

BAB II

TINJAUAN PUSTAKA

Pada penelitian ini dilakukan klasifikasi jenis terhadap ikan cupang menggunakan metode GLCM (*Gray Level Co- Occurrence Matrix*) dan SVM (*Support Vector Machine*) untuk memudahkan para masyarakat dalam membedakan ikan cupang berdasarkan jenis. Penelitian ini menggunakan dua metode yaitu metode GLCM sebagai metode ekstraksi fitur, dan SVM sebagai metode pengklasifikasinya.

2.1 Penelitian terdahulu

Dalam penelitian ini penulis menggunakan empat penelitian terdahulu yang nantinya akan digunakan sebagai acuan yang tentunya akan sangat relevan dengan judul pada penelitian ini yaitu “Klasifikasi Jenis Ikan Cupang Menggunakan Metode GLCM dan SVM”.

Penelitian pertama berjudul “KLASIFIKASI JENIS KUALITAS KEJU DENGAN MENGGUNAKAN METODE GRAY LEVEL COOCCURRENCE MATRIX (GLCM) DAN SUPPORT VECTOR MACHINE (SVM) PADA CITRA DIGITAL” , dengan penulis dan tahun Reni Anggraini, Bambang Hidayat, Sjafril Darana (2017). Metode yang digunakan adalah GLCM dan SVM, GLCM digunakan sebagai metode untuk ekstraksi fitur dan SVM untuk metode pengklasifikasi. Hasil pengujian dilakukan dengan dataset citra keju sejumlah 48 citra dengan klasifikasi keju sangat layak makan, keju kurang layak makan, dan keju tidak layak makan.

Penelitian kedua berjudul “Wayang Classification Image Using SVM Method and GLCM Feature Extraction” , dengan penulis dan tahun Muhathir, Hamdani Santoso, Diah Ayu Larasati (2021). Metode yang digunakan adalah GLCM dan SVM, GLCM digunakan sebagai metode untuk ekstraksi fitur dan SVM untuk metode pengklasifikasi. Hasil pengujian dilakukan dengan dataset citra wayang masing-masing berjumlah 100 citra untuk tiap kelasnya dengan kelas wayang Arjuna, Batara Wisnu, Gareng, Werkudara, Yudhistira

Penelitian ketiga berjudul “Klasifikasi Jenis Ikan Cupang Menggunakan Metode GLCM Dan KNN” , dengan penulis dan tahun Mohamad Anjas Dwi Akbar, Ahmad Bagus Setiawan, Ratih Kumalasari Niswatin (2021). Metode yang digunakan adalah GLCM dan KNN, GLCM digunakan sebagai metode untuk ekstraksi fitur dan K-NN untuk metode pengklasifikasi. Hasil pengujian dilakukan dengan 2 citra ikan cupang jenis halfmoon dan 2 citra ikan jenis cupang plakat.

Penelitian terakhir yaitu berbentuk skripsi berjudul “Klasifikasi dan Pengenalan Objek Ikan Menggunakan Algoritma Support Vector Machine”, dengan penulis dan tahun Faisal Ferdiansyah(2014). Metode yang digunakan hanya SVM, SVM untuk metode pengklasifikasi. Hasil pengujian dilakukan dengan menggunakan 50 citra untuk tiap kelasnya dan bila di total berjumlah 250 dataset. Kelas terdiri dari ikan Komet, ikan Molly, ikan Manfish, ikan Zebra, ikan Redfin.

2.2 Ikan Cupang

Secara umum, ikan cupang dibagi menjadi 3 jenis, yaitu cupang aduan, liar, dan hias. Namun karena beberapa jenis cupang aduan memiliki bentuk indah, maka tak jarang beberapa diantaranya juga dimanfaatkan sebagai ikan hias. Cupang berasal dari beberapa negara di Asia Tenggara, seperti Indonesia, Malaysia, Brunei Darussalam, Singapura, dan Vietnam.

Salah satu cupang asli Indonesia adalah *Betta channoides* yang ditemukan di Pampang, Kalimantan Timur. Jenis-jenis ikan cupang dikelompokkan menjadi 4 bagian yang setiap bagiannya masih memiliki jenis-jenisnya lagi, tak dapat dipungkiri bahwa sangat banyak dan sangat beragam jenis-jenis yang dimiliki ikan cupang dan tercatat kurang lebih sekitar 73 jenis di seluruh dunia dan Sebagian besar berasal dari Asia Tenggara. Berikut adalah 4 jenis ikan cupang secara garis besar :

1. Cupang Halfmoon

Jenis ikan cupang ini cukup digemari dan nilai jualnya lumayan tinggi. Ekornya berbentuk lebar dan berkibar menjadikannya sangat indah, sehingga tak heran jika harganya mahal. Satu ekor cupang halfmoon harganya dapat mencapai 500 ribu. Meski termasuk jenis ikan cupang aduan, jenis *halfmoon* yang mulai masuk ke Indonesia sekitar tahun 2010 ini lebih disarankan untuk

dijadikan ikan hias.

2. Cupang *Plakat*

Cupang ini berasal dari Thailand. Dalam bahasa Thailand, *plakat* artinya adalah “tarung”. Jenis cupang yang sangat agresif ini harganya meroket karena termasuk cupang langka. Warna tubuh ikan cupang plakat cenderung keabu-abuan dan sisiknya berkilau. Sirip dan ekornya dihiasi warna semburat merah yang indah. Saat sedang menakuti lawan, cupang plakat akan melebarkan sirip bagian depannya. Warnanya cupang ini tidak akan pudar meski dalam keadaan bertarung. Cupang *plakat* juga merupakan perenang yang handal.





3. Cupang *Crowntail*

Ikan ini juga disebut cupang serit dan diduga menjadi cupang pertama yang dibudidayakan di Indonesia. Jenis cupang ini memiliki sirip seperti jarum, serit atau sisir. Cupang ini berasal dari hasil persilangan cupang lokal dengan indukan impor Thailand berwarna biru dengan ekor delta dan bergerigi. Pengembang variasi cupang *crowntail* adalah Ahmad Yusuf, seorang peternak asal Jakarta Timur pada tahun 1997.

4. Cupang *Fancy*

Cupang *fancy* adalah ikan cupang mahal. Sesuai dengan namanya, *fancy* artinya adalah mewah, cocok dengan penampilannya yang terlihat elegan. Keanggunannya disebabkan oleh efek 3 dimensi pada warna tubuhnya. Saat berenang, warna tubuh cupang *fancy* akan terlihat seolah-olah seperti bergerak. Keunikan inilah yang tidak dimiliki cupang jenis lainnya. Warna cupang *fancy* memiliki kombinasi 5 hingga 7 warna. Umumnya ikan ini berwarna hijau, biru metalik, putih, merah, pink, dan peach. Bibit atau anakan cupang *fancy* dijual sekitar 200 hingga 300 ribu per ekor. Sementara untuk dewasa, terlebih yang masuk dalam kategori ikan kontes harganya dapat mencapai jutaan. Tabel 2.1 adalah empat gambar ikan cupang dan jenisnya

Tabel 2.1 Jenis-jenis ikan cupang

No.	Jenis	Gambar
1.	<i>Halfmoon</i>	
2.	<i>Plakat</i>	
3.	<i>Crowntail</i>	
4.	<i>Fancy</i>	

2.3 Klasifikasi

Klasifikasi adalah proses menentukan suatu obyek kedalam suatu kelas atau kategori yang telah ditentukan (Rahardjo, 2014). Komponen-komponen utama dari proses klasifikasi antara lain :

- a. Kelas, merupakan variabel tidak bebas yang merupakan label dari hasil klasifikasi.
- b. Predictor, merupakan variabel bebas suatu model berdasarkan dari karakteristik

- atribut data yang diklasifikasi.
- c. Set data pelatihan, merupakan sekumpulan data lengkap yang berisi kelas dan predictor untuk dilatih agar model dapat mengelompokkan ke dalam kelas yang tepat.
 - d. Set data uji, berisi data-data baru yang akan dikelompokkan oleh model guna mengetahui akurasi dari model yang telah dibuat.(Widodo dkk, 2014)
 - e. Klasifikasi citra merupakan salah satu teknik menginterpretasi citra digital. Klasifikasi citra didasarkan pada sampel masing-masing kelas, kualitasnya harus diperiksa dan dikuantifikasi setelahnya, dilakukan dengan pendekatan sampling dan hasil klasifikasi dengan kelas sesungguhnya dibandingkan. Perbandingan tersebut dilakukan dengan membuat error matrix yang dapat menghasilkan akurasi yang berbeda.

2.4 Citra Digital

2.4.1 Citra berwarna

Setiap piksel dari gambar berwarna mewakili warna yang merupakan kombinasi dari tiga warna primer (RGB = Merah Hijau Biru). Setiap warna primer menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna memiliki 255 warna. Artinya setiap piksel memiliki $28 \times 28 \times 28 = 224 = 16$ juta lebih kombinasi warna. Itulah mengapa bentuk ini disebut true color, karena memiliki banyak sekali warna sehingga bisa dikatakan menutupi hampir semua warna alam.

2.4.2 Citra *Grayscale*

Citra *grayscale* adalah jenis citra yang hanya memiliki satu *channel* warna. Hal ini menyebabkan nilai yang ditampilkan pada jenis citra ini hanyalah nilai intensitas atau lebih sering disebut *grayscale*. Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. Citra 2 bit mewakili 4 warna, citra 3 bit mewakili 8 warna, dan seterusnya. Semakin besar jumlah bit warna yang disediakan di memori, semakin halus gradasi warna yang terbentuk.

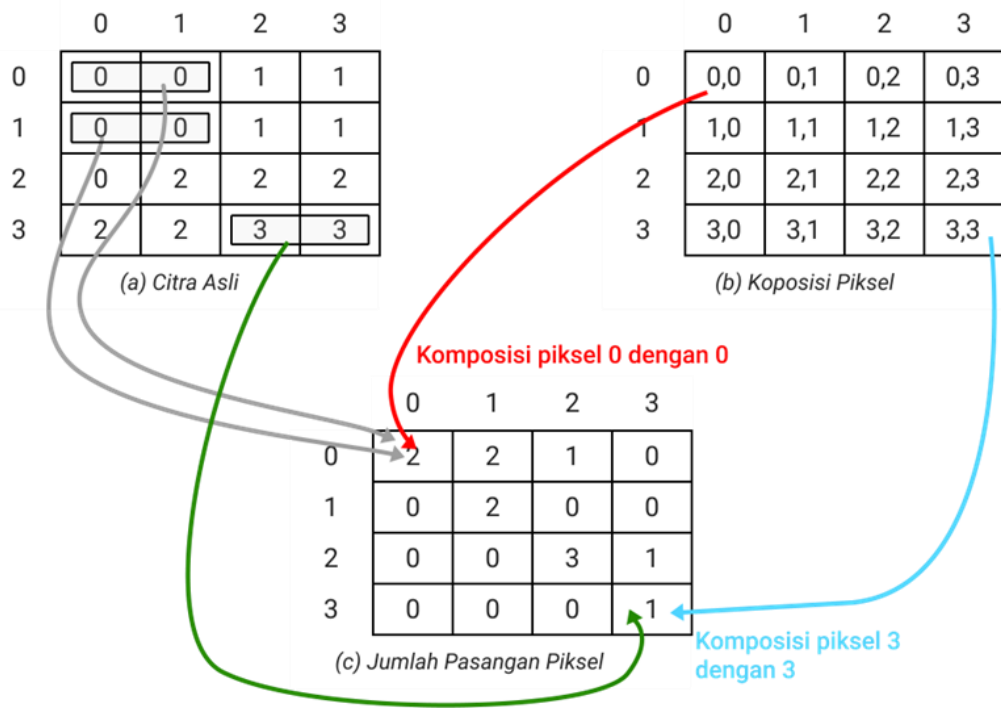
2.4.3 Citra biner

Citra ini hanya memiliki satu *channel* warna dan hanya memiliki dua kemungkinan nilai, yaitu 0 dan 1. Jenis citra ini adalah jenis citra yang sering digunakan untuk proses segmentasi citra.

2.5 GLCM (Gray Level Co-occurrence Matrix)

GLCM merupakan salah satu teknik tertua yang khusus dibuat untuk analisis tekstur (Muhathir et al., 2021). GLCM dibuat dengan menghitung nilai intensitas yang terjadi di satu garis abu yang ditemukan dalam gambar dengan hubungan spasial linier antara dua piksel. Hubungan spasial didefinisikan oleh pasangan parameter (θ, d) di mana θ adalah orientasi dan d adalah jarak antara dua piksel (Lloyd et al., 2017).

Koordinat pasangan piksel memiliki jarak d dan orientasi sudut θ . Jarak direpresentasikan dalam piksel dan sudut direpresentasikan dalam derajat. Kemudian dilakukan normalisasi terhadap matrik dengan menghitung probabilitas setiap elemen matrik (Rahardjo, 2014)



Gambar 2.1 Ilustrasi piksel matriks GLCM

Sumber gambar : www.eprints.utdi.ac.id

Pada matriks Gambar 2.1 adalah *matrix framework*, yang perlu diolah menjadi matrik yang simetris dengan cara ditambahkan dengan hasil transposnya, terdapat pada Gambar 2.2

$$\begin{array}{cccc|cccc}
 2 & 2 & 1 & 0 & 2 & 0 & 0 & 0 & 4 & 2 & 1 & 0 \\
 0 & 2 & 0 & 0 & 2 & 2 & 0 & 0 & 2 & 4 & 0 & 0 \\
 0 & 0 & 3 & 1 & 1 & 0 & 3 & 0 & 1 & 0 & 6 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 2
 \end{array} + = \begin{array}{cccc}
 4 & 2 & 1 & 0 \\
 2 & 4 & 0 & 0 \\
 1 & 0 & 6 & 0 \\
 0 & 0 & 1 & 2
 \end{array}$$

\longrightarrow
 Transpos GLCM sebelum dinormalisasi

Gambar 2.2 GLCM matriks sebelum normalisasi

Sumber gambar : www.eprints.utdi.ac.id

Kemudian untuk menghapus ketergantungan pada citra, nilai pada GLCM harus di normalisasikan seperti pada Gambar 2.3

$$\begin{array}{cccc}
 4 & 2 & 1 & 0 \\
 \hline
 24 & 24 & 24 & 24 \\
 2 & 4 & 0 & 0 \\
 \hline
 24 & 24 & 24 & 24 \\
 1 & 0 & 6 & 1 \\
 \hline
 24 & 24 & 24 & 24 \\
 0 & 0 & 1 & 2 \\
 \hline
 24 & 24 & 24 & 24
 \end{array}$$

Gambar 2.3 Normalisasi matriks GLCM

Sumber gambar : www.eprints.utdi.ac.id

GLCM menentukan frekuensi kombinasi dari nilai kecerahan piksel yang ditentukan. Artinya, mewakili pembentukan frekuensi dari pasangan piksel. Properti GLCM dari sebuah gambar dinyatakan sebagai matriks dengan jumlah baris dan kolom yang sama seperti nilai abu-abu pada gambar. Elemen dari matriks ini tergantung pada frekuensi dua piksel yang ditentukan. Kedua pasangan piksel dapat bervariasi tergantung pada area terdekat mereka. Setelah mendapatkan hasil normalisasi matriks akan dijadikan sebagai input pengukuran *probabilistic* yang merepresentasikan fitur tekstur, terdapat empat Parameter tekstur yang sering digunakan dalam GLCM, adalah sebagai berikut (Agustina & Ardiansyah, 2020):

1. Kontras

Merupakan perhitungan yang berhubungan dari jumlah keberagaman intensitas *Grayscale*.

$$\sum_{x,y} (x - y)^2 p(x, y)$$

(2.1)

2. Korelasi

Memberikan petunjuk dengan adanya struktur linier dalam citra dengan menunjukkan ukuran ketergantungan linier derajat keabuan.

$$\sum_{x,y} \frac{(x - \mu_x)(y - \mu_y)p(x,y)}{\sigma_x \sigma_y} \quad (2.2)$$

3. Homogen

Merupakan jumlah level keabuan sejenis dalam citra jika piksel semakin seragam maka *homogeneity* akan tinggi.

$$\sum_{x,y} \frac{p(x,y)}{1 + |x-y|} \quad (2.3)$$

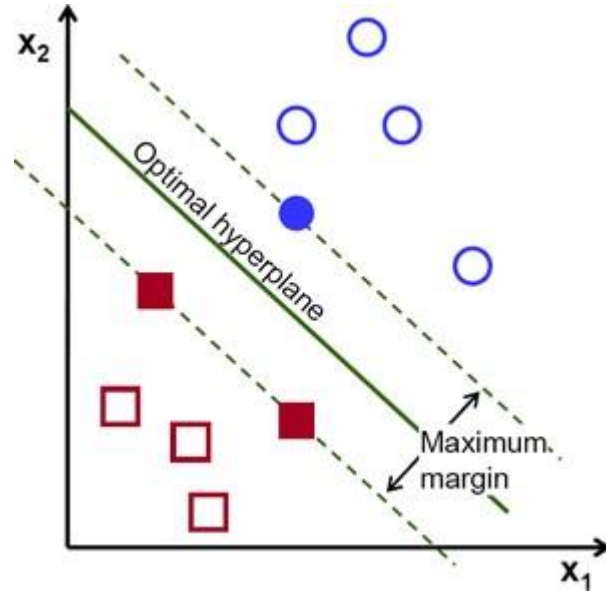
4. Energi

Fitur tekstur dari Energi dipresentasikan terhadap ukuran konsentrasi dari pasangan intensitas dalam matriks. Semakin tinggi energi semakin tinggi juga nilai kemiripan.

$$\sum_{x,y} p(x,y)^2 \quad (2.4)$$

2.6 SVM (Support Vector Machine)

2.6.1 Prinsip kerja SVM secara umum (Linearly separable)



Gambar 2.4 Ilustrasi SVM

Sumber gambar : www.trivusi.web.id

Linearly separable data merupakan data yang dapat dipisahkan secara linier. Misalkan $\{x_1, \dots, x_n\}$ adalah dataset dan $y_i \in \{-1, 1\}$ adalah label kelas dari data x_i . Pada Gambar 2.4 dapat dilihat berbagai alternatif bidang pemisah yang dapat memisahkan semua data set sesuai dengan kelasnya. Namun, bidang pemisah terbaik tidak hanya dapat memisahkan data tetapi juga memiliki margin paling besar. Adapun data yang berada pada bidang pembatas ini disebut *support vector*. Dalam contoh di atas, dua kelas dapat dipisahkan oleh sepasang bidang pembatas yang sejajar. Bidang pembatas pertama membatasi kelas pertama sedangkan bidang pembatas kedua membatasi kelas kedua, sehingga diperoleh:

$$\begin{aligned}x_i \cdot w + b &\geq +1 \text{ for } y_i = +1 \\x_i \cdot w + b &\leq -1 \text{ for } y_i = -1\end{aligned}\tag{2.5}$$

w adalah normal bidang dan b adalah posisi bidang relatif terhadap pusat koordinat. Nilai *margin* (jarak) antara bidang pembatas (berdasarkan rumus jarak garis ke titik pusat) adalah

Nilai margin ini dimaksimalkan dengan tetap memenuhi (2.1). Dengan

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

mengalikan b dan w dengan sebuah konstanta, akan dihasilkan nilai margin yang dikalikan dengan konstanta yang sama. Oleh karena itu, konstrain (2.1) merupakan scaling constraint yang dapat dipenuhi dengan *rescaling* b dan w . Selain itu, karena Memaksimalkan $1/(w)$ sama dengan meminimumkan $1/w^2$ dan jika kedua bidang pembatas pada (2.1) direpresentasikan dalam pertidaksamaan (2.2),

$$y_i(x_i \cdot w + b) - 1 \geq 0 \quad (2,6)$$

Maka pencarian bidang pemisah terbaik dengan nilai margin terbesar dapat dirumuskan menjadi masalah optimasi konstrain, yaitu

$$\begin{aligned} \min \frac{1}{2}|w|^2 \\ \text{s.t } y_i(x_i \cdot w + b) - 1 \geq 0 \end{aligned} \quad (2,7)$$

Persoalan ini akan lebih mudah diselesaikan jika diubah ke dalam formula lagrangian yang menggunakan *lagrange multiplier*. Dengan demikian permasalahan optimasi konstrain dapat diubah menjadi:

$$\min_{w,b} L_p(w,b,\alpha) \equiv \frac{1}{2}|w|^2 - \sum_{i=1}^n \alpha_i y_i(x_i \cdot w + b) + \sum_{i=1}^n \alpha_i \quad (2,8)$$

dengan tambahan konstrain, $\geq 0 \alpha_i$ (nilai dari koefisien *lagrange*). Dengan meminimumkan L_p terhadap w dan b , maka dari

$$\frac{\partial}{\partial b} L_p(w, b, \alpha) = 0$$

Maka diperoleh (2.5) dan (2,6)

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (2.9)$$

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad (2.10)$$

Vektor w sering kali bernilai besar (mungkin tak terhingga), tetapi nilai α_i terhingga. Untuk itu, *formula lagrangian* L_p (*primal problem*) diubah kedalam *dual problem* L_D . Dengan mensubsitusikan persamaan (2.6) ke LP diperoleh *dual problem* L_D dengan konstrain berbeda.

$$L_D(\alpha) \equiv \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (2.11)$$

$\min L \max L, \alpha =$. Jadi persoalan pencarian bidang pemisah terbaik dapat dirumuskan sebagai berikut:

$$\begin{aligned} \max_{\alpha} L_D &\equiv \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ \text{s.t.} \quad &\sum_{i=1}^n \alpha_i y_i = 0, \alpha_i \geq 0 \end{aligned} \quad (2.12)$$

Dengan demikian, dapat diperoleh nilai α_i yang nantinya digunakan untuk menemukan w . Terdapat nilai α_i untuk setiap data pelatihan. Data pelatihan yang memiliki nilai > 0 α_i adalah *support vector* sedangkan sisanya memiliki nilai $\alpha_i = 0$. Dengan demikian fungsi keputusan yang dihasilkan hanya dipengaruhi oleh *support vector*.

Formula pencarian bidang pemisah terbaik ini adalah pemasalahan *quadratic programming*, sehingga nilai maksimum *global* dari α_i selalu dapat ditemukan. Setelah solusi pemasalahan *quadratic programming* ditemukan (nilai α

i), maka kelas dari data pengujian x dapat ditentukan berdasarkan nilai dari fungsi keputusan:

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

x adalah *support vector*, ns = jumlah support vector dan d x adalah data yang akan diklasifikasikan. (Meliana N, 2022)

2.6.2 Prinsip kerja SVM Multiclass (*Non Separably*)

Untuk mengklasifikasikan data yang tidak dapat dipisahkan secara linier formula SVM harus dimodifikasi karena tidak akan ada solusi yang ditemukan. Oleh karena itu, kedua bidang pembatas (2.1) harus diubah sehingga lebih fleksibel (untuk kondisi tertentu) dengan penambahan variabel ξ_i ($\xi_i \geq 0, \forall_i : \xi_i = 0$. jika i x diklasifikasikan dengan benar) menjadi $x_i \cdot w + b \geq 1 - \xi_i$ untuk kelas 1 dan $x_i \cdot w + b \leq -1 + \xi_i$ untuk kelas 2. Pencarian bidang pemisah terbaik dengan dengan penambahan variabel ξ_i sering juga disebut *soft margin hyperplane*. Dengan demikian formula pencarian bidang pemisah terbaik berubah menjadi:

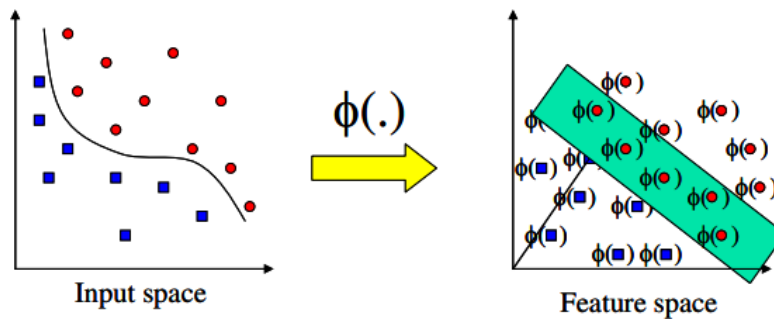
$$\begin{aligned} \min \frac{1}{2} |w|^2 + C \left(\sum_{i=1}^n \xi_i \right) \\ \text{s.t. } y_i (w \cdot x_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{aligned} \quad (2.14)$$

C adalah parameter yang menentukan besar penalti akibat kesalahan dalam klasifikasi data dan nilainya ditentukan oleh pengguna. Bentuk persoalan (2.10) memenuhi prinsip SRM, dimana meminimumkan $\frac{1}{2} |w|^2$ ekuivalen dengan meminimumkan dimensi $VC \left(\sum_{i=1}^n \xi_i \right)$ berarti meminimumkan error pada data pelatihan [OSU97]. Selanjutnya, bentuk *primal problem* sebelumnya berubah menjadi

$$\min_{w,b} L_p(w,b,\alpha) = \frac{1}{2} |w|^2 + C \left(\sum_{i=1}^n \xi_i \right) - \sum_{i=1}^n \alpha_i \{ y_i (x_i \cdot w + b) - 1 + \xi_i \} - \sum_{i=1}^n \mu_i \xi_i \quad (2.15)$$

Pengubahan L_p ke dalam dual problem, menghasilkan formula yang sama dengan persamaan (2.6) sehingga pencarian bidang pemisah terbaik dilakukan dengan cara yang hampir sama dengan kasus dimana data dapat dipisahkan secara linier, tetapi rentang nilai α_i adalah $0 \leq \alpha_i \leq C$. *Instance* yang memiliki nilai $\alpha_i = C$ disebut *bounded support vector*. Metode lain untuk mengklasifikasikan data yang tidak dapat dipisahkan secara linier adalah dengan mentransformasikan data ke dalam dimensi ruang fitur (*feature space*) sehingga dapat dipisahkan secara linier

pada *feature space*.



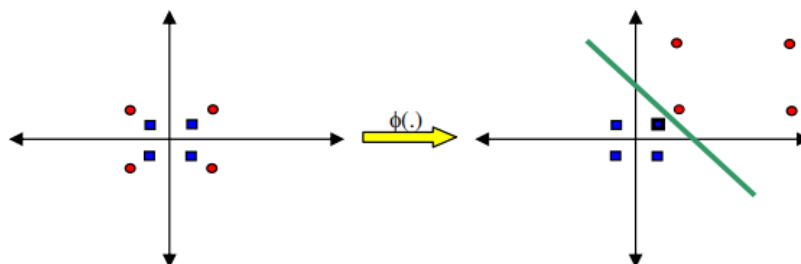
Gambar 2.5 Transformasi input ke ruang fitur

Sumber gambar : www.jurnal.unimus.ac.id

Caranya, data dipetakan dengan menggunakan fungsi pemetaan (transformasi) $(k \times k \rightarrow \phi \times k)$ ke dalam *feature space* sehingga terdapat bidang pemisah yang dapat memisahkan data sesuai dengan kelasnya Gambar 2.5. Misalkan terdapat data set yang datanya memiliki dua atribut dan dua kelas yaitu kelas positif dan negatif. Data yang memiliki kelas positif adalah $\{(2,2) (2, -2) (-2,2) (-2,2)\}$ dan data yang memiliki kelas negatif $\{(1,1)(1,-1) (-1,1) (-1,-1)\}$. Apabila data ini digambarkan dalam ruang dua dimensi Gambar 2.6 dapat dilihat data ini tidak dapat dipisahkan secara linier. Oleh karena itu, digunakan fungsi transformasi berikut:

$$\phi(x_1, x_2) = \begin{cases} \sqrt{x_1^2 + x_2^2} > 2 \rightarrow (4 - x_2 + |x_1 - x_2|, 4 - x_1 + |x_1 - x_2|) \\ \sqrt{x_1^2 + x_2^2} \leq 2 \rightarrow (x_1, x_2) \end{cases} \quad (2.16)$$

Data sesudah transformasi adalah $\{(6,2) (6,6) (2,6) (2,2)\}$ untuk kelas negatif, dan $\{(1,1) (1,-1) (-1,1) (-1,-1)\}$ untuk kelas positif. Selanjutnya pencarian bidang pemisah terbaik dilakukan pada data ini.



Gambar 2.6 Ilustrasi transformasi data non-linear

Sumber gambar : www.jurnal.unimus.ac.id

Dengan menggunakan fungsi transformasi $(\cdot) \mapsto \phi(\cdot)$, maka nilai

$$w = \sum_{i=1}^{ns} \alpha_i y_i \phi(x_i) \quad (2.17)$$

dan fungsi hasil pembelajaran yang dihasilkan adalah

$$f(x_d) = \sum_{i=1}^{ns} \alpha_i y_i \phi(x_i) \phi(x_d) + b \quad (2.18)$$

Feature space dalam prakteknya biasanya memiliki dimensi yang lebih tinggi dari vektor input (*input space*). Hal ini mengakibatkan komputasi pada *feature space* mungkin sangat besar, karena ada kemungkinan *feature space* dapat memiliki jumlah *feature* yang tidak terhingga. Selain itu, sulit mengetahui fungsi transformasi yang tepat. Untuk mengatasi masalah ini, pada SVM digunakan "kernel trick". Dari persamaan (2.11) dapat dilihat terdapat *dot product* $\phi(x_i) \phi(x_d)$. Jika terdapat sebuah fungsi kernel K sehingga $K(x_i, x_d) = \phi(x_i) \phi(x_d)$, maka fungsi transformasi $\phi(x_k)$ tidak perlu diketahui secara persis. Dengan demikian fungsi yang dihasilkan dari pelatihan adalah

$$f(x_d) = \sum_{i=1}^{ns} \alpha_i y_i K(x_i, x_d) + b \quad (x_i = \text{support vector}). \quad (2.19)$$

Syarat sebuah fungsi untuk menjadi fungsi kernel adalah memenuhi teorema *Mercer* yang menyatakan bahwa matriks kernel yang dihasilkan harus bersifat *positive semi-definite*. Fungsi kernel yang umum digunakan adalah sebagai berikut (Meliana N, 2022):

a. Kernel *Linear*

$$K(x_i, x) = x_i^T x \quad (2.20)$$

b. Kernel *Polynomial*

$$K(x_i, x) = (\gamma \cdot x_i^T x + r)^p, \gamma > 0 \quad (2.21)$$

c. Kernel *RBF*

$$K(x_i, x) = \exp(-\gamma |x_i - x|^2), \gamma > 0 \quad (2.22)$$

d. Kernel *Sigmoid*

$$K(x_i, x) = \tanh(\gamma x_i^T x + r) \quad (2.23)$$

Kemudian terdapat nilai C dan γ yang masing-masing dimiliki oleh masing kernel yang akan ditunjukkan nilainya dengan Tabel 2.2

Tabel 2.2 Parameter hyperparameter SVM

Kernel	Nilai C (Regularisation)	Gamma
Linear	[0.001, 0.01, 0.1, 1, 10,100,1000]	-
RBF	[0.001, 0.01, 0.1, 1, 10,100,1000]	[1, 0.1, 0.01, 0.001, 0.0001]
Polynomial	[0.001, 0.01, 0.1, 1, 10,100,1000]	[1, 0.1, 0.01, 0.001, 0.0001]
Sigmoid	[0.001, 0.01, 0.1, 1, 10,100,1000]	[1, 0.1, 0.01, 0.001, 0.0001]

2.6.3 Kelebihan dan kekurangan SVM

SVM ditemukan oleh Vladimir N. Vapnik dan Aleksei Ya. Chervonenkis pada tahun 1963 (Kristian, 2018). Sejak itu, SVM telah digunakan untuk mengklasifikasikan teks, *hypertext*, dan gambar. SVM dapat bekerja dengan karakter tulisan tangan, dan algoritma ini sudah digunakan di laboratorium biologi untuk melakukan tugas seperti penyortiran protein. Metode ini tentunya juga memiliki kelebihan dan kekurangan seperti halnya metode-metode yang lainnya. Berikut adalah kelebihan metode SVM ini

a. Kelebihan

- Cocok untuk ruang dimensi tinggi.
- Sangat efektif digunakan apabila dimensi lebih tinggi daripada jumlah sampel.
- Bekerja dengan maksimal ketika terdapat margin yang jelas antara kelas atau class.
- Hemat memori

b. Kekurangan

- Tidak cocok untuk dataset yang berukuran besar
- Kurang maksimal bila kualitas dataset memiliki banyak noise karena dapat menyebabkan kelasnya tumpang tindih

- Beban komputasi yang tinggi karena SVM bekerja meletakkan beberapa titik diatas maupun dibawah hyperplane sehingga tidak ada kejelasan mengenai probabilitik yang terjadi.

2.7 Google Colab

Google Colab atau kependekan dari *Google Colaboratory* adalah sebuah IDE berbasis *Python* yang dikeluarkan oleh *Google Inc.* IDE ini merupakan IDE yang memungkinkan penggunaanya untuk memakai cloud secara gratis, tidak hanya itu aplikasi ini juga memiliki fungsi *built-in library* yang artinya semua *library* sudah tersedia dan dapat dipanggil.

Aplikasi ini diciptakan untuk keperluan *data scientific* dan apapun yang berkaitan dengan *machine learning* ataupun *deep learning*. *Google Colab* umumnya digunakan untuk mendukung kebutuhan *developer* dan praktisi informasi akan informasi yang sangat teknis. Selain itu, *Google Colab* bisa di-*share*, sehingga sangat mendukung kebutuhan kolaborasi antar anggota tim.

Menggunakan *Google Colab* memberikan banyak kemudahan dan keunggulan bagi para user. Keunggulan *Google Colab* antara lain lebih fleksibel, GPU yang gratis, hingga bisa melakukan kolaborasi. Berikut di antaranya yang bisa dirasakan oleh pengguna.

2.8 Machine Learning

Machine Learning adalah salah satu cabang dari *Artificial Intelligence* atau kecerdasan buatan. *Machine Learning* ini adalah mesin yang mampu belajar dengan sendirinya dan dikembangkan agar dapat belajar tanpa bantuan ataupun arahan dari user atau manusianya. *Machine Learning* pertama kali dikemukakan oleh beberapa ilmuwan matematika seperti Adrien Marie Legendre, Thomas Bayes dan Andrey Markov sekitar tahun 1920. *Machine learning* ini juga merambah ke dunia matematika, statistika, data mining, dan lainnya. Teknik-teknik *machine learning* pun fleksibel artinya dapat digunakan sesuai dengan kebutuhan dalam bidang masing-masing.

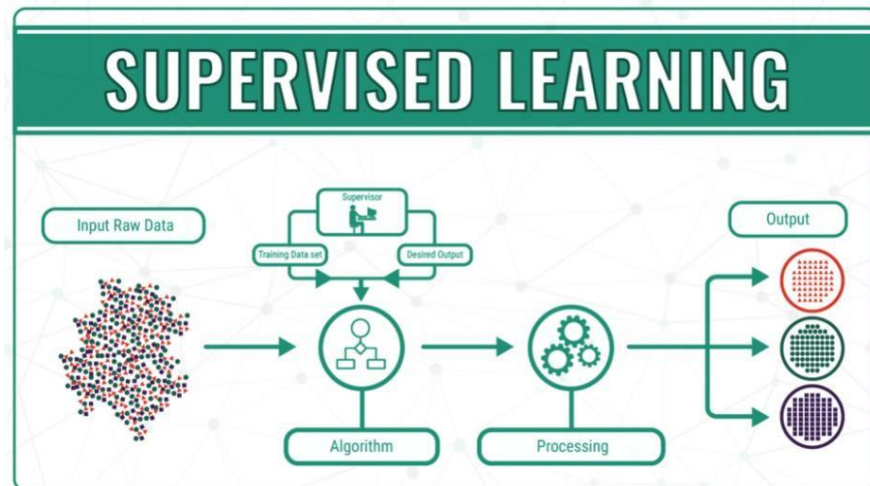
Machine Learning bekerja dengan belajar dari data untuk meningkatkan data, mendeskripsikan data dan juga memprediksi hasil dengan perintahnya sendiri.

Hal ini juga mengakibatkan persentase keberhasilan atau keefektifan dari *Machine Learning* ini sangat tinggi karena model yang ada dapat belajar sendiri dan dapat dikatakan *Machine Learning* ini adalah model yang efektif. *Machine Learning* sendiri dibagi menjadi tiga bagian yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*.

2.8.1 Supervised Learning

Supervised Learning atau pembelajaran secara terarah merupakan pembelajaran yang dilatih untuk pola input data dan label outputnya. *Supervised* biasanya dimulai dengan pengumpulan data dan merupakan pemahaman tentang bagaimana data diklasifikasikan. *Supervised Learning* adalah metode pembelajaran yang hasilnya harus diketahui oleh sistem. Artinya, cara ini bekerja dengan cara menggunakan kembali data dan hasil cetak yang telah dimasukkan ke dalam sistem atau dibuat sebelumnya. Dalam hal ini, pola input dan output perlu untuk mengenali sebuah informasi dalam memori pada sistem. Sistem akan menerima data dan akan menampilkan pola output dan mencocokkan polanya dengan pola input. Jika hasil cocok, maka akan ditampilkan dari memori pada sistem dalam bentuk output. Dan jika hasilnya tidak cocok, maka output akan *error*.

Supervised Learning biasa digunakan pada prosedur yang lebih kompleks dan rumit. Penjelasan lebih singkatnya yaitu *Supervised Learning* ini merupakan pembelajaran yang ada *supervisornya*, dimana *supervisor* ini merupakan label atau *tag* dari tiap datanya. Contohnya yaitu gambar monyet di *tag* “monyet” di setiap gambar masing-masing monyet, kemudian gambar babon di *tag* “babon” di setiap masing-masing gambar babon. *Machine Learning* kategori dapat berupa klasifikasi (“monyet”, “babon”, “beruk”, dll.) dan regresi (berat badan, tinggi badan, yang dihasilkan, dan lainnya). Data yang telah dikategorikan mudah untuk dipahami pengguna atau user yang melatih data simulasi agar sesuai dengan *tag* yang sudah tersedia. Untuk Langkah-langkah dan algoritmanya dapat dilihat pada Gambar 2.7



Gambar 2.7 Supervised Learning

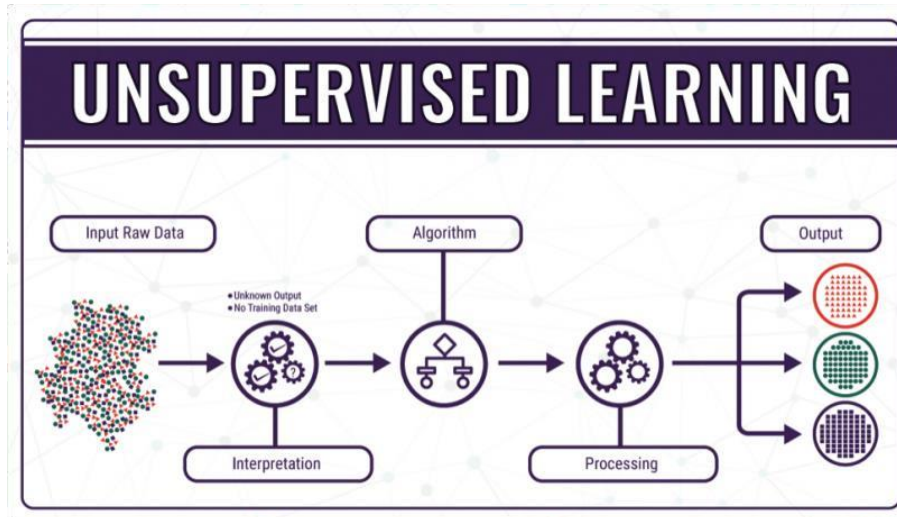
Sumber gambar : www.techfor.id

2.8.2 Unsupervised Learning

Unsupervised learning adalah metode pembelajaran mesin yang tidak dipandu sedemikian rupa sehingga tidak ada yang dapat mengetahui hasil yang diharapkan atau dalam artian hasil yang ditampilkan bergantung pada bobot yang disepakati pada awal pembuatan sistem dan dalam kerangka tertentu. Algoritma *Unsupervised Learning* bersifat deskriptif, yang akan berguna untuk mengelompokkan atau mengkategorikan data. Algoritma ini tidak mendapatkan training dataset, karena algoritma ini bukan bersifat prediktif, sehingga membutuhkan pembelajaran dari data yang telah ada. Metode ini memiliki keunggulan dibandingkan dengan *Supervised Learning*.

Supervised Learning memiliki label dasar prediksi baik dan membuat klasifikasi serta regresi yang memungkinkan. *Unsupervised Learning* tidak menggunakan label dalam memprediksi fitur data atau variabel. Jika melibatkan data berjumlah besar, maka *Unsupervised Learning* adalah yang terbaik contoh platform yang tidak menggunakan tag yaitu *Instagram, Snapchat dan Twitter*. Untuk memahami makna dari data, dibutuhkan algoritma yang dapat memahami maksud dari data tersebut untuk proses klasifikasi berdasarkan pola data. Cara kerja *Unsupervised Learning* yaitu dengan mencari pola tersembunyi dari dataset yang ada. Algoritma ini bekerja dengan menganalisis data yang tidak berlabel untuk menemukan pola yang tersembunyi dan menentukan hubungan atau korelasinya.

Metode ini juga tidak menggunakan data training untuk melakukan prediksi. Untuk Langkah-langkah dan algoritmanya dapat dilihat pada Gambar 2.8.



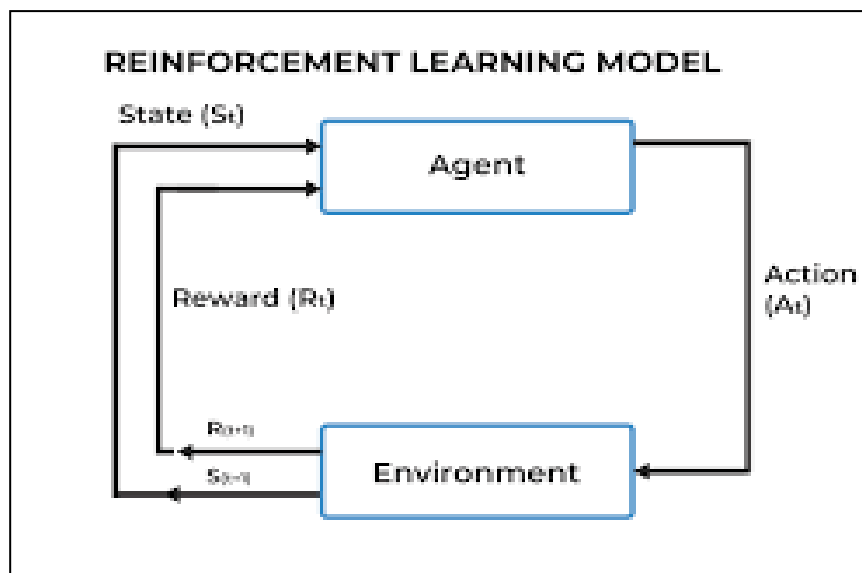
Gambar 2.8 Unsupervised Learning

Sumber gambar : www.techfor.id

2.8.3 Reinforcement Learning

Reinforcement Learning merupakan algoritma yang dapat belajar sendiri dari lingkungan atau (environment) dengan melakukan pencaharian sendiri untuk menentukan perilaku yang ideal sehingga dapat membuat kinerja algoritmanya menjadi maksimal dengan cara berinteraksi dengan environment yang ada. Algoritma *Reinforcement Learning* ini memiliki karakteristik, seperti tidak adanya *supervisor* karena ada bilangan real saja, pengambilan keputusan secara berurutan, *feedback* yang diterima selalu tertunda, waktu sangat berperan dalam masalah di *reinforcement* dan agen menentukan aksi selanjutnya yang diterima. Algoritma ini digunakan jika ingin menemukan situasi yang membutuhkan tindakan dan memungkinkan untuk mengetahui metode terbaik untuk mendapatkan *reward* besar. Cara kerja *Reinforcement Learning* ini yaitu selama proses training, komputer dituntun oleh algoritma dalam melakukan *trial* dan *error* dan akan ada *feedback* yang didapatkan komputer itu. *Feedback* yang di dapat dari aksi sebelumlah akan digunakan sebagai panduan atau (*guide*) untuk melakukan aksi setelahnya.

Algoritma ini memiliki dua tipe, yakni tipe positif dan negatif. *Reinforcement* Positif diartikan sebagai aksi yang akan terjadi karena perilaku dengan meningkatkan kekuatan dan frekuensi yang akan berdampak positif pada tindakan yang diambil. Namun, terlalu banyak *reinforcement* juga dapat mengakibatkan pengoptimalan *state* yang berlebihan. Sedangkan *reinforcement learning* negatif adalah penguatan perilaku yang terjadi karena adanya kondisi yang harusnya dihindari atau dihentikan. Tipe algoritma ini dapat bekerja dan membantu untuk menentukan standar kerja minimum. Pada Gambar 2.9 dijelaskan ilustrasi *Reinforcement Learning*.



Gambar 2.9 Reinforcement Learning
 Sumber gambar : www.spiceworks.com

2.9 Evaluasi performa

2.9.1 Confusion Matrix

Confusion Matrix adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan kombinasi berbeda dari nilai prediksi dan nilai aktual. Terdapat 4 istilah didalam *Confusion Matrix* yaitu *True Positive*, *True Negative*, *False Positive*, *False Negative*. Contoh gambar confusion matriks dapat dilihat pada Gambar 2.10

	Prediksi	
Aktual	TRUE	FALSE
TRUE	TP	FP
FALSE	FN	TN

Gambar 2.10 Tabel Confusion Matrix

Sumber gambar : www.pengalaman-edukasi.com

2.9.2 Accuracy

Setelah nilai *Confusion Matrix* didapatkan maka dapat dihitung nilai *accuracy*. *Accuracy* menggambarkan seberapa akurat model dalam mengklasifikasikan dengan benar. Rumus dari perhitungan akurasi dapat dilihat pada rumus dibawah.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.24)$$

2.9.3 Precision

Kemudian dihitung nilai *precision* yang menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Rumus *precision* dapat dilihat pada rumus dibawah.

$$Precision = \frac{TP}{TP + FP} \quad (2.25)$$

2.9.4 Recall

Nilai *recall* menggambarkan keberhasilan suatu model saat menemukan kembali sebuah informasi. Rumus *recall* dapat dilihat pada rumus dibawah ini.

$$Recall = \frac{TP}{TP + FN} \quad (2.26)$$

2.9.5 F-1 Score

F-1 Score menggambarkan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. *Accuracy* tepat kita gunakan sebagai acuan performansi algoritma jika dataset kita memiliki jumlah data *False Negatif* dan *False Positif* yang sangat mendekati atau simetris. Namun jika jumlahnya tidak mendekati, maka kita harus

menggunakan *F1 Score* sebagai acuan. Rumus *F-1 Score* dapat dilihat pada rumus dibawah ini.

(2.27)

$$\frac{1}{F1} = \frac{1}{2} \left(\frac{1}{Precision} + \frac{1}{recall} \right)$$