

BAB II

DASAR TEORI

2.1 TINJAUAN PUSTAKA

Penelitian oleh T. Saputri yang berjudul “Klasifikasi Citra Tanaman Hidroponik Pakcoy Menggunakan Ekstraksi Fitur *Histogram* dan Metode *K-Nearest Neighbor*”. Penelitian tersebut bertujuan untuk mengidentifikasi tanaman hidroponik pakcoy yang telah “siap” untuk dipanen dan yang “belum siap” untuk dipanen. Dataset diambil secara manual berupa citra tanaman pakcoy dengan total jumlah citra sebanyak 200 citra. Analisis klasifikasi menggunakan metode ekstraksi fitur *Histogram* dan klasifikasi K-NN dengan perhitungan jarak *Chi-Square* yang memberikan hasil klasifikasi dengan akurasi tertinggi mencapai 100% [6].

Penelitian sebelumnya oleh M. Setiawan, Y. Sari, P. Adikara yang berjudul judul “Klasifikasi Citra Makanan Menggunakan *K-Nearest Neighbor* dengan Fitur Bentuk *Simple Morphological Shape Descriptors* dan Fitur Warna *Greyscale Histogram*”. Penelitian tersebut bertujuan untuk mengidentifikasi makanan berdasarkan bentuk dan warna. Berdasarkan hasil pengujian, metode yang digunakan adalah metode *Grayscale Histogram* yang menghasilkan nilai akurasi sebesar 60%. Sedangkan untuk metode SMSD menghasilkan nilai akurasi sebesar 54,8%. Dan jika menggunakan gabungan metode *Grayscale Histogram* dan metode SMSD menghasilkan nilai akurasi sebesar 77,8% [7].

Penelitian sebelumnya Oleh M. Akbar dengan judul “Klasifikasi Biji Kopi Sangrai Menggunakan Pengolahan Citra dengan Metode LBP dan Klasifikasi KNN”. Penelitian tersebut bertujuan untuk mengidentifikasi kualitas biji kopi dengan menggunakan *image processing* dengan menganalisis citra berdasarkan warna biji kopi setelah proses *roasting*. Metode yang digunakan adalah *Histogram* dengan klasifikasi *K-Nearest Neighbor*. Data latih yang digunakan pada penelitian inialah sebanyak 120 *dataset* yang perkelasnya terdiri atas 40 *dataset*, dan untuk data uji sebanyak 90 *dataset* yang perkelasnya terdiri atas 30 *dataset*. Hasil tertinggi dari performansi sistem yang diperoleh dari dipenelitian ini ialah 91,11% [8].

Penelitian sebelumnya oleh A. B . Prawira dengan judul “Sistem Klasifikasi Panen Tanaman Bawang Merah Berbasis Fitur *Histogram*”. Penelitian tersebut bertujuan untuk mengidentifikasi tingkat kematangan tanaman bawang merah

dengan menggunakan metode klasifikasi *K-Nearest Neighbor*. Penelitian ini menggunakan 600 citra tanaman bawang merah yang terbagi atas 300 citra tanaman bawang merah yang sudah matang dan 300 citra tanaman bawang merah belum matang. Hasil dari penelitian ini menunjukkan bahwa metode yang digunakan dapat dinyatakan mampu melakukan klasifikasi citra dengan akurasi 96,5%, dengan nilai presisi sebesar 94,5%, dan hasil *recall* bernilai 94,9%. Dapat disimpulkan bahwa hasil penelitian ini menunjukkan bahwa model sistem yang dirancang dapat bekerja dengan cukup baik untuk menentukan tingkat kesiapan panen tanaman Bawang merah [9].

Penelitian sebelumnya oleh S. Adha, Y. Sari, dan R. Wihandika dengan judul “Klasifikasi Jenis Citra Makanan Tunggal Berdasarkan Fitur *Histogram* dan *Hue Saturation Value* Menggunakan *Improved K-Nearest Neighbor*”. Penelitian tersebut bertujuan untuk menganalisis jenis makanan yang berkualitas buruk dan dapat berpotensi menyebabkan suatu penyakit maupun alergi. Pada penelitian ini, digunakan fitur warna dan tekstur mengklasifikasi jenis citra makanan yang ingin diteliti. Ekstraksi fitur warna dilakukan dengan menggunakan ruang warna *Hue Saturation Value* (HSV) sedangkan untuk fitur tekstur menggunakan metode *Histogram*. Untuk klasifikasinya menggunakan metode *Improved K-Nearest Neighbor* (*Improved K-NN*). Hasil pengujian dengan nilai $k=1$ menunjukkan bahwa nilai akurasi tertinggi pada penelitian ini sebesar 90,47%. Saat yang digunakan hanya fitur warna saja, didapatkan nilai akurasi tertingginya sebesar 90,47%. Kemudian, saat yang digunakan hanya fitur tekstur saja, nilai akurasi yang didapatkan sebesar 85,71% [10].

Penelitian sebelumnya menjadi landasan untuk kelengkapan data sekaligus mempertajam masalah yang dikaji. Berdasarkan tinjauan pustaka dari para peneliti sebelumnya, menunjukkan bahwa metode ekstraksi fitur *Histogram* dan klasifikasi dengan *K-Nearest Neighbor* (K-NN) dapat diaplikasikan dalam pembuatan model klasifikasi *Supervised Learning* (*Dataset* berlabel). Gabungan dari kedua metode tersebut memungkinkan untuk mencapai tingkat akurasi yang baik dalam mengklasifikasi citra adonan donat. Tabel 2.1 merupakan ringkasan dari penelitian yang terkait.

Tabel 2. 1 Tinjauan Pustaka

No	Judul	Metode	Hasil	Kelemahan
1.	Klasifikasi Citra Tanaman Hidroponik Pakcoy Menggunakan Ekstraksi Fitur <i>Histogram</i> dan Metode <i>K-Nearest Neighbor</i> . T. Saputri	<i>Histogram</i> dan Metode <i>K-Nearest Neighbor</i>	Hasil klasifikasi dengan akurasi tertinggi mencapai 100%	Kurang optimal parameter jumlah cek poin dan jarak yang digunakan pada metode LBP.
2.	Klasifikasi Citra Makanan Menggunakan <i>K-Nearest Neighbor</i> dengan Fitur Bentuk <i>Simple Morphological Shape Descriptors</i> dan Fitur Warna <i>Greyscale Histogram</i> . M. Setiawan, Y. Sari, P.Adikara	<i>K-Nearest Neighbor</i> dengan Fitur Bentuk <i>Simple Morphological Shape Descriptors</i> dan Fitur Warna <i>Greyscale Histogram</i>	Hasil pengujian jika hanya menggunakan metode <i>Grayscale Histogram</i> menghasilkan nilai akurasi sebesar 60%. Jika hanya menggunakan metode SMSD menghasilkan nilai akurasi sebesar 54,8%. Jika menggunakan metode <i>Grayscale Histogram</i> dan metode SMSD menghasilkan nilai akurasi sebesar 77,8%	Kurang data <i>training</i> yang digunakan, serta diperlukan penambahan ekstraksi fitur.

3.	Klasifikasi Biji Kopi Sangrai Menggunakan Pengolahan Citra dengan Metode LBP dan Klasifikasi KNN. M. Akbar	<i>Histogram</i> dan Metode <i>K-Nearest Neighbor</i>	Hasil performansi sistem tertinggi yang didapat oleh penulis dipenelitian ini adalah 91,11%	Kurangnya <i>dataset</i> , disarankan menggunakan metode deep learning untuk hasil yang lebih akurat.
4.	Sistem Klasifikasi Panen Tanaman Bawang Merah Berbasis Fitur Histogram. A. B. Prawira	Metode <i>K-Nearest Neighbor</i> .	Hasil citra mata dengan akurasi 100% pada $K = 3, 5, 7$, dan 9. Sedangkan klasifikasi citra kulit mendapatkan akurasi sekitar 70% pada $K = 3$ dan 9, 63% pada $K = 7$ serta 60% pada $K = 9$	Data uji yang digunakan jumlahnya ditambah serta berbeda dengan data latih untuk meyakinkan akurasi sistem.
5.	Klasifikasi Jenis Citra Makanan Tunggal Berdasarkan Fitur <i>Histogram</i>	<i>Histogram</i> dan <i>Hue Saturation Value</i> Menggunakan	Hasil pengujian nilai k menunjukkan bahwa didapatkan akurasi tertinggi sebesar 90,476% dengan nilai $k=1$.	Perlu dilakukan modifikasi nilai radius dan point

	<p>dan <i>Hue Saturation Value</i> Menggunakan <i>Improved K-Nearest Neighbor</i></p>	<p>metode <i>Improved K-Nearest Neighbor</i></p>	<p>Ketika fitur yang digunakan hanya fitur warna, didapatkan nilai akurasi tertinggi 90,476% dengan nilai k=1. Ketika fitur yang digunakan hanya fitur tekstur, didapatkan nilai akurasi tertinggi 85,714% dengan nilai k=1. Hasil pengujian metode klasifikasi menunjukkan bahwa metode <i>Improved K-NN</i> menghasilkan akurasi yang lebih tinggi dari metode K-NN dengan rata-rata akurasi 80,306%. Sehingga hasil klasifikasi terbaik didapatkan dengan penggunaan kombinasi fitur warna dan tekstur dengan metode klasifikasi <i>Improved K-NN</i></p>	<p>atau jumlah ketetangaan pada metode ekstraksi fitur LBP agar hasil akurasi yang dihasilkan dari fitur tekstur lebih tinggi. Selain itu perlu dilakukan penambahan data latih pada tiap jenis variasi citra agar bobot pada data latih yang didapatkan lebih tinggi</p>
--	---	--	--	---

2.2 DASAR TEORI

2.2.1 Donat

Donat adalah salah satu kue populer dan menjadi favorit di seluruh dunia. donat merupakan salah satu produk *bakery* yang terbuat dari campuran tepung terigu, kuning telur, gula, air, mentega, dan ragi. Donat yang paling umum adalah donat yang berbentuk bulat dengan lubang di tengahnya, tetapi ada juga varian bulat tanpa lubang yang disebut bomboloni, yang kemudian diisi dengan berbagai selai atau *filling* seperti pada gambar 2.1. Untuk menciptakan donat berkualitas tinggi, diperlukan percobaan berulang (*experiment*) baik dalam metode pembuatan maupun formulasi bahan untuk mendapatkan hasil yang diinginkan. Bahan yang digunakan dalam pembuatan donat memiliki fungsi yang spesifik sehingga harus diambil dari bahan-bahan yang sesuai dengan kriteria yang diperlukan, adapun fungsi dari tiap bahan dapat dilihat pada tabel 2.2 [11].

Tabel 2. 2 Bahan dasar pembuatan donat

Bahan Dasar Pembuatan Donat	Fungsi
Tepung terigu	Berfungsi untuk menambah nilai gizi, kelembatan, membangun struktur adonan, menambah volume, dan menambah warna.
Ragi	Berfungsi untuk membuat adonan mengembang; serta berkontribusi dalam pembentukan aroma dan rasa selama proses fermentasi.
Air	Air merupakan medium fermentasi yang berfungsi untuk pertumbuhan ragi.
Gula	Berfungsi sebagai sumber nutrisi ragi sehingga mempercepat ragi bekerja; menambah rasa dan warna pada donat.
Margarin/lemak	Berfungsi untuk menambah rasa, aroma, keempukan serta memperlambat basi.
Telur	Berfungsi untuk menambah nilai gizi, kelembatan, membangun struktur adonan, menambah volume, dan menambah warna.



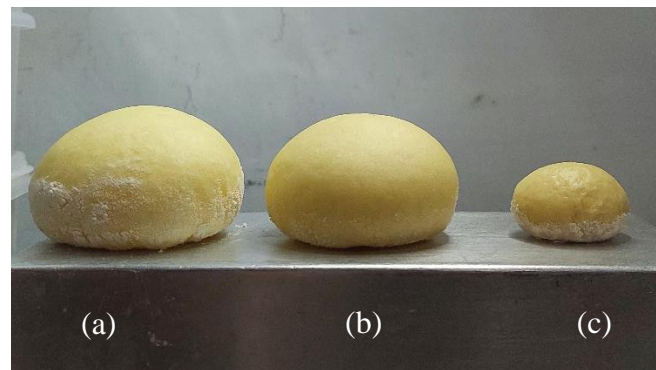
Gambar 2. 1 (a) Donat Bomboloni, (b) Donat Cincin

Proses pembuatan donat meliputi beberapa tahapan utama, berikut adalah penjelasan singkat mengenai setiap tahap:

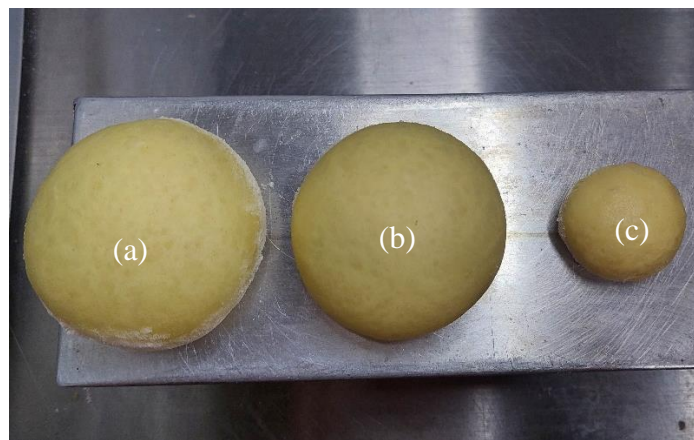
1. *Mixing* (Pencampuran): Bahan-bahan seperti tepung terigu, kuning telur, gula, air, mentega, dan ragi dicampur hingga membentuk adonan yang elastis.
2. *Fermentation* (Fermentasi): Adonan yang telah terbentuk dibiarkan untuk mengalami fermentasi. Proses ini melibatkan penguraian karbohidrat dalam tepung oleh ragi, menghasilkan gas karbon dioksida (CO₂) dan alkohol, yang menyebabkan adonan mengembang dan meningkatkan volume.
3. *Make up* (Pembentukan): Adonan dipotong, ditimbang, dan dibentuk sesuai dengan bentuk donat yang diinginkan.
4. *Proofing* (Pengembangan): Adonan yang telah dibentuk ditempatkan di loyang dan diberi waktu untuk berkembang kembali hingga mencapai dua kali ukuran semula.
5. *Frying* (Penggorengan): Adonan yang telah mencapai volume maksimal digoreng dalam minyak yang banyak dengan api sedang (80°C).
6. *Finishing* (Penyelesaian): Donat yang telah digoreng dapat diberi berbagai macam *topping*, hiasan di atas permukaan donat, atau diisi dengan *filling* sesuai selera [12].

Adonan donat yang memiliki kualitas yang baik dan kualitas yang tidak baik secara gampang dapat ditandai secara visual atau kasat mata dengan parameter tekstur dari adonan donat serta volume dari adonan donat. Adonan donat yang tidak memenuhi standar dapat dinilai dari tekstur yang tidak merata dan terdapat kerutan pada badan adonan serta terdapat gelembung udara yang besar pada permukaan adonan yang membuat adonan terlihat lebih transparan, dan juga volume pada

adonan donat terlihat *over* dari adonan asli seperti pada Gambar 2.2 (a) dan Gambar 2.3 (a). Adonan donat yang memiliki kualitas baik memiliki tekstur yang halus serta gelembung udara yang merata, dan juga volume pada adonan donat terlihat proporsional yaitu tidak terlalu kecil dan tidak terlalu besar seperti pada gambar 2.2 (b) dan Gambar 2.3 (b). Adonan donat yang belum memenuhi standar dapat dinilai dari tekstur yang masih terlihat kasar dan berminyak, belum memiliki gelembung udara, dan volume yang masih sangat kecil seperti pada Gambar 2.2 (c) dan Gambar 2.3 (c).



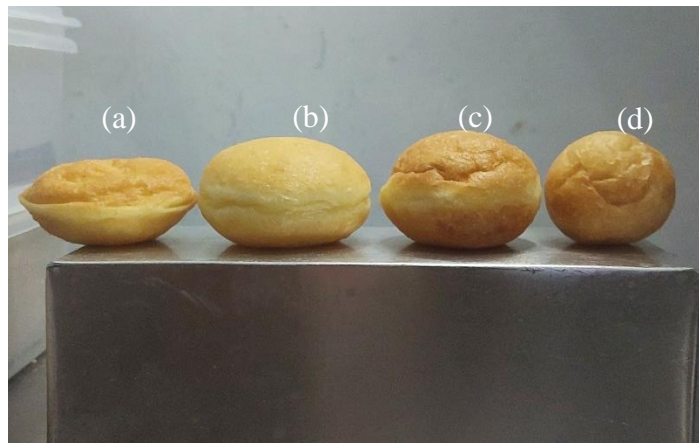
Gambar 2. 2 Perbandingan Adonan Donat Tampak Samping



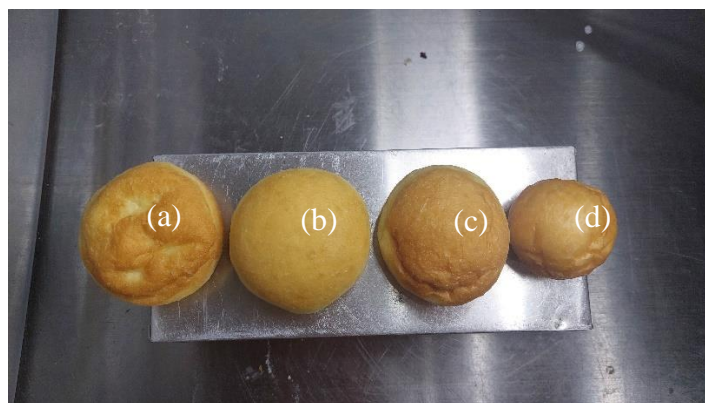
Gambar 2. 3 Perbandingan Adonan Donat Tampak Atas

Pada Gambar 2.4 (Produk donat dari samping), Gambar 2.5 (Produk donat dari atas), dan Gambar 2.6 (Bagian dalam produk donat). Dapat dilihat perbandingan tekstur dari adonan donat yang telah digoreng dimana pada gambar (a) yaitu adonan yang *over proofing* atau adonan yang proses *proofing*nya melebihi waktu yang seharusnya, menghasilkan produk donat yang kempis, dengan tekstur mengkerut pada bagian atas atau bagian permukaan produk, serta serat dalam produk yang tidak merata, terdapat bagian yang terlalu jarang dan terdapat juga bagian serat yang

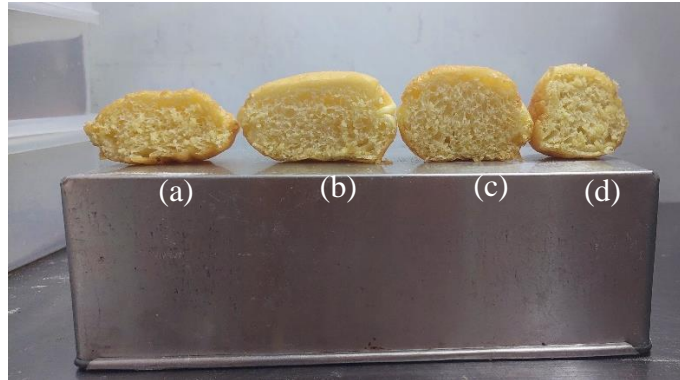
terlalu rapat dan kasar. Pada gambar (b) yaitu adonan yang mengembang sempurna, menghasilkan produk yang baik pada bagian atas atau bagian samping terlihat halus dan volume yang proporsional, serta serat dalam produk yang tidak terlalu jarang dan tidak terlalu rapat. Pada gambar (c) yaitu adonan donat setengah mengembang, menghasilkan produk yang kurang sempurna dimana terdapat kerutan pada produk serta serat produk yang terlalu rapat sehingga saat dimakan produk akan terasa kurang lembut dan agak keras. Pada gambar (d) yaitu adonan donat yang belum mengembang dan baru saja melalui proses make up, menghasilkan produk yang sangat kacil dengan permukaan tidak rata dan pecah-pecah, serta serat adonan yang sangat rapat sehingga menghasilkan produk yang keras.



Gambar 2. 4 Produk donat dari samping



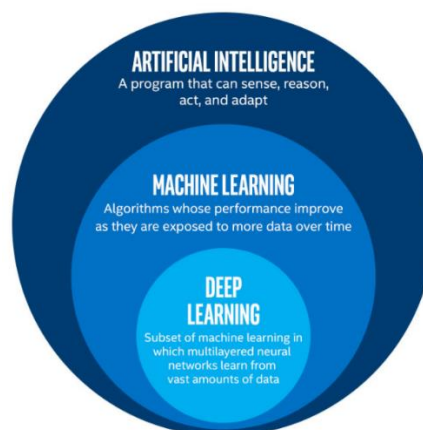
Gambar 2. 5 Produk donat dari atas



Gambar 2. 6 Bagian dalam produk donat

2.2.2 *Machine Learning*

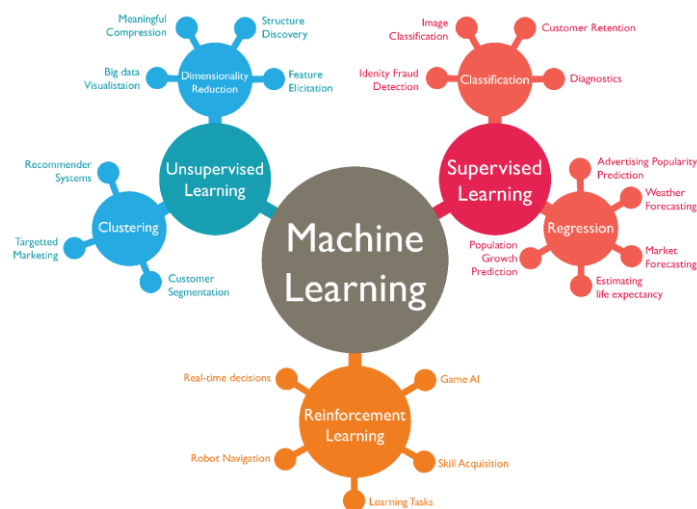
Machine learning adalah bagian dari *Artificial Intelligence* (AI) atau kecerdasan buatan yang berfokus pada desain dan analisis algoritma sehingga komputer memiliki kemampuan mempelajari suatu data atau informasi yang diberikan secara mandiri, algoritma dalam *machine learning* menggunakan teknik-teknik statistik untuk menemukan suatu pola. Dengan tujuan untuk mengubah data menjadi suatu tindakan yang nyata, sehingga komputer dapat berperilaku seperti manusia, yaitu dapat meningkatkan pemahamannya melalui pengalaman atau dari data yang ada untuk memperbaiki cara kerjanya bila dihadapkan pada hal serupa di masa depan, sehingga komputer dapat membuat keputusan secara mandiri tanpa perlu di program lagi secara eksplisit, dan intervensi manusia akan semakin minim [4].



Gambar 2. 7 AI, *Machine Learning*, *Deep Learning* [13]

Secara umum, algoritma pada *machine learning* dirancang untuk memenuhi tiga tujuan utama, yakni: prediksi, klasifikasi, dan pengelompokan (*clustering*).

Prediksi adalah proses di mana mesin mencoba untuk memprediksi nilai *output* berdasarkan data yang telah dimasukkan sebelumnya. Klasifikasi merupakan kegiatan di mana mesin membangun sistem untuk mengelompokkan data ke dalam kategori tertentu, yang kemudian digunakan untuk mengidentifikasi kelompok dari data tersebut. Sementara itu, *clustering* adalah proses di mana mesin mengelompokkan data berdasarkan kemiripan mereka. *Machine learning* dibagi berdasarkan cara pembelajarannya sebanyak tiga kategori, yakni *supervised learning*, *unsupervised learning*, dan *reinforcement learning* seperti pada Gambar 2.8 dibawah ini [14].



Gambar 2. 8 Cara Belajar *Machine Learning* [15]

1) *Supervised learning*

Supervised learning merupakan bagian dari *machine learning* yang mengacu pada pembelajaran yang terarah atau terawasi. Dalam *supervised learning*, algoritma belajar berdasarkan data latih yang telah diberi label, dengan tujuan untuk menggeneralisasi data input. Data latih ini berfungsi sebagai contoh untuk membangun suatu fungsi, yang kemudian akan digunakan untuk melakukan prediksi terhadap data *output*. Untuk menguji sejauh mana akurasi fungsi tersebut, digunakan data uji. Data uji ini berperan dalam menguji kinerja atau performa dari model yang telah dilatih dengan data latih.

Dengan demikian, *supervised learning* memanfaatkan data yang diketahui labelnya untuk membentuk model yang dapat digunakan untuk prediksi pada data baru yang belum dilihat sebelumnya. Model ini berusaha untuk belajar dari pola yang ada dalam data latih sehingga dapat mengenali pola serupa pada data uji atau

data baru. [16].

2) *Unsupervised Learning*

Unsupervised learning adalah salah satu tipe pembelajaran mesin di bidang kecerdasan buatan (*artificial intelligence*) di mana komputer atau sistem akan mempelajari data yang tidak memiliki petunjuk maupun label yang jelas. Dalam metode ini, komputer mencoba untuk mengidentifikasi pola, struktur, dan hubungan dalam data tanpa bantuan langsung dari manusia.

Pada dasarnya, *unsupervised learning* berusaha untuk menemukan informasi tersembunyi dalam data tanpa mengetahui hasil yang diharapkan. Dalam beberapa kasus, ini dapat membantu mengungkap wawasan baru dan mendapatkan pemahaman yang lebih dalam tentang data yang ada.

Contoh umum dari tipe *unsupervised learning* adalah *clustering*, di mana data dikelompokkan berdasarkan kesamaan fitur atau atribut. Dengan demikian, data yang memiliki kemiripan akan dikelompokkan bersama dalam satu kelompok. Ini dapat membantu mengidentifikasi pola alami atau klasifikasi data yang sebelumnya tidak terlihat [17].

3) *Reinforcement learning*

Reinforcement learning adalah salah satu cabang dari *machine learning* yang fokus pada bagaimana mesin belajar untuk mengambil tindakan dalam suatu lingkungan dengan tujuan untuk memaksimalkan *reward* yang diberikan oleh lingkungan tersebut. Dalam *reinforcement learning*, mesin akan melakukan tindakan berulang kali dan belajar dari hasil tindakannya untuk mencapai tujuan tertentu. Dengan menggunakan pendekatan *trial and error*, mesin secara berangsur-angsur meningkatkan kinerjanya. Untuk mencapai tujuan, mesin menggunakan konsep keadaan (*state*) untuk merepresentasikan informasi tentang lingkungan pada waktu tertentu. Mesin juga menggunakan kebijakan (*policy*) untuk menentukan tindakan yang seharusnya diambil dalam setiap keadaan. Kebijakan ini adalah fungsi yang memetakan keadaan menjadi tindakan yang diambil oleh mesin.

Reinforcement learning digunakan dalam berbagai aplikasi, seperti permainan video, robotika, kendali otomatis, sistem rekomendasi, manajemen sumber daya, dan banyak lagi. [18].

Dalam pengembangan model *machine learning*, terdapat dua istilah penting yaitu *training data* dan juga *testing data*. *Training* merupakan suatu proses pembangunan model, sedangkan *testing* merupakan suatu proses untuk menguji kinerja dari model pembelajaran.

Dataset adalah kumpulan data atau sampel yang digunakan dalam pembuatan dan evaluasi model *machine learning*. Umumnya, *dataset* terbagi atas tiga jenis yang tidak tumpang tindih [19]:

1. *Training set* adalah sekumpulan data yang digunakan dalam melatih atau membangun suatu model. Data dalam *training set* digunakan untuk melatih model bagaimana cara mengenali pola dan beradaptasi dengan data yang diberikan.
2. *Development set* atau biasa dikenal juga dengan *validation set* adalah data yang digunakan untuk mengoptimalkan model selama proses pelatihan. Biasanya, kinerja model saat pelatihan diuji menggunakan *validation set*. Penggunaan *validation set* berguna untuk memastikan model dapat mengenali pola secara umum dan mencegah *overfitting*.
3. *Testing set* adalah data yang digunakan untuk pengujian model setelah proses pelatihan selesai. Data ini tidak digunakan selama proses pelatihan, sehingga pengujian dilakukan pada data yang tidak pernah "dilihat" oleh model sebelumnya. Penggunaan *testing set* membantu dalam mengevaluasi kinerja model secara objektif [20].

2.2.3 Pengolahan Citra Digital

Citra merupakan representasi visual dari suatu objek atau gambar yang dibentuk oleh cahaya atau sinar elektromagnetik yang dipantulkan atau dipancarkan oleh objek tersebut. Dalam arti yang lebih umum, citra adalah gambar atau visualisasi dari dunia nyata yang dapat diabadikan oleh perangkat optik, seperti kamera atau mata manusia. Secara khusus, dalam bidang ilmu komputer dan pengolahan citra, citra sering diartikan sebagai representasi digital dari data visual dalam bentuk matriks atau kumpulan piksel. Setiap piksel dalam citra digital merepresentasikan nilai intensitas cahaya pada lokasi tertentu di gambar. Citra digital terdiri dari sekumpulan piksel yang tersusun dalam baris dan kolom, membentuk matriks dua dimensi $f(x, y)$, di mana x dan y adalah koordinat bidang

dan amplitudo dari setiap pasangan koordinat (x, y) disebut sebagai intensitas citra pada titik tersebut. Citra dapat berbentuk gambar hitam-putih (citra keabuan) atau gambar berwarna (citra RGB atau citra dengan sistem warna lainnya). Citra berwarna menggunakan kombinasi tiga saluran warna utama, yaitu merah (R), hijau (G), dan biru (B), untuk membentuk berbagai variasi warna yang dapat dilihat oleh mata manusia [21].

Citra digital adalah representasi daripada citra dua dimensi yang terbentuk atas sejumlah titik terbatas yang biasanya disebut sebagai elemen gambar atau piksel. Setiap piksel direpresentasikan dengan nilai numerik, dimana untuk citra monokrom (keabuan), nilai tersebut mencerminkan intensitas dari piksel (biasanya dalam rentang $[0, 255]$); sementara untuk citra berwarna, terdapat tiga nilai komponen, yaitu merah (*Red*), hijau (*Green*), dan biru (*Blue*) atau sering disebut sebagai sistem warna RGB. Umumnya, nilai numerik yang digunakan adalah bilangan biner 8 bit. Pengolahan citra digital merujuk pada proses pemrosesan gambar digital dengan menggunakan komputer digital. Ini mencakup berbagai teknik dan algoritma untuk mengubah, memodifikasi, dan menganalisis citra, termasuk teknik pengolahan citra, analisis citra, pengenalan pola, segmentasi, dan lainnya [21].

Hasil dari pencuplikan dan kuantisasi citra ialah sebuah matriks yang memuat nilai-nilai riil. Untuk mempresentasikan sebuah citra digital dapat dilakukan dengan dua cara. Dapat diasumsikan bahwa sebuah citra $f(x, y)$ jika di cuplik dan menghasilkan M baris dan N kolom, maka diketahui bahwa ukuran citra adalah $M \times N$.

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N - 1) \\ f(1,0) & f(1,1) & \dots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1,0) & f(M - 1,1) & \dots & f(M - 1, N - 1) \end{bmatrix} \quad (2.1)$$

Sisi kanan dari persamaan ini adalah sebuah citra digital berdasarkan definisi, setiap elemen dari array ini dinamakan dengan elemen citra atau piksel.

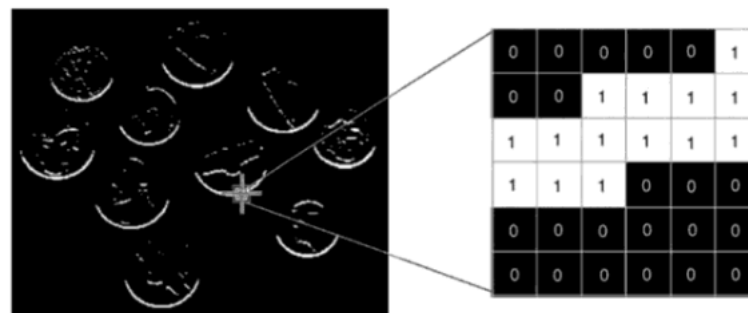
2.2.3.1 Citra Biner

Citra biner adalah citra digital yang menggunakan dua warna saja, yaitu hanya

warna hitam dan putih. Dalam representasi nilai biner 8 bit, warna hitam direpresentasikan sebagai 0000 0000, sedangkan warna putih direpresentasikan sebagai 1111 1111 seperti pada Gambar 2.9. Citra biner sering digunakan dalam pemrosesan citra dan analisis karena hanya memuat informasi yang sangat sederhana dan mudah diolah. Citra biner sering dihasilkan melalui proses *thresholding*, di mana ambang batas (*threshold*) ditentukan untuk memisahkan nilai intensitas menjadi dua kategori, yaitu hitam dan putih. Pixel yang memiliki nilai intensitas di atas ambang batas akan diubah menjadi putih, sementara pixel dengan nilai intensitas di bawah atau sama dengan ambang batas akan diubah menjadi hitam. Aplikasi utama citra biner adalah dalam pemrosesan dan analisis citra, seperti deteksi tepi, segmentasi objek, dan pengenalan pola. Citra biner juga digunakan dalam operasi matematis dan logika, seperti operasi *AND*, *OR*, dan *XOR* untuk memproses data biner. Keuntungan dari citra biner adalah representasi datanya sangat sederhana dan ringkas, yang memudahkan dalam analisis dan pemrosesan. Namun, citra biner juga memiliki kelemahan, yaitu informasi warna atau tingkat keabuan dari citra asli hilang sehingga detail dan tekstur mungkin tidak terlihat dengan jelas dalam citra biner. Oleh karena itu, citra biner lebih cocok digunakan dalam aplikasi khusus yang membutuhkan representasi data biner seperti pengolahan dan analisis citra sederhana [22]. Rumus untuk menghasilkan citra biner dapat dituliskan sebagai berikut:

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad (2.2)$$

Dimana $d(x,y)$ adalah citra biner dari citra *grayscale* $f(x,y)$, dan T menyatakan nilai *threshold*.



Gambar 2. 9 Citra biner dengan nilai piksel (6 × 6) [23]

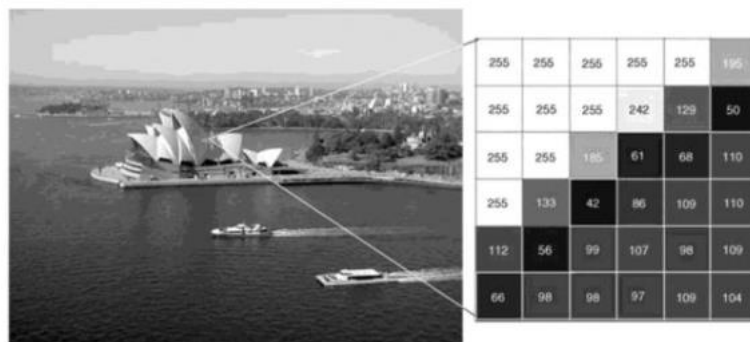
2.2.3.2 Citra Keabuan

Citra keabuan (*grayscale image*) yang juga dikenal sebagai citra monokrom, merupakan citra digital yang memiliki 256 tingkat keabuan atau level intensitas warna. Dalam citra keabuan, nilai intensitas atau tingkat keabuan umumnya diukur dalam skala abu-abu (*grayscale*) dari 0 hingga 255, di mana 0 mewakili hitam (tidak ada cahaya) dan 255 mewakili putih (cahaya maksimum). Semakin tinggi nilai intensitas, semakin terang piksel tersebut, contoh ilustrasi dari citra keabuan dapat dilihat pada Gambar 2.10. Citra keabuan dapat dienkodesikan sebagai *array* 2D dengan setiap piksel direpresentasikan oleh 8 bit. Keuntungan dari citra keabuan adalah data yang dihasilkan lebih ringkas karena hanya menggunakan satu saluran. Selain itu, citra keabuan sering digunakan dalam beberapa aplikasi seperti deteksi tepi, segmentasi, dan analisis citra di mana warna tidak memainkan peran penting. Untuk kekurangan dari citra keabuan yaitu kehilangan informasi warna yang dapat menyampaikan makna tertentu atau mengidentifikasi objek dengan lebih jelas. Oleh karena itu, dalam beberapa kasus, citra berwarna (*RGB image*) lebih sering digunakan karena memberikan lebih banyak informasi dalam pemrosesan dan analisis citra [22].

Persamaan untuk citra keabuan dapat dituliskan sebagai berikut:

$$\text{Gray Scale} = 0.30R + 0.59G + 0.11B \quad (2.3)$$

Dimana R=*Red* (Merah) berkontribusi 30%, G = *Green* (Hijau) berkontribusi 59%, B=*Blue* (Biru) berkontribusi 11%.



Gambar 2. 10 Citra keabuan dengan nilai piksel (6 × 6) [23]

2.2.3.3 Citra RGB

Representasi citra warna memiliki tingkat kompleksitas yang lebih tinggi dan beragam, dengan cara paling umum untuk menyimpan isi citra warna adalah melalui representasi RGB. Setiap piksel dalam citra RGB direpresentasikan oleh nilai 24-bit yang menggunakan tiga *array* 2D yang berukuran sama. Setiap elemen dalam *array* ini menyimpan nilai 8-bit yang mengidentifikasi intensitas warna merah (R), hijau (G), dan biru (B) dalam skala [0, 255]. Kombinasi dari ketiga nilai tersebut menghasilkan total 2^{24} (16777216 atau 16 juta) kombinasi warna yang berbeda [22]. Citra RGB dapat dinyatakan dengan persamaan berikut:

$$f(x; y) = fr(x; y)j . dg(x; y)j . fb(x; y)j \quad (2.4)$$

Dimana $fr(x,y)$ adalah nilai intensitas warna merah, $fg(x,y)$ adalah nilai intensitas warna hijau, dan $fb(x,y)$ adalah nilai intensitas warna biru pada piksel (x,y) dalam citra RGB.

Secara umum untuk mengubah citra RGB ke citra *grayscale*, dapat digunakan persamaan yang menghasilkan nilai tingkat keabuan untuk setiap piksel dalam citra *grayscale* sebagai berikut [22]:

$$Gray = I = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B \quad (2.5)$$

Dimana I adalah tingkat keabuan, R = *Red* (Merah), G = *Green* (Hijau), B=*Blue* (Biru).

2.2.4 Histogram

Histogram adalah representasi grafis dari penyebaran nilai-nilai intensitas piksel daripada suatu citra digital ataupun bagian tertentu pada sebuah citra. *Histogram* citra menggambarkan frekuensi relative atas kemunculan suatu intensitas pada citra tersebut. *Histogram* memberikan informasi tentang kecerahan (*brightness*) pada suatu citra dan juga kontras (*contrast*) pada citra seperti pada Gambar 2.11. Sebagai alat bantu dalam pengolahan citra, *histogram* memiliki nilai dalam analisis baik secara kualitatif maupun kuantitatif [26]. Dari sebuah histogram dibutuhkan normalisasi dengan jumlah seluruh pixel,

histogram yang telah dinormalisasi dapat diketahui frekuensi kemunculan nisbi dari nilai-nilai intensitas dalam citra tersebut. Normalisasi histogram dihitung dengan rumus :

$$h_i = \frac{n_i}{n}$$

(2. 6)

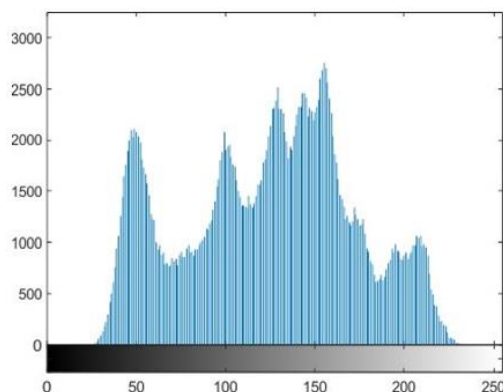
Dimana n_i adalah jumlah pixel yang memiliki derajat keabuan i , dan n adalah jumlah seluruh pixel didalam citra. Sedangkan i merupakan satu interval yang digunakan untuk mengelompokkan data.

Beberapa indikasi yang dapat diperoleh dari histogram suatu citra adalah sebagai berikut:

- Pada citra gelap: *Histogram* akan memiliki konsentrasi level keabuan pada bagian bawah atau sisi paling rendah.
- Pada citra terang: *Histogram* akan memiliki konsentrasi level keabuan pada bagian atas atau sisi paling tinggi.
- Pada citra dengan kontras rendah yang signifikan: *Histogram* akan memiliki konsentrasi level keabuan yang terpusat di sekitar nilai tengah dan padat.
- Pada citra dengan kontras tinggi yang signifikan: *Histogram* akan memiliki level keabuan yang terdistribusi luas dan menyebar di berbagai nilai intensitas [24].



(a) Citra Grayscale



(b) Histogram Citra

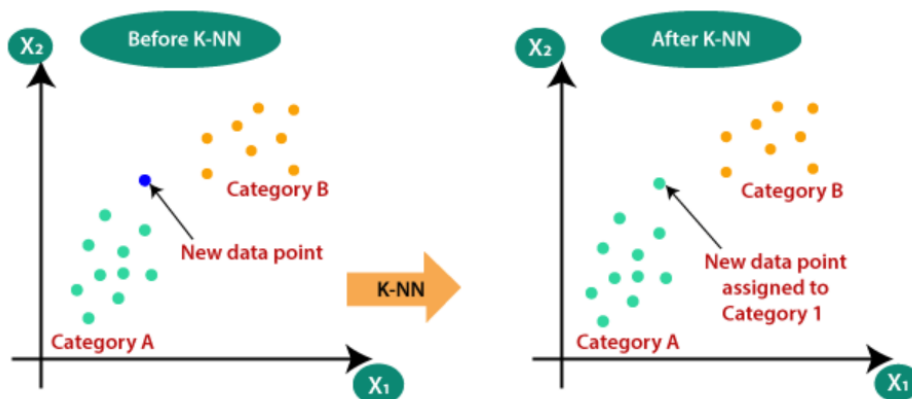
Gambar 2. 11 *Histogram Citra* [25]

2.2.5 K- Nearest Neighbor (K-NN)

K-Nearest Neighbor (KNN) adalah metode klasifikasi yang digunakan untuk mengelompokkan data berdasarkan pembelajaran dari data yang sudah memiliki klasifikasi sebelumnya. KNN (*K-Nearest Neighbors*) adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. *Hyperparameter* utama dalam algoritma ini adalah K. KNN termasuk dalam kategori *supervised learning*, di mana ketika ada data baru yang akan diklasifikasikan, algoritma ini akan mencari mayoritas dari kategori data terdekat berdasarkan jaraknya dalam KNN. Dalam KNN, terdapat sejumlah objek dengan variabel k atau titik *training* yang paling dekat dengan data *query*. Algoritma KNN menggunakan metode klasifikasi berdasarkan kedekatan dengan mengambil *voting* mayoritas dari kategori pada k objek terdekat. Jarak antara tetangga biasanya dihitung dengan menggunakan *Euclidean Distance* [26].

$$D(a, b) = \sqrt{\sum_{k=1}^d (a_k - b_k)^2} \quad (2.7)$$

Dalam rumus *Euclidean Distance*, D adalah jarak skalar antara dua vektor data training (a dan b) yang akan diklasifikasikan, dengan $a = a_1, a_2, \dots, a_i$ dan $b = b_1, b_2, \dots, b_i$ sebagai nilai atribut dimensi yang mewakili data tersebut. Dimana d adalah jumlah dimensi dalam ruang, a_k adalah koordinat dari titik pertama dalam dimensi ke-i, dan b_k adalah koordinat dari titik kedua dalam dimensi ke-i



Gambar 2. 12 K-Nearest Neighbor [27]

Dapat dilihat pada Gambar 2.12 dimana terdapat suatu data baru yang belum diketahui akan masuk dalam kategori kelas mana. Nilai dari hasil klasifikasi KNN yang menentukan jumlah tetangga terdekat yang akan digunakan untuk membuat prediksi. Ketika $K=1$, data akan diklasifikasikan berdasarkan label dari tetangga terdekat. Namun, jika $K>1$, prediksi akan diambil berdasarkan mayoritas label dari tetangga terdekat.

Berikut ialah penjelasan singkat mengenai KNN:

- K : K adalah *hyperparameter* dalam algoritma KNN yang menentukan jumlah tetangga terdekat yang akan digunakan untuk membuat prediksi.
- Jarak: Algoritma KNN mengukur jarak antara data yang akan diprediksi dengan data latih yang ada dalam ruang fitur. Jarak ini dapat dihitung menggunakan berbagai metode seperti jarak *Euclidean*, jarak Manhattan, atau metrik jarak lainnya, tergantung pada kasus penggunaan.
- Proses Klasifikasi: Pada tugas klasifikasi, setelah mengukur jarak antara data yang akan diprediksi dengan data latih, KNN akan mengidentifikasi K tetangga terdekat tergantung pada nilai K yang ditentukan. Kemudian, algoritma akan menentukan label mayoritas dari tetangga terdekat sebagai prediksi untuk data yang akan diprediksi.
- Proses Regresi: Pada tugas regresi, algoritma KNN akan mengambil rata-rata nilai target dari K tetangga terdekat sebagai nilai prediksi untuk data yang akan diprediksi.

Kelebihan dan Kekurangan: KNN memiliki kelebihan sebagai algoritma sederhana dan mudah diimplementasikan. Namun, algoritma ini memerlukan waktu komputasi yang lebih tinggi saat dataset besar karena harus menghitung jarak untuk semua data latih. Selain itu, memilih nilai K yang tepat juga merupakan tantangan dalam algoritma KNN.

Algoritma KNN berguna ketika data berada dalam kelompok yang terpisah dengan baik, tetapi mungkin tidak akan menghasilkan nilai yang baik apabila data terdistribusi secara acak. Oleh sebab itu, metode yang dipilih dalam pengukuran jarak dan nilai K yang tepat sangat penting dalam penerapan KNN untuk mendapatkan hasil yang optimal [28].

2.2.6 Confusion Matrix

Confusion matrix adalah sebuah metode evaluasi untuk dataset yang terdiri dari dua kelas, di mana satu kelas akan dianggap sebagai kelas yang bernilai positif dan kelas yang lainnya sebagai kelas yang bernilai negatif. *Confusion matrix* digunakan dalam model klasifikasi untuk menghitung nilai presisi, sensitivitas, dan akurasi. Biasanya, nilai-nilai dalam *confusion matrix* diungkapkan dalam bentuk persentase (%) [29].

Perhitungan menggunakan *confusion matrix* sebagai berikut :

- a. Menentukan *precision*. Presisi (*precision*) adalah ukuran akurasi dari kelas yang diprediksi.

$$\frac{TP}{TP + FP} \times 100\% \quad (2.8)$$

TP = *True Positive*, FP = *False Positive*

- b. Menentukan *recall*. *Recall* adalah ukuran kemampuan model prediksi untuk mengidentifikasi instansi kelas tertentu dalam dataset.

$$\frac{TP}{(TP + FN)} \times 100\% \quad (2.9)$$

TP = *True Positive*, FN = *False Negative*

- c. Menentukan *accuracy*. Akurasi (*accuracy*) adalah proporsi dari jumlah total prediksi yang benar.

$$\frac{TP + TN}{Total Sampel} \times 100\% \quad (2.10)$$

- TP = *True Positive* , dimana data yang digunakan merupakan data yang berlabel positif dan diprediksi oleh sistem sebagai kelas positif dengan nilai sesuai kriteria yang ditetapkan.
- TN = *True Negative*, dimana data yang berlabel negatif dan diprediksi oleh sistem sebagai kelas negatif dengan nilai tidak sesuai kriteria yang ditetapkan.
- FP = *False Positive*, dimana data yang berlabel negatif diprediksi oleh sistem sebagai kelas positif dengan nilai sesuai kriteria yang ditetapkan.

- FN = *False Negative*, dimana data yang berlabel positif diprediksi oleh sistem sebagai kelas negatif dengan nilai tidak sesuai kriteria yang ditetapkan [29].

Nilai klasifikasi yang dianggap baik pada klasifikasi citra bervariasi tergantung dari kompleksitas masalah yang diangkat. Tujuan utama untuk mendapatkan nilai klasifikasi tinggi pada klasifikasi citra pada tiga metrik yaitu akurasi, presisi dan *recall*, ialah untuk memberikan gambaran bahwa model yang telah dibuat memiliki kemampuan yang baik dalam membedakan dan mengenali berbagai objek penelitian dalam citra. Namun, tidak ada standar nilai pasti yang bisa diberikan. tetapi pada umumnya yang baik dalam machine learning ialah lebih besar dari nilai 70%. Ukuran realistis dalam nilai klasifikasi ideal yang biasa digunakan dalam standar industri ialah berkisar 70%-90%. Pada beberapa tugas klasifikasi citra yang kompleks, tujuan untuk mencapai tingkat akurasi yang lebih tinggi, misalnya di atas 95%, bisa menjadi relevan. Namun, perlu diingat bahwa nilai-nilai tersebut hanya berfungsi sebagai pedoman umum. Dalam beberapa tugas klasifikasi citra yang sangat spesifik atau sulit, mencapai tingkat akurasi, presisi, dan *recall* yang sangat tinggi bisa menjadi tantangan yang sulit bahkan mungkin tidak dapat dicapai. Selain itu, dalam beberapa situasi, nilai akurasi, presisi, dan *recall* dapat diubah dengan menyesuaikan ambang batas keputusan (*threshold*) untuk mengoptimalkan kinerja model sesuai dengan kebutuhan khusus proyek tersebut. Karena itu, dalam pemilihan metrik dan target nilai, perlu mempertimbangkan karakteristik khusus dari setiap tugas klasifikasi citra [30].

2.2.7 K- Fold Cross Validation

K-Fold Cross Validation adalah salah satu teknik yang biasa digunakan dalam pengukuran *cross validation* yang digunakan untuk mengukur kinerja dari suatu algoritma dengan membagi sampel data menjadi K kelompok secara acak. Setiap kelompok menjadi data uji satu kali, sementara kelompok yang lain digunakan sebagai data latih [31]. Hal ini dilakukan secara berulang-ulang hingga setiap kelompok menjadi data uji seperti pada Gambar 2.13.

Cross validation adalah metode yang berguna untuk mengevaluasi efisiensi suatu model dengan melatih model tersebut pada subset data input dan mengujinya pada subset data input yang belum digunakan sebelumnya. Tujuan utamanya adalah

untuk mendapatkan informasi tentang performa model algoritma melalui percobaan sebanyak K kali. Metode *cross validation* juga membantu meningkatkan performa model yang digunakan serta membantu dalam pengolahan *dataset* [32].

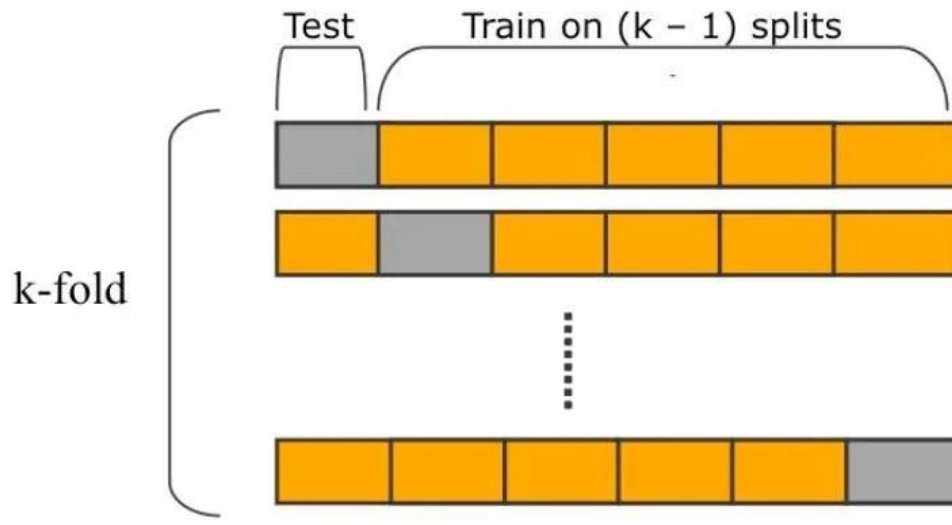
Dengan menggunakan *K-Fold Cross Validation*, kita dapat memastikan bahwa model algoritma tidak hanya mengandalkan satu pembagian data latih dan data uji saja, melainkan mengambil rata-rata performa dari beberapa percobaan yang berbeda, sehingga memberikan hasil evaluasi yang lebih konsisten dan dapat diandalkan. Dengan teknik *cross validation*, kita dapat memiliki gambaran yang lebih akurat tentang seberapa baik model algoritma bekerja dan seberapa baik model tersebut generalisasi pada data yang belum pernah dilihat sebelumnya [33].



Gambar 2. 13 K-Fold Cross Validation [34]

2.2.8 Leave One Out Cross Validation

Leave one out - cross validation (LOO) adalah salah satu teknik *cross validation* yang populer dalam pengujian model atau algoritma *machine learning*. Teknik ini digunakan dengan cara membagi atau melakukan *split* terhadap *dataset* menjadi dua bagian, yaitu salah satu data akan menjadi data uji (*testing data*), dan data lainnya akan menjadi data latih (*training data*). Setiap data uji hanya berisi satu data, sedangkan data latih berisi seluruh data kecuali data uji tersebut seperti pada ilustrasi di Gambar 2.14. Proses ini diulang sebanyak jumlah data yang ada dalam *dataset*. Misalnya, jika *dataset* memiliki N data, maka *cross validation* LOO akan melakukan proses pengujian sebanyak N kali, di mana setiap kali akan memilih satu data berbeda sebagai data uji dan menggunakan data-data lainnya sebagai data latih [35].



Gambar 2. 14 *Leave One Out – Cross Validation* [36]

Keuntungan dari teknik LOO adalah setiap data pada *dataset* digunakan sebagai data uji, sehingga memungkinkan untuk mendapatkan evaluasi yang lebih akurat tentang performa model. Namun, kelemahannya adalah jika *dataset* sangat besar, proses ini akan menjadi sangat mahal komputasionalnya karena harus melakukan pelatihan model sebanyak N kali [37].