

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pengelana wajah akan lebih efektif dan efisien dengan metode yang tepat. Metode dan teknologi yang akan dipakai dan dibahas pada aplikasi adalah *Support Vector Machine* dan *face recognition*. Sistem dan aplikasi pengenalan wajah sebelumnya sudah banyak di buat, tetapi metode dan program aplikasinya yang digunakan berbeda – beda.

Tinjauan pustaka merupakan sebuah acuan utama dalam beberapa studi yang pernah dilakukan yang berkaitan dengan penelitian ini. Terdapat beberapa penelitian yang digunakan sebagai acuan dalam penelitian ini. Beberapa penelitian terdahulu:

1. Pada penelitian sebelumnya yang berjudul “Implementasi Gabor Wavelet dan Support Vector Machine Pada Sistem Pengenalan Wajah 3D”, dalam penelitian ini mengusulkan sistem pengenalan wajah 3D dengan menggunakan metode ekstraksi ciri *Gabor Wavelet* dan klasifikasi dengan *Support Vector Machine (SVM)*. Pada penelitian ini jumlah data wajah yang akan digunakan sebanyak 20 individu. Hasil akurasi yang dicapai dengan menggunakan metode *Gabor Wavelet* sebagai metode ekstraksi ciri dan metode *SVM* untuk klasifikasi yaitu 51,67%. (Nafisa, Novamizanti, & Susatio, 2019)
2. Pada penelitian sebelumnya yang berjudul “*Face Detection System Menggunakan Metode Histogram of Oriented Gradients (HOG) dan Support Vector Machine (SVM)*”, dalam penelitian ini mengusulkan pengembangan *face detection* untuk sistem presensi menggunakan metode *Histogram of Oriented Gradients (HOG)* sebagai ekstraksi cirinya dan *Support Vector Machine (SVM)* digunakan sebagai klasifikasinya. Dalam penelitian ini dataset yang digunakan sebanyak 644 citra positif dan 2.572 citra negatif untuk proses training, 110 citra positif untuk proses testing,

dan 10 citra untuk proses deteksi. Untuk sistem yang dibangun menggunakan *Histogram of Oriented Gradients* dengan block yang terdiri dari 2x2 cell dimana satu cell terdiri dari 8x8 pixel dan menggunakan kernel linear dalam *Support Vector Machine*. Sistem ini menghasilkan nilai f-1 score sebesar 71,442%. (Pranoto, Ramadhani, & Arifianto, 2017)

3. Pada penelitian sebelumnya yang berjudul “Pembangunan Aplikasi Presensi Magang Berbasis Mobile Menggunakan *Face Recognition*”, dalam penelitian ini mengusulkan membangun aplikasi presensi magang berbasis mobile menggunakan face recognition dengan library *Android Face Recognition with Deep Learning* untuk mempermudah mengimplementasi *Viola – Jones* untuk face detection dan *Support Vector Machine* untuk *face recognition* pada aplikasi. Dalam penelitian sistem presensi menggunakan *face recognition* ini mendapatkan keakuratan sebesar 95% dengan set data training dari 20 data teratas oleh 20 orang yang berbeda. (Pramono, Ardanari, S.Si., M.T, & Maslim, S.T., M.T., 2020)
4. Pada penelitian sebelumnya yang berjudul “Sistem Presensi Karyawan Berbasis Pengenalan Wajah Dengan Metode *Support Vector Machine*”, dalam penelitian ini mengusulkan untuk mengembangkan sebuah sistem presensi karyawan suatu perusahaan melalui sistem pengenalan wajah dengan menggunakan metode *Support Vector Machine (SVM)* yang digabungkan dengan metode *Histogram of Oriented Gradient*. Dengan penelitian ini dilakukan harapan kedepannya melalui sistem ini dapat meningkatkan kesadaran karyawan dalam melakukan presensi dan terciptanya keterbukaan informasi pengelolaan SDM antara karyawan dengan manajemen perusahaan. Dalam penelitian hasil pengujian sistem presensi karyawan berbasis pengenalan wajah ini disimpulkan bahwa hasil pengambilan gambar berdasarkan jarak menunjukkan hasil yang paling baik diantara 30 cm – 90 cm. serta pengujian kemiringan didapatkan bahwa sistem ini mampu mendeteksi dan mengenali wajah dengan posisi *Roll* antara -40° ke arah kiri sampai 40° ke arah kanan. Posisi *Yaw* tegak

lurus dengan gravitasi antara -30° sampai 30° . Sedangkan untuk posisi *Pitch* alat mampu mendeteksi dari -30° atau posisi menunduk sampai 40° posisi menengadah. Untuk hasil pengujian presensi didapatkan sistem dapat menemukan kembali informasi dengan baik ditunjukkan dengan hasil *recall* sebesar 77.78%, tingkat kesalahan saat proses identifikasi atau nilai *spesifitas* 32.22%, akurasi sistem 72.78%, dan kepresisian sistem mencapai 70.71%. (Setiyadi, Atabiq, & Aisyah, 2021)

5. Pada penelitian sebelumnya yang berjudul “Presensi Kelas Berbasis Pola Wajah, Senyum dan Wi-Fi Terdekat dengan Deep Learning”, dalam penelitian ini mengusulkan untuk merancang sebuah aplikasi presensi jelas berbasisi pengenalan pola wajah, senyum, dan Wi-Fi terdekat. Dimana penelitian ini menggunakan pendekatan *Deep Learning* dengan arsitektur *CNN FaceNet* yang berfungsi untuk mengenali wajah. Selain itu juga sistem akan memvalidasi presensi dengan kesesuaian lokasi dan waktu. Data lokasi akan diperoleh melalui *SSID* dengan database, dan data waktu akan diambil saat mahasiswa mengirimkan data kehadiran melalui *API*. Penulis akan membuat dalam 3 aplikasi yaitu web, mobile, dan service yang dipasang di komputer mini, yang saling terintegrasi untuk mengirimkan data presensi ke sistem akademik secara otomatis. Kemudian untuk konfirmasi, siswa diwajibkan selfie tersenyum untuk memperkuat validitas kehadiran. Dalam penelitian ini hasil akurasi yang didapatkan dari *testing* model sebesar 92.6% sedangkan untuk *testing live* akurasi sebesar 66.7%. Nilai *testing live* lebih kecil dan cukup jauh dari *testing* model yang dapat diartikan hasil *training* model terlalu *overfitting*. (Miftakhurrokhmat, Rajagede, & Rahmadi, 2021)

Tabel 2. 1. Tinjauan pustaka

Penulis	Obyek	Metode	Tools	Hasil / Keluaran Penelitian
Wan Nadilla Nafisa, Ledy Novamizanti, Eko Susatio (2019)	Pengenalan wajah untuk 3D	<i>Gabor Wavelet dan Support Vector Machine (SVM)</i>	<i>KScan3D</i>	Hasil implementasi pada sistem pengenalan wajah 3D dengan hasil akurasi 51,67% dari 20 data wajah.
Miftahul Bagus Pranoto, Kurniawan Nur Ramdhani, S.T., M.T., Anditya Arifianto, S.T., M.T. (2017)	Sistem untuk face detection	<i>Histogram of Oriented Gradients (HOG) dan Support Vector Machine (SVM)</i>	<i>Matlab</i>	Hasil pengembangan sistem untuk mendeteksi wajah. Hasil akurasi 95,45% dengan menggunakan <i>kernel</i> yang paling optimal yaitu linear.
Ananda Pramono, Patricia Ardani, S.Si., M.T., Martinus Maslim, S.T., M.T. (2020)	Pengenalan wajah untuk sistem presensi pegawai magang	<i>Viola-Jones dan Support Vector Machine (SVM)</i>	<i>Library Android Face Recognition with Deep Learning</i>	Sistem presensi berbasis mobile yang digunakan untuk melakukan presensi pada pegawai magang. Hasil akurasi 95%.
David Setiyadi, Fauzun Atabiq, Siti Aisyah (2021)	Pengenalan wajah untuk sistem presensi karyawan	<i>Support Vector Machine (SVM) dan Histogram of Oriented Gradient (HOG)</i>	<i>Framework python, postgresql</i>	Hasil pengembangan sistem presensi untuk presensi berbasis pengenalan wajah beserta hasil akurasinya 77,78%.
Nadya Viana de Lima, Ledy Novamizanti, Eko Susatio (2018)	Pengenalan wajah untuk 3D	<i>Iterative Closest Point (ICP) dan Support Vector Machine (SVM)</i>	<i>KScan3D</i>	Hasil implementasi pada sistem pengenalan wajah 3D dengan hasil akurasi 97,56% dari 80 citra uji.

2.2 Dasar Teori

2.2.1 Pengenalan Wajah (Face Recognition)

Pengenalan wajah (*face recognition*) adalah salah satu teknologi biometric yang telah banyak diaplikasikan dalam sistem keamanan selain pengenalan retina, sidik jari, iris mata (DewiNoviana & IsmawanFiqih, 2021). Dalam aplikasinya sendiri pengenalan wajah menggunakan kamera untuk menangkap wajah seseorang kemudian dibandingkan dengan basis data yang sudah tersimpan di local komputer dan menemukan basis data wajah yang paling cocok dengan citra masukan tersebut. *Face recognition* adalah teknologi dari komputer yang memungkinkan kita untuk mengidentifikasi atau memverifikasi wajah seseorang melalui sebuah gambar digital.

Pengenalan wajah akan ditinjau dari posisi yang sangat berpengaruh terhadap akurasi pengenalan wajah. Beberapa posisi wajah antara lain melihat kedepan, kanan, dan kiri. Dalam *face recognition* tahapan awal yang sangat penting yaitu *face detection*. Ada 3 tahap dalam pengenalan wajah yaitu:

1. *Detection*, digunakan untuk mengekstraksi pola dalam sebuah gambar lalu membandingkannya. Jika terdapat pola yang sama, maka sistem akan mengasumsikan bahwa ada wajah dalam basis data gambar.
2. *Faceprint creation*, digunakan sebagai ‘cetakan’ atau ‘model’ wajah. Dalam *faceprint* ada 2 cara yaitu:

- a) Pendekatan Geometris (*Geometric Approach*)

Digunakan untuk mengukur jarak dan relasi special antara fitur wajah seperti titik pusat mata, bagian ujung hidung, ataupun garis bibir untuk mengenali wajah.

- b) Pendekatan Fotometrik (*Photometric Approach*)

Digunakan untuk menganalisis foto dan membandingkannya dengan *database* untuk mengenali identitas seseorang berdasarkan statistiknya.

c) Analisis Tekstur Wajah (*Skin Texture Analysis*)

Digunakan untuk memetakan lokasi unik pori – pori, garis atau bercak pada kulit yang berbeda antara seseorang dengan yang lainnya.

3. Verifikasi atau Identifikasi

a) Verifikasi (otentikasi)

Melakukan perbandingan input foto wajah dengan data foto pengguna yang membutuhkan autentikasi. Agar lebih mudah di ingat, dimana perbandingannya 1 x 1.

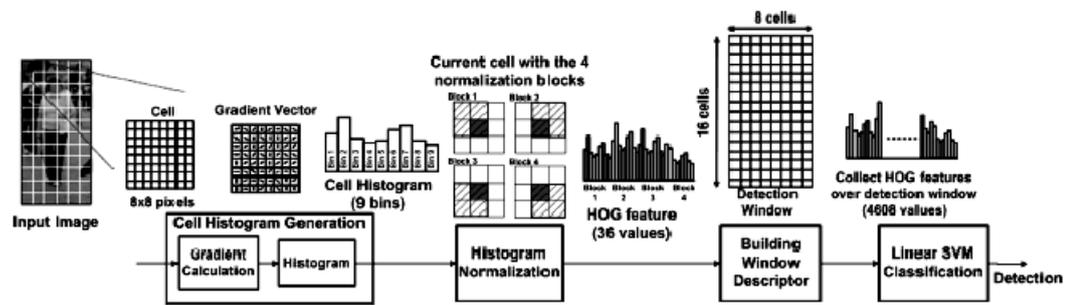
b) Identifikasi

Pada identifikasi ini melakukan perbandingan 1 x N. Jadi, input foto wajah dibandingkan dengan seluruh foto wajah yang ada di dalam *database* untuk menemukan orang yang cocok dengan input foto tersebut (Binus University School of Information Systems, 2022).

Untuk data yang akan digunakan menggunakan gambar siswa di sekolah yang menjadi tempat penelitian. Data gambar yang akan digunakan berasal dari kelas VII, VIII, dan IX. Setelah mendapatkan data gambar tersebut maka akan melakukan tahapan dalam *face recognition*. Ketika sudah selesai melalui tahapan tersebut maka para siswa akan melakukan presensi dengan menggunakan pengenalan wajah.

2.2.2 Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients adalah metode yang digunakan untuk mendeteksi objek, histogram berisi channel – channel arah / *orientasi gradient* dari piksel – piksel pada gambar, dimana penampilan serta bentuk objek dapat diketahui melalui hasil komputasi *gradient* dari citra. *Histogram of Oriented Gradients (HOG)* adalah salah satu metode ekstraksi ciri yang digunakan dalam *image processing* untuk mendeteksi suatu objek seperti manusia. (Ilmi, Novianty, S.T, M.T, & Ahmad, S.T, M.T, 2015)



Gambar 2. 1 Histogram Of Oriented Gradients

(Sumber Gambar : https://www.researchgate.net/figure/Overview-of-Histogram-of-Oriented-Gradients-21_fig1_331860076)

Pada gambar 2.1 merupakan tahapan untuk metode *histogram of oriented gradients (HOG)* sebagai berikut :

(https://www.researchgate.net/figure/Overview-of-Histogram-of-Oriented-Gradients-21_fig1_331860076)

1. Pada tahap pertama akan melakukan proses normalisasi terhadap gambar yang diambil. Dimana proses ini dilakukan untuk memastikan bahwa gambar memiliki skala intensitas yang seragam sehingga dapat diproses dengan baik. Kemudian dari gambar tersebut diekstraksi nilai intensitas dari tiga plane warna yang ada di dalam gambar. Plane warna mengacu pada tiga komponen warna utama yaitu merah, hijau, dan biru atau bisa disebut sebagai RGB. Tujuannya agar memperoleh informasi tentang tingkat kecerahan dan warna dalam gambar yang akan digunakan dalam proses selanjutnya.
2. Langkah berikutnya menghitung gradien setiap elemen gambar dengan membandingkannya dengan piksel yang lain. Dimana gradien menggambarkan perubahan intensitas piksel di sekitarnya. Selain itu gradien juga digunakan untuk mengidentifikasi arah dan kekuatan perubahan intensitas di dalam gambar. Dan dengan menghitung gradien, dapat mengekstraksi informasi tentang arah tekstur dan pola yang ada dalam gambar.
3. Langkah berikutnya melakukan klasifikasi pasangan gradien berdasarkan sumbu saling tegak lurus, yaitu sumbu x dan y, ke dalam kategori frekuensi

kemunculan yang disebut sebagai bin. Dalam konteks ini, bin menggambarkan sebagai rentang nilai sudut gradien. Dimana untuk setiap sel 8 x 8 pada gambar, dihasilkan vector dengan 9 elemen tunggal. Vector ini merepresentasikan distribusi frekuensi kemunculan bin – bin tersebut dalam sel tersebut. Maka, vector 9 elemen ini mengandung informasi tentang arah dan kekuatan gradien dalam sel 8 x 8 tersebut.

4. Untuk mengurangi pengaruh pencahayaan terhadap akurasi klasifikasi maka dilakukan pembuatan blok – blok dengan menggabungkan sel – sel yang intensitasnya telah dinormalisasi. Proses ini, setiap sel dalam gambar diambil dan nilai intensitasnya disesuaikan agar tidak terlalu dipengaruhi oleh perbedaan pencahayaan. Sel – sel yang telah dinormalisasi kemudian digabungkan menjadi blok – blok yang lebih besar. Dengan begitu informasi tentang pola dan fitur wajah dapat tetap terjaga tanpa terpengaruh oleh perubahan pencahayaan yang memungkinkan terjadi.
5. Blok – blok yang telah dibuat kemudian diambil di seluruh jendela deteksi untuk menghasilkan rangkaian descriptor *HOG*. Jendela deteksi merupakan area yang digeser secara sistematis di seluruh gambar untuk mencakup semua potensi objek yang ingin dideteksi. Dengan mengambil blok – blok di dalam jendela deteksi, kita dapat mengumpulkan informasi deskriptif tentang pola orientasi gradien dalam gambar. Kemudian rangkaian dari descriptor *HOG* ini akan digunakan sebagai representasi fitur untuk membedakan objek yang berbeda dalam proses klasifikasi
6. Descriptor *HOG* yang telah dihasilkan dimasukkan ke dalam *SVM (Support Vector Machine)*. *SVM* akan melakukan proses analisis data dengan menggunakan bobot yang telah ditentukan sebelumnya. *SVM* akan belajar dari karakteristik fitur – fitur representative yang terdapat dalam descriptor *HOG* dan menggunakan pengetahuan tersebut untuk mengklasifikasi gambar apakah termasuk dalam kategori yang sudah dibuat. Dengan menggunakan *SVM*, sistem yang dibuat dapat membedakan dan mengenali objek manusia dalam gambar berdasarkan fitur – fitur yang telah diambil dari descriptor *HOG*.

Image Gradient Vector (<https://lilianweng.github.io/posts/2017-10-29-object-recognition-part-1/>)

Pertama akan membedakan istilah – istilah pada tabel 2.2. Walaupun mirip, terkait erat satu sama lain tetapi tidak persis sama.

Tabel 2. 2 Tabel Istilah Dalam HOG

	Derivative	Directional Derivative	Gradient
Value type	Scalar	Scalar	Vector
Definition	Tingkat perubahan pada fungsi $f(x, y, z, \dots)$ pada suatu titik (x_0, y_0, z_0, \dots) yang merupakan kemiringan garis singgung di titik tersebut	Tingkat perubahan sesaat dari $f(x, y, z, \dots)$ dalam arah vector satuan \vec{u}	Ini menunjukkan ke arah laju peningkatan fungsi terbesar, yang berisi semua informasi turunan parsial dari fungsi multivariabel

Dalam pemrosesan citra, tertarik untuk mengukur perubahan warna warni ekstrem satu ke ekstrem lainnya, seperti perubahan dari hitam ke putih pada citra *grayscale*. Untuk itu, memerlukan pengukuran “gradien” pada piksel warna. Kemudian gradien pada gambar yang memiliki sifat diskrit karena setiap piksel akan dianggap independent dan tidak dapat dipisahkan dengan lebih jauh.

Kemudian vector gradien gambar di definisikan sebagai metrics untuk setiap piksel secara individual, yang mengindikasikan perubahan warna piksel pada sumbu x (horizontal) dan sumbu y (vertical). Definisi ini sejajar dengan konsep gradien pada fungsi multi variable kontinu, yang merupakan vector yang terdiri dari turunan parsial terhadap setiap variable. Contohnya adalah jika $f(x, y)$ merupakan warna piksel pada posisi (x, y) , maka vector gradien pikselnya adalah (x, y) didefinisikan sebagai persamaan sebagai berikut ini :

$$\nabla f(x, y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} f(x+1, y) - f(x-1, y) \\ f(x, y+1) - f(x, y-1) \end{bmatrix} \quad (2.1)$$

$\frac{\partial f}{\partial x}$ merupakan turunan parsial pada arah x , yang dihitung sebagai perbedaan warna piksel yang berdekatan di kiri dan kanan target, $f(x+1, y) - f(x-1, y)$. $\frac{\partial f}{\partial y}$ merupakan turunan parsial pada arah y , diukur sebagai $f(x, y+1) - f(x, y-1)$, perbedaan warna antara piksel yang berdekatan di atas dan di bawah target.

Dalam gradien gambar terdapat dua atribut penting yaitu :

- Magnitude, merupakan norma L2 dari vector $g = \sqrt{g_x^2 + g_y^2}$
- Direction, merupakan arctangent dari rasio antara turunan parsial pada dua arah, $\theta = \arctan\left(\frac{g_y}{g_x}\right)$

			90 (x, y+1)	
		105 (x-1, y)	Target Pixel =? (x, y)	55 (x+1, y)
			40 (x, y-1)	

Gambar 2. 2 Untuk menghitung vektor gradien dari piksel

(Sumber gambar : <https://lilianweng.github.io/posts/2017-10-29-object-recognition-part-1/>)

Untuk persamaan vektor gradien yang terdapat pada gambar 2.2 bisa dilihat sebagai berikut ini :

$$\nabla f = \begin{bmatrix} f(x+1, y) - f(x-1, y) \\ f(x, y+1) - f(x, y-1) \end{bmatrix} = \begin{bmatrix} 55 - 105 \\ 90 - 40 \end{bmatrix} = \begin{bmatrix} -50 \\ 50 \end{bmatrix}$$

Maka, hasil yang didapatkan adalah:

- a) Magnitude = $\sqrt{50^2 + (-50)^2} = 70.7107$, dan
 b) Direction = $\left(-\frac{50}{50}\right) = -45^\circ$.

Mengulangi proses perhitungan gradien pada setiap kernel secara iteratif terlalu lambat. Dengan melakuakn penerapan operator *convolution* pada seluruh matriks gambar dan diberikan label sebagai **A** menggunakan salah satu kernel *convolutional* yang dirancang khusus.

Berdasarkan arah - x dari contoh gambar 2.2 dengan menggunakan kernel $[-1,0,1]$ meluncur di atas sumbu x; * merupakan operator dalam konvolusi :

$$G_x = [-1,0,1] * [105,255,55] = -105 + 0 + 55 = -50$$

Demikian pula, pada arah y, persamaan kernel $[+1,0,-1]^T$:

$$G_y = [+1,0,-1]^T * \begin{bmatrix} 90 \\ 255 \\ 40 \end{bmatrix} = 90 + 0 - 40 = 50$$

2.2.3 Support Vector Machine (SVM)

Support Vector Machine merupakan salah satu metode yang digunakan untuk klasifikasi citra dan termasuk dalam jenis *supervised learning*. Pada dasarnya *SVM* bekerja pada *linear classifier* yang kemudian terus dikembangkan untuk bekerja *non-linear* dengan memanfaatkan konsep kernel *trick* di ruang berdimensi tinggi. Cara kerja metode ini dengan menggunakan sebuah fungsi atau *hyperplane* untuk memisahkan dua buah kelas pola, yang mana *SVM* akan menerima *hyperplane* yang optimal untuk memisahkan dua kelas pola ditemukan dengan mengukur jarak margin *hyperplane* dan mencari titik maksimalnya (Nafisa, Novamizanti, & Susatio, 2019).

Support Vector Machine (SVM) adalah salah satu metode dalam *supervised learning*. *SVM* dapat digunakan untuk mengklasifikasikan data

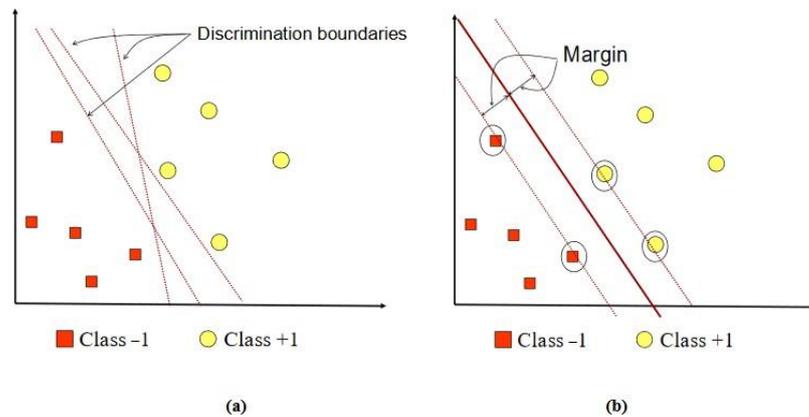
(*Support Vector Classification*) dan melakukan regresi (*Support Vector Regression*). *SVM* memiliki konsep yang matang dan jelas secara matematis dibandingkan dengan metode klasifikasi lainnya. Kemudian *SVM* dapat mengatasi masalah klasifikasi dan regresi dengan data yang linear atau nonlinear.

SVM mencari garis pemisah terbaik antara kelas – kelas data dengan memaksimalkan jarak di antara kelas – kelas tersebut. Garis pemisah ini disebut dengan *hyperplane* dan berfungsi untuk memisahkan kelas – kelas tersebut. *Support Vector Machine* dikembangkan oleh Boser, Guyon dan Vapnik pada tahun 1992 (PranotoMiftahul, RamadhaniKurniawan, ArifiantoAnditya, 2017). Metode ini berprinsip pada *Structural Risk Minimization (SRM)*, yang memiliki tujuan untuk menemukan *hyperplane* terbaik yang memisahkan dua kelas data dengan meminimalkan kesalahan umum yang mungkin terjadi (RivanMuhammad, ArmanMolavi, IrsyadHafiz, PrameswaraReynald, 2022).

Kelebihan dari metode *SVM* adalah kemampuan untuk menangani model yang nonlinear dan kompleks. *SVM* hanya menyimpan sebagian kecil dari data latih saat dilakukannya pelatihan. Meskipun waktu pelatihan dari metode ini cenderung lambat, namun metode ini sangat akurat (Lima, Novamizanti, & Susatio, 2019). Selain itu *SVM* juga efektif dalam mengatasi dataset dengan jumlah fitur yang besar. Namun, metode ini memiliki kelemahan sensitivitas terhadap parameter yang bisa mempengaruhi kinerjanya, dan memerlukan waktu komputasi yang lama untuk dataset yang sangat besar. Metode *Support Vector Machine* juga memiliki kemampuan generalisasi yang baik, artinya dapat menghasilkan prediksi yang baik meskipun dengan sedikit sampel data. Tetapi ini tergantung dalam kompleksitas dan variasi data yang dihadapi.

Konsep *SVM* dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah class pada input space. Pada gambar 2.3 (a) terlihat bahwa ada beberapa *pattern* yang merupakan anggota dari dua buah class: positif (dinotasikan dengan +1) dan negative (dinotasikan dengan -1). *Pattern* yang tergabung pada class negative disimbolkan dengan lingkaran. Proses pembelajaran dalam problem klasifikasi diterjemahkan

sebagai upaya menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Berbagai alternatif garis pemisah (*discrimination boundaries*) (SetiyadiDavid, AtabiqFauzun, AisyahSiti, 2021).



Gambar 2. 3 *Hyperplane SVM sebagai pemisah kelas positif dan negative*

Pada gambar 2.3 diperlihatkan bahwa beberapa *pattern* yang merupakan bagian dari anggota dari dua buah class yaitu +1 dan -1. *Pattern* yang bergabung pada class -1 diberi simbol dengan warna merah kotak sedangkan untuk class +1 diberi simbol dengan warna kuning lingkaran. Problem yang terjadi pada klasifikasi tersebut dapat dijelaskan dengan usaha menemukan garis *hyperplane* yang memisahkan antara kedua kelompok tersebut. Garis solid pada gambar 2 menunjukkan bahwa *hyperplane* yang baik yaitu yang terdapat pada tengah – tengah kedua class, sedangkan titik merah dan kuning yang berada di dalam lingkaran hitam (gambar 2.3 (b)) dinamakan *support vector* (ParapatIndri, FurqonMuhammad, Sutrisno, 2018).

Misalkan diberikan 2 atribut input yaitu A_1 dan A_2 , himpunan $W = \{w_1, w_2, \dots, w_d\}$; W adalah bobot vector / *weight*, d adalah jumlah atribut, dan tupel *training* $X = (x_1, x_2)$ dimana x_1 dan x_2 merupakan nilai dari atribut A_1 dan A_2 , maka fungsi *hyperplane* dapat kita lihat pada persamaan berikut ini :

$$f(x) = W \cdot X + b \quad 2.2$$

Dimana $W, X \in \mathbb{R}^d$ dan b adalah bias yang berupa scalar (hamel,2009)

Hyperplane yang terletak diantara dua set objek dari kelas positif ($y_1 = +1$) dan kelas negatif ($y_2 = -1$) dapat dilihat pada persamaan :

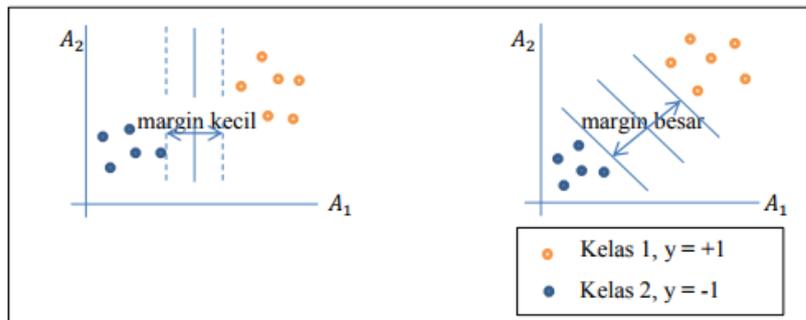
$$H_1 : x_i w + b \geq 1 \text{ untuk } y_1 = +1 \quad (2.3)$$

$$H_2 : x_i w + b \leq -1 \text{ untuk } y_2 = -1 \quad (2.4)$$

Penggabungan dari persamaan (2.3) dan (2.4) menghasilkan pertidaksamaan seperti dibawah ini :

$$y_i(x_i w + b) \geq 1, \text{ untuk } \forall i = 1, 2, \dots, n \quad (2.5)$$

Dengan n merupakan jumlah dari data *training*.



Gambar 2. 4 Margin Besar dan Margin Kecil

Pada gambar 2.4 terdapat margin antara dua kelas dapat dihitung dengan mencari jarak antara kedua *hyperplane* H_1 atau H_2 . Masing – masing tuple pelatihan yang jatuh pada *hyperplane* H_1 atau H_2 yang memenuhi persamaan (2.3) disebut *support vector*. Jarak terdekat suatu titik di bidang H_1 terhadap pusat dapat dihitung dengan meminimalkan $x^T x$ dengan memperhatikan kendala $H_1 : x_i w + b \geq 1$ untuk $y_1 = +1$. Dengan menggunakan bentuk fungsi *Lagrange* dan turunan pertamanya diperoleh :

$$\min x^T x - \lambda(w^T x + b - 1)$$

$$\frac{d}{dx}(x^T x - \lambda(w^T x + b - 1)) = 0$$

$$\Rightarrow 2x - \lambda w = 0$$

$$\Rightarrow x = \frac{\lambda}{2} w$$

Dengan mensubstitusikan x ke bidang $H_1: x_i w + b = 1$ maka diperoleh :

$$\begin{aligned} \frac{\lambda}{2} w^T w + b &= 1 \\ \Rightarrow \lambda &= \frac{2(1-b)}{w^T w} \end{aligned}$$

Sehingga dengan mensubstitusikan kembali λ pada x maka diperoleh :

$$\begin{aligned} x &= \frac{(1-b)}{w^T w} \\ w^T w &= \frac{(1-b)^2}{(w^T w)^2} w^T w = \frac{(1-b)^2}{w^T w} \end{aligned}$$

Maka jarak H_1 ke pusat adalah :

$$\|x\| = \sqrt{w^T w} = \frac{\sqrt{(1-b)^2}}{w^T w} = \frac{(1-b)}{\|x\|}$$

Untuk mencari jarak terdekat suatu titik pada bidang H_2 terhadap titik pusat dapat dihitung dengan meminimalkan $w^T w$ dengan memperhatikan kendala $H_2: x_i w + b \leq 1$ untuk $y_2 = -1$. Dengan menggunakan bentuk fungsi *lagrange* maka turunan pertama diperoleh :

$$\begin{aligned} \min x^T x + \lambda(w^T x + b + 1) \\ \frac{d}{dx} (x^T x + \lambda(w^T x + b + 1)) &= 0 \\ \Rightarrow 2x + \lambda w &= 0 \\ \Rightarrow x &= -\frac{\lambda}{2} w \end{aligned}$$

Dengan mensubstitusikan x ke bidang $H_2: x_i w + b = -1$ maka diperoleh :

$$-\frac{\lambda}{2} w^T w + b = -1$$

$$\Rightarrow \lambda = \frac{2(b+1)}{w^T w}$$

Sehingga dengan mensubstitusikan kembali λ pada x maka diperoleh :

$$x = \frac{(-b-1)}{w^T w}$$

$$w^T w = \frac{(-b-1)^2}{(w^T w)^2} w^T w = \frac{(-b-1)^2}{w^T w}$$

Maka jarak H_1 ke pusat adalah :

$$\|x\| = \sqrt{w^T w} = \frac{\sqrt{(-b-1)^2}}{w^T w} = \frac{(-b-1)}{\|x\|}$$

Maka untuk mendapatkan margin maksimum dengan memaksimalkan jarak antara H_1 dan H_2 :

$$\left| \frac{(1-b)}{\|w\|} - \frac{(-b-1)}{\|w\|} \right| = \frac{2}{\|w\|}$$

Oleh karena itu memaksimalkan $\frac{1}{\|w\|}$ sama dengan meminimumkan $\|w\|^2$ dan untuk menyederhanakan penyelesaian ditambahkan factor $\frac{1}{2}$. Untuk model persamaannya menjadi :

$$\min \frac{1}{2} \|w\|^2$$

$y_i(x_i w + b) \geq 1, \text{ untuk } \forall i = 1, 2, \dots, n$ (n : jumlah data *training*).

Untuk menyelesaikan persamaan diatas, secara komputasi lebih sulit dan perlu waktu lama. Maka diperkenalkan **pengali Langrangian** α_i , dengan $i = 1, 2, \dots, n$. Sehingga model permasalahan sebelumnya dapat ditulis :

$$\min L_p = \frac{1}{2} \|w\|^2 + \alpha_i (0 - \sum_{i=1}^n [y_i(x_i \cdot w + b) - 1])$$

$$L_p = \frac{1}{2} \|w\|^2 + 0 - \alpha_i (\sum_{i=1}^n [y_i(x_i \cdot w + b) - 1])$$

$$L_p = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i(x_i \cdot w + b) - 1]$$

$$L_p = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i y_i (w + b) + \sum_{i=1}^n \alpha_i \quad (2.6)$$

Solusi dapat diperoleh dengan meminimalkan L_p terhadap variable w, b (*primal variabel*) :

$$\frac{\partial L_p}{\partial b} = 0$$

$$\sum_{i=1}^n \partial_i y_i = 0 \quad (2.7)$$

dan

$$\frac{\partial L_p}{\partial w} = 0$$

$$w - \sum_{i=1}^n \partial_i y_i x_i = 0$$

$$w = \sum_{i=1}^n \partial_i y_i x_i \quad (2.8)$$

Dan menghasilkan L_D terhadap variable α_i (*dual variables*). Untuk mendapatkan problem dual (L_D). Substitusikan persamaan (2.7) dan (2.8) ke persamaan (2.5) :

$$maks L_D = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i(x_i \cdot w + b) - 1]$$

$$\|w\|^2 = (w \cdot w)$$

$$= \left(\sum_{i=1}^n \alpha_i y_i x_i \cdot \sum_{j=1}^n \alpha_j y_j x_j \right)$$

$$= \underbrace{\sum_{i=1, j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j}_A \quad (2.9)$$

dan

$$\begin{aligned}
\sum_{i=1}^n \alpha_i [y_i(x_i \cdot w + b) - 1] &= \sum_{i=1}^n \alpha_i y_i x_i w + \sum_{i=1}^n \alpha_i y_i b - \sum_{i=1}^n \alpha_i \\
&= \sum_{i=1}^n \alpha_i y_i x_i \sum_{j=1}^n \alpha_j y_j x_j + 0 - \sum_{i=1}^n \alpha_i \\
&\quad \underbrace{\sum_{i=1, j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j}_A - \underbrace{\sum_{i=1}^n \alpha_i}_B
\end{aligned} \tag{2.10}$$

Sehingga diperoleh

$$\begin{aligned}
L_D &= \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i(x_i \cdot w + b) - 1] \\
&= \frac{1}{2} A - (A - B) \\
&= \frac{1}{2} A - A + B \\
&= -\frac{1}{2} A + B \\
&= B - \frac{1}{2} A \\
&= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j
\end{aligned} \tag{2.11}$$

$$\sum_{i=1}^n \alpha_i y_i = 0, \text{ dan } 0 \leq \alpha_i, i = 1, 2, \dots, n$$

Dengan demikian, dapat diperoleh nilai α_i yang akan digunakan untuk menentukan w dan b dengan rumus :

$$\begin{aligned}
w &= \sum_{i=1}^n \alpha_i y_i x_i \\
b &= y_i - w x_i
\end{aligned} \tag{2.12}$$

Terhadap nilai α_i untuk setiap data pelatihan. Data pelatihan yang memiliki nilai $\alpha_i > 0$ adalah *support vector*. Dengan demikian fungsi keputusan yang dihasilkan hanya dipengaruhi oleh *support vector*.

Berdasarkan formula *lagrangian*, MMH dapat didefinisikan sebagai *boundary decision*:

$$f(x_d) = \sum_{i=1}^l \alpha_i y_i x_i + b \quad (2.13)$$

Dimana y_i adalah label kelas *support vector* X_i, X_z adalah data uji, α_i dan b adalah parameter numerik yang ditentukan secara otomatis oleh optimasi algoritma *SVM* dan l adalah jumlah *support vector*.

Dalam klasifikasi kadang – kadang dijumpai bidang pemisah yang tidak bisa diambil dengan linier sehingga diperlukan penyelesaian khusus untuk permasalahan ini. Untuk data – data yang tidak dapat dipisahkan secara linier tersebut ditambahkan variable *slack* $\xi_i \geq 0$ ke pertidaksamaan (2.3) sehingga kendala dan fungsi tujuan menjadi :

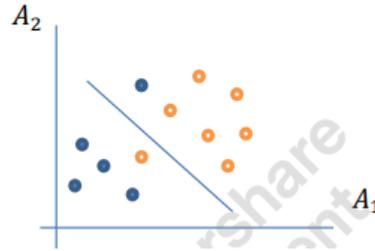
$$y_i(x_i \cdot w + b) - 1 + \xi_i \geq 0, \text{ untuk } \forall_i = 1, 2, \dots, n \quad (2.14)$$

Dengan n merupakan jumlah data *training*.

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (2.15)$$

Dengan $y_i(x_i \cdot w + b) - 1 + \xi_i \geq 0, \xi_i \geq 0, \text{ untuk } \forall_i$

Dengan meminimalkan $\|w\|^2$ persamaan (2.15), ini juga berarti memaksimalkan margin antara dua kelas. Hal ini dilakukan untuk mengurangi jumlah kesalahan klasifikasi (*miss classification error*) yang dinyatakan dengan adanya variable *slack* ξ_i penggunaan variable *slack* ini adalah untuk mengatasi kasus ketidaklayakan (*infeasibility*) dari pembahasan (*constraints*) dengan cara memberikan penalti untuk data yang tidak memenuhi pembahatas tersebut. Pinalti ini dinyatakan dengan C (Vijayakumar dan Wu, 1999).



Gambar 2.5 Sebaran Data yang terpisah secara nonlinear

Selanjutnya, bentuk *primal problem* pada persamaan (2.6) berubah menjadi

$$\min L_p = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \{y_i(x_i \cdot w + b) - 1 + \xi_i\} - \sum_{i=1}^n q_i \xi_i \quad (2.16)$$

Dengan cara yang sama dengan penurunan persamaan dual problem pada data linier, maka persamaan dual problem untuk data nonlinear adalah sebagai berikut :

$$\min L_p = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \{y_i(x_i \cdot w + b) - 1 + \xi_i\} - \sum_{i=1}^n q_i \xi_i$$

$$\begin{aligned} \|w\|^2 &= (w \cdot w) \\ &= \left(\sum_{i=1}^n \alpha_i y_i x_i \cdot \sum_{j=1}^n \alpha_j y_j x_j \right) \\ &= \sum_{i=1, j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j \end{aligned}$$

Dan

$$\begin{aligned} &C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \{y_i(x_i \cdot w + b) - 1 + \xi_i\} - \sum_{i=1}^n q_i \xi_i \\ &= C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i y_i x_i w + \sum_{i=1}^n \alpha_i y_i x_i b - \sum_{i=1}^n \alpha_i + \sum_{i=1}^n q_i \xi_i \\ &\quad - \sum_{i=1}^n q_i \xi_i \end{aligned}$$

$$\begin{aligned}
&= C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i y_i x_i \sum_{i=1}^n \alpha_i y_i x_i + 0 - \sum_{i=1}^n \alpha_i + \sum_{i=1}^n q_i \xi_i \\
&\quad - \sum_{i=1}^n q_i \xi_i \\
&= C \sum_{i=1}^n \xi_i - \sum_{i=1, j=1}^n \alpha_i y_i x_i \alpha_j y_j x_j - \sum_{i=1}^n \alpha_i
\end{aligned}$$

Sehingga diperoleh :

$$\begin{aligned}
L_p(w, x, \xi_i) &= \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \tag{2.17} \\
&\quad - \sum_{i=1}^n \alpha_i \{y_i(x_i \cdot w + b) - 1 + \xi_i\} - \sum_{i=1}^n q_i \xi_i
\end{aligned}$$

Untuk meminimumkan $L_p(w, x, \xi_i)$ digunakan $\frac{\partial L_p}{\partial b} = 0, \frac{\partial L_p}{\partial w} = 0$, dan $\frac{\partial L_p}{\partial \xi_i} = 0$ dengan $i = 1, 2, \dots, n$. Maka dari itu

$$\frac{\partial L_p}{\partial b} = \sum_{i=1}^n \alpha_i y_i = 0$$

$$\frac{\partial L_p}{\partial w} = w - \sum_{i=1}^n \alpha_i y_i x_i = 0$$

Dengan $w = \sum_{i=1}^n \alpha_i y_i x_i$

$$\frac{\partial L_p}{\partial \xi_i} = C - \alpha_i - q_i = 0$$

Akibatnya, $C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n q_i \xi_i = 0$

Dengan memanfaatkan turunan parsial L_p terhadap w, b, ξ_i , masalah dual menjadi :

$$\begin{aligned}
\text{Maks } L_D &= \frac{1}{2} \left(\sum_{i=1}^n \alpha_i y_i x_i \right) \left(\sum_{j=1}^n \alpha_j y_j x_j \right) + C \sum_{i=1}^n \xi_i \\
&\quad - \sum_{i=1}^n \alpha_i y_i \left(\left(\sum_{i=1}^n \alpha_i y_i x_i \right) \cdot x_i + b \right) + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i \xi_i \\
&\quad - \sum_{i=1}^n q_i \xi_i
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i y_i \alpha_j y_j (x_i \cdot x_j) - \sum_{i=1}^n \sum_{j=1}^n \alpha_i y_i \alpha_j y_j (x_i \cdot x_j) \\
&\quad - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \xi_i \\
&= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i y_i \alpha_j y_j (x_i \cdot x_j) \tag{2.18}
\end{aligned}$$

Dengan kendala :

$$\sum_{i=1}^n \alpha_i y_i = 0$$

Karena

$$\frac{\partial L_p}{\partial \xi_i} = C - \alpha_i - q_i = 0$$

Dan semua nilai q_i tidak diperhitungkan pada fungsi tujuan, maka kendalanya menjadi

$$0 \leq \alpha_i \leq C, \text{ untuk } i = 1, 2, \dots, n$$

Bias yang sebelumnya telah didefinisikan sebagai b , dapat didefinisikan secara implisit sebagai atribut tambahin. Sehingga data *input* $X_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ menjadi $X_j^T = [x_{i1}, x_{i2}, \dots, x_{im} \lambda]$ dan vector bobot w menjadi $w^T = [w_{i1}, w_{i2}, \dots, w_m, b/\lambda]$ dengan λ merupakan konstanta.

Bentuk persamaan dari permasalahan menjadi meminimumkan :

$$L_p = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i (1 - y_i) (x_i^T \cdot w^T) (-\xi_i) + \sum_{i=1}^n q_i (-\xi_i)$$

Untuk meminimumkan L_p dengan menggunakan $\frac{\partial L_p}{\partial w^T} = 0$, dan $\frac{\partial L_p}{\partial \xi_i} = 0$ dengan $i = 1, 2, \dots, n$. Maka :

$$\frac{\partial L_p}{\partial w^T} = w^T - \sum_{i=1}^n \alpha_i y_i X_i = 0 \Rightarrow w = \sum_{i=1}^n \alpha_i y_i X_i$$

$$\frac{\partial L_p}{\partial \xi_i} = C - \alpha_i - q_i = 0$$

Bentuk dual dari masalah itu adalah :

$$\begin{aligned} L_p &= \frac{1}{2} \left(\sum_{i=1}^n \alpha_i y_i X_i \right) \left(\sum_{j=1}^n \alpha_j y_j X_j^T \right) + C \sum_{i=1}^n \xi_i \\ &\quad + \sum_{i=1}^n \alpha_i \left(1 - y \left(\sum_{j=1}^n \alpha_j y_j X_j \cdot X_j^T \right) - \xi_i \right) + \sum_{i=1}^n q_i (-\xi_i) \\ &= \frac{1}{2} \sum_{i=1}^n \alpha_i y_i X_i \sum_{j=1}^n \alpha_j y_j X_j^T + \sum_{i=1}^n \alpha_i \left(y_j \left(\sum_{j=1}^n \alpha_j y_j X_j \cdot X_j^T \right) \right) \\ &\quad + \sum_{i=1}^n \alpha_i + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i \xi_i - \sum_{i=1}^n q_i \xi_i \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \alpha_i y_i X_i \sum_{j=1}^n \alpha_j y_j X_j^T \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (X_i \cdot X_j^T) \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(X_i, X_j^T) \end{aligned}$$

Dengan kendala

$$0 \leq \alpha_i \leq C, i = 1, 2, \dots, n$$

2.3 Tools

2.3.1 Python, Jupyter Notebook, dan Flask

Python merupakan salah satu bahasa pemrograman yang bersifat interpreter, interaktif, object – oriented dan dapat beroperasi hampir di semua platform, seperti Linux, Mac, Windows, dan lainnya (digitalskola, 2022). Python termasuk dalam salah satu bahasa pemrograman yang mudah dipelajari karena sintaksnya yang jelas dan elegan, yang dikombinasikan dengan menggunakan

modul – modull siap pakai dan struktur data tingkat tinggi dan efisien. Selain itu juga banyak digunakan dalam aplikasi web, pengembangan perangkat lunak, ilmu data, dan *machine learning* (ML) (Amazon, 2023).

Bahasa pemrograman python sendiri diciptakan di Belanda pada tahun 1990 oleh Guido van Rossum dan namanya diambil dari acara tv kesukaan Guido yaitu Monty Python’s Flying Circus. Kemudian python digunakan sebagai bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintaks intuitif dan memiliki pustaka yang luas.



Gambar 2. 6. *Logo Python*

(Sumber Gambar : <https://www.python.org/community/logos/>)

Jupyter Notebook merupakan aplikasi web gratis yang populer untuk mengelola data di Python dan banyak digunakan oleh data scientist untuk membuat dan membagikan dokumen berisi kode, proses perhitungan, analisis, visualisasi, dan pengelolaan data. Jupyter Notebook memungkinkan untuk mengintegrasikan antara kode dengan output di dalam satu dokumen secara interaktif (DqLab, Jupyter Notebook : Tools Penting Data Scientist, 2023).



Gambar 2. 7. *Jupyter Notebook*

(Sumber gambar : <https://jupyter.org/>)

Flask salah satu *framework* yang dimiliki oleh Python. Flask merupakan sebuah *micro – framework* berbasis bahasa Python yang tidak memiliki banyak

tools dan *library*. Untuk membuat pengembangan lebih efisien maka digunakan Flask. Flask diterapkan karena dapat berjalan pada program yang berkapasitas energi kecil serta dengan memory yang rendah sehingga tidak memerlukan *resource* yang besar. Walaupun *framework* Flask ringan tapi fungsinya masih bisa berjalan sesuai yang dibutuhkan. Di dalam Flask terdapat 2 komponen yaitu *Werkzeug* memiliki peran sebagai menyediakan *routing*, *debugging* dan *web server interface gateway (WSGI)* dan *Jinja2* berperan sebagai *template engine* (DqLab, 2021).



Gambar 2. 8 Flask

(Sumber gambar : <https://id.wikipedia.org/wiki/Flask>)

2.3.2 HTML

HyperText Markup Language (HTML) merupakan sebuah bahasa yang menggunakan *markup* atau penanda untuk membuat halaman web. Dimana penanda atau *markup* ini, nanti akan disebut dengan Tag. Dalam membuat sebuah web terdiri dari menentukan layout website, mengatur paragraph, mengatur huruf, membuat list, membuat table, membuat link, formular, dan menyisipkan gambar, audio, dan video (kemdikbud, 2021).



Gambar 2. 9. HTML

(Sumber Gambar : <https://en.wikipedia.org/wiki/HTML>)