen notebook yang tersedia untuk umum, mengonversinya menjadi HTML dengan cepat dan menampilkannya kepada pengguna [30].


UNIVERSITAS KOMPUTER INDONESIA
UNIKOM