

BAB II

TINJAUAN PUSTAKA

2.1 Songket

Songket adalah jenis kain tenunan tradisional Melayu dan Minangkabau di Indonesia, Malaysia, dan Brunei. Songket digolongkan dalam keluarga tenunan brokat. Songket ditenun dengan tangan dengan benang emas dan perak dan pada umumnya dikenakan pada acara – acara resmi. Keindahan kain songket ini diharapkan bisa menarik wisatawan dalam negeri maupun mancanegara yang suka dengan seni kain tradisional Indonesia (Salamah & Kusmanto, 2017).

Salah satu kain tenun tradisional yang sudah dikenal luas hingga ke manca negara, dengan kualitas dan keunikan motifnya adalah songket Batu Bara. Keunikan corak dan bahan kainnya menjadi salah satu daya tarik kain ini diminati hingga keluar negeri. Ketertarikan konsumen dengan industri kerajinan songket Batu Bara, karena desain atau motif tenunan asal daerah Batu Bara memiliki nilai seni budaya yang cukup tinggi. Industri kerajinan rumah tangga ini sudah berjalan dari turun-temurun sejak ratusan tahun silam (Ratna, 2021).

Tenun songket melayu Batu Bara merupakan kain hasil kerajinan tangan orang-orang Melayu yang dilakukan dengan melalui proses menenun benang yang diselingi dengan tenunan benang emas atau benang perak dengan ragam motif/corak tenunan tertentu. Pengrajin tenunan songket melayu Batu Bara memiliki keunikan dan kaya akan nilai keindahan dan estetika sebagai gabungan unsur-unsur budaya yang melambangkan corak, pandangan dan pemikiran masyarakat Melayu. Ragam motif/corak kain tenunan songket sangat erat hubungannya antara manusia dengan alam baik hewan maupun tumbuhan. Ragam ini juga mencerminkan cara dan pandangan hidup umat manusia. (Ria Afrilia, 2018).

Ciri khas kain tenun songket Batu Bara dapat dilihat dari proses pembuatan kain songket yang hingga saat ini masih menggunakan alat tenun dari kayu dengan cara tradisional, namun tetap memiliki kualitas yang baik. Dalam pemilihan warna kain, songket Batu Bara memiliki berbagai jenis warna seperti biru, merah jambu, biru muda, hijau laut, kuning, merah hati, krem, ungu, merah

muda dan kombinasi warna lainnya. Kain tenunan Batu Bara dibuat dari benang polyester dan benang emas, ditenun dengan peralatan yang cukup sederhana / tradisional, dikerjakan oleh tangan pengrajin yang terampil dengan penyusunan benang. Dari ragam hias ilmu ukur (geometris) dan tumbuh - tumbuhan terlihat motif – motif kain yang diciptakan seperti : motif pucuk betikam, pucuk perak, pucuk pandan, pucuk caul, pucuk rebung, pucuk kol, pucuk cempaka, bunga bertabur, bunga dadu, bunga jambangan anggur, cengkeh, mawar bertabur (Ratna, 2021).

2.2 Citra

Citra merupakan fungsi menerus (*continue*) atas intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, objek memantulkan kembali seluruh atau sebagian berkas cahaya kemudian ditangkap oleh alat optik atau elektro-optis (Sriani, 2017).

Citra adalah suatu representasi, kemiripan, atau imitasi dari suatu obyek atau benda. Sebuah citra mengandung informasi tentang obyek yang direpresentasikan. Citra dapat dikelompokkan menjadi citra tampak dan citra tak tampak. Untuk dapat dilihat mata manusia, citra tak tampak harus dirubah menjadi citra tampak, misalnya dengan menampilkannya di monitor, dicetak di kertas dan sebagainya. Salah satu citra tak tampak adalah citra digital. Citra dapat juga didefinisikan sebagai sumber dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses sampling. Gambar analog dibagi menjadi N baris dan M kolom sehingga menjadi gambar diskrit. Persilangan antara baris dan kolom tertentu disebut dengan piksel (Sriani, 2017).

2.2.1 Jenis Citra

Pada umumnya citra dibagi menjadi 2 jenis di antaranya adalah :

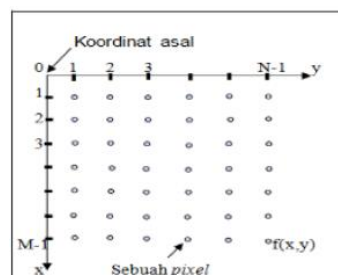
A. Citra Analog

Citra analog adalah citra yang bersifat kontinu yang diperoleh dari sistem optik yang menerima sinyal analog. Citra analog tidak dapat direpresentasikan dalam komputer, sehingga tidak bisa diproses di komputer, citra analog harus dinversi menjadi citra digital (Andono et al., 2017).

B. Citra Digital

Citra digital adalah citra yang dapat diolah oleh komputer. Sedangkan citra yang dihasilkan dari peralatan digital (citra digital) langsung bisa diolah oleh komputer, karena di dalam peralatan digital terdapat system sampling dan kuantisasi. Sedangkan peralatan analog tidak dilengkapi kedua sistem tersebut. Sistem sampling adalah sistem yang mengubah citra kontinu menjadi citra digital dengan cara membagi citra analog menjadi M baris dan N kolom, sehingga menjadi citra diskrit. Semakin besar nilai M dan N, semakin halus citra digital yang dihasilkan. Pertemuan antara baris dan kolom disebut piksel. Sistem kuantisasi adalah sistem yang melakukan perubahan intensitas analog ke intensitas diskrit, sehingga dengan proses ini dimungkinkan untuk membuat gradasi warna sesuai dengan kebutuhan. Kedua sistem inilah yang bertugas untuk memotong-motong citra menjadi M baris dan N kolom (proses sampling) sekaligus menentukan besar intensitas yang terdapat di titik tersebut (proses kuantisasi), sehingga menghasilkan resolusi citra yang diinginkan.

Telah diketahui bahwa hasil sampling dan kuantisasi dari sebuah citra adalah bilangan *real* yang membentuk sebuah matriks M baris dan N kolom. Ini berarti ukuran citra adalah $M \times N$. Secara umum, sistem koordinat yang dipergunakan untuk mewakili citra dalam teori pengolahan citra seperti digambarkan pada Gambar 2.1. Sebuah citra digital diwakili oleh matriks yang terdiri dari M baris dan N kolom, dimana perpotongan antara baris dan kolom disebut piksel. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah $f(x,y)$, yaitu besar intensitas atau warna dari piksel di titik itu (Andono et al., 2017).



Gambar 2. 1 Sistem koordinat yang dipergunakan untuk mewakili citra

(Sumber : Andono et al., 2017)

Artinya, sebuah citra digital dapat ditulis dalam matriks berikut :

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

Berdasarkan gambaran di atas, maka secara matematis citra digital dapat dituliskan sebagai fungsi intensitas $f(x, y)$, dimana harga x (baris) dan y (kolom) merupakan koorditas posisi dan $f(x, y)$ adalah nilai fungsi pada setiap titik (x, y) yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari piksel di titik tersebut (Andono et al., 2017).

1. Jenis Citra Digital

Dalam pemrosesan citra, terdapat tiga jenis citra yang umum digunakan yaitu:

a. Citra Berwarna

Citra berwarna atau yang sering disebut citra RGB adalah jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Setiap komponen warna tersebut menggunakan delapan bit yang nilainya berkisar antara 0 sampai dengan 255. Dengan demikian, komponen warna yang dapat disajikan mencapai 16.581.375 warna. Tabel 2.1 berikut menjelaskan contoh warna dan nilai RGB-nya (Neneng et al., 2016).

Tabel 2. 1 Warna dan nilai penyusun warna
(Sumber : Neneng et al., 2016).

Warna	R	G	B
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255
Kuning	0	255	255

b. Citra Berskala Keabuan

Citra jenis ini menangani gradasi warna hitam dan putih yang menghasilkan efek warna abu-abu. Pada jenis gambar ini, warna dinyatakan dengan intensitas yang berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam, sedangkan nilai 255 menyatakan putih (Neneng et al., 2016).

c. Citra Biner

Citra biner merupakan citra yang setiap pikselnya hanya dinyatakan dengan nilai dari dua kemungkinan yakni 0 dan 1. Nilai 0 menyatakan warna hitam, sedangkan nilai 1 menyatakan putih.

Dalam prakteknya, mengingat banyak pemrosesan yang bekerja pada sala keabuan, maka citra berwarna seringkali harus dikonversi terlebih dahulu ke dalam bentuk citra berskala keabuan. Untuk mengubah citra RGB menjadi citra sala keabuan dapat dilakukan dengan rumus sebagai berikut :

$$\text{Skala keabuan} = 0.299R + 0.587G + 0.144 B$$

Atau

$$\text{Skala keabuan} = 0.333R + 0.333G + 0.333 B$$

2. Elemen Dasar Citra Digital

Citra digital memiliki beberapa elemen dasar antara lain sebagai berikut:

a. Kecerahan (*Brightness*)

Kecerahan (*Brightness*) merupakan cahaya yang dipancarkan piksel dari citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah titik (piksel) didalam citra yang merupakan intensitas rata-rata dari suatu area yang melingkupinya (Nazariana et al., 2018).

b. Kontras (*Contrast*)

Kontras (*Contrast*) merupakan sebaran terang dan gelap dalam sebuah citra (Nazariana et al., 2018).

c. Kontur (*Contur*)

Kontur (*Contur*) adalah keadaan yang ditimbulkan oleh perubahan

intensitas pada piksel-piksel yang bertetangga. Karena adanya perubahan intensitas inilah mata mampu mendeteksi tepi-tepi objek didalam citra (Nazariana et al., 2018).

d. Warna

Warna sebagai persepsi yang ditangkap sistem visual terhadap panjang gelombang cahaya yang dipantulkan oleh objek (Nazariana et al., 2018).

e. Bentuk (*Shape*)

Bentuk adalah *property intrinsic* dari objek 3 dimensi, dengan pengertian bahwa bentuk merupakan properti intrinsik utama untuk sistem visual manusia (Nazariana et al., 2018).

f. Tekstur (*Texture*)

Tekstur dicirikan sebagai distribusi spasial dari derajat keabuan didalam sekumpulan piksel-piksel yang bertetangga. Tekstur adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut. Informasi tekstur dapat digunakan untuk membedakan sifat- sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus, juga sifat- sifat spesifik dari kekasaran dan kehalusan permukaan tadi yang sama sekali terlepas dari warna permukaan tersebut (Nazariana et al., 2018)

2.2.2 Format File Citra Digital

Hampir semua format file citra yang dipakai untuk mempresentasikan citra-citra bitmap memuat sebuah file header yang diikuti dengan data piksel. File header dari citra menyimpan informasi seputar citra tersebut, seperti tinggi dan lebar citra, banyak pita, banyak bit per piksel, dan sejumlah byte sidik digital yang mengindikasikan tipe file. Pada format file yang lebih kompleks, file header juga memuat informasi tentang jenis kompresi yang digunakan dan parameter-parameter lain yang diperlukan untuk mendekode citra.

Format file paling sederhana adalah format BIN dan PPM. Format BIN hanya memuat data piksel mentah tanpa file header. Jadi, pengguna dari file BIN harus mengetahui parameter-parameter citra yang relevan seperti tinggi dan lebar citra

sebelum menggunakan citra tersebut. Format PPM dan varian-varianya yaitu PBM untuk citra biner, PGM untuk citra keabuan, PPM untuk citra warna, dan PNM untuk semuanya secara luas digunakan dalam penelitian pemrosesan citra dan banyak perangkat lunak grafis untuk konversi ke format ini.

Format BMP (*Microsoft windows bitmap*) juga secara luas dipakai dan merupakan format yang cukup sederhana, yang memuat sebuah file header yang diikuti dengan data piksel mentah.

Format JPEG merupakan format file yang paling populer untuk representasi citra dengan kualitas fotografis. Format ini dapat memiliki derajat kompresi yang tinggi dengan rugi kualitas perseptual minimal. Detail-detail teknis dari algoritma kompresi JPG.

Dua format file citra lain yang secara luas dipakai dalam pekerjaan-pekerjaan pemrosesan citra GIF (*graphics interchange format*) dan TIFF (*tagged image file format*). GIF menggunakan sebuah representasi berindeks untuk citra-citra warna dengan peta warna memuat maksimum 256 warna, algoritma kompresi LZW (*Lempel ziv welch*), dan header berukuran 13 byte. TIFF merupakan format yang lebih kompleks dengan sejumlah opsi dan kapabilitas, mencakup kemampuan untuk merepresentasikan citra warna 24 bpp dan mendukung lima skema kompresi yang berbeda.

Format PNG (*portable network graphics*) merupakan format file yang semakin populer belakangan ini yang mendukung baik citra warna 24 bpp maupun citra warna berindeks.

File gambar berfungsi untuk menyimpan sebuah gambar yang dapat ditampilkan di layar ke dalam suatu media penyimpanan data. Untuk penyimpanan tersebut digunakan format gambar. Setiap format gambar memiliki karakteristik masing-masing. Beberapa format umum saat ini, yaitu bitmap (.bmp), *tagged image format* (.tif, tiff), portable network graphics (.png), *graphics interchange format* (.gif), jpeg (.jpg), mpeg (.mpg), dan lain-lain.

Untuk kepentingan memudahkan dalam proses pengolahan citra, perlu diketahui berbagai macam format citra yang didapat digunakan atau diproses. Seperti dalam tabel 2.2 di bawah ini. (Vivian, 2020).

Tabel 2. 2 Daftar Format File Gambar
(Sumber : Vivian, 2020)

Format Gambar	Ekstensi	Keterangan
TIFF	.tiff, .tif	<i>Tagged Image File Format</i> merupakan format citra yang mula-mula dibuat oleh Aldus, kemudian dikembangkan oleh <i>Microsoft</i> dan terakhir oleh Adobe.
JPEG	.jpg, .jpeg	<i>Joint Photographics Expert Group</i> adalah format citra yang dirancang agar dapat memampatkan data dengan rasio 1 : 16
GIF	.gif	<i>Graphics Interface Format</i> merupakan format yang memungkinkan pemampatan data hingga 50%. Cocok untuk citra yang memiliki area cukup besar dengan warna sama.
BMP	.bmp	<i>Windows Bitmap</i> merupakan format bitmap pada <i>windows</i>
PNG	.png	<i>Portable Network Graphics</i> biasa dibaca ping. Asal mulanya dikembangkan sebagai format GIF karena adanya penerapan lisensi GIF. Format ini mendukung pemampatan data tanpa menghilangkan informasi aslinya.

2.2.3 Pengolahan Citra

Pengolahan citra adalah salah satu area paling penting dari aplikasi multimedia. Aplikasi tersebut dapat ditemukan hampir dimana saja pada dunia modern saat ini. Oleh karena itu, jumlah orang yang bekerja dengan citra semakin meningkat pesat, ini berarti bahwa permintaan tool untuk pengolahan citra juga bertumbuh. Citra yang dikompresi, didekompresi, dikirim melalui internet, dimodifikasi atau bahkan terdistorsi dalam berbagai cara, ini semua memengaruhi kualitas citra, memprediksi dan memperbaikinya. Pada gilirannya, kriteria kualitas

dapat digunakan sebagai fungsi tujuan, dimasukkan dalam desain, seperti teknik pengolahan citra kompresi, peningkatan dan pemulihan (Andono et al., 2017).

Pengolahan citra merupakan suatu metode atau teknik yang dapat digunakan untuk memproses citra atau gambar dengan jalan memanipulasinya menjadi suatu data gambar yang diisikan untuk mendapatkan suatu informasi tertentu mengenai objek yang sedang diamati. Pengolahan citra digital dilakukan dengan komputer digital, maka citra yang akan diolah terlebih dahulu ditransformasikan kedalam bentuk besaran-besaran diskrit dari nilai tingkat keabuan pada titik element citra. Bentuk dari citra ini disebut citra digital. Elemen-elemen citra digital apabila ditampilkan dalam layer monitor akan menempati sebuah ruang yang disebut piksel (*picture element*) (Putri, 2016).

2.3 Ekstraksi Fitur

Ekstraksi fitur merupakan suatu pengambilan ciri dari suatu bentuk yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Ekstraksi Fitur bertujuan untuk mencari daerah fitur yang signifikan pada gambar tergantung pada karakteristik intrinsik dan aplikasinya.

Ekstraksi fitur dilakukan dengan cara menghitung jumlah titik atau piksel yang ditemui dalam setiap pengecekan, dimana pengecekan dilakukan dalam berbagai arah *tracing* pengecekan pada koordinat kartesian dari citra digital yang di analisis, yaitu vertikal, horizontal, diagonal kanan, dan diagonal kiri. Ekstraksi fitur terbagi menjadi tiga macam yaitu ekstraksi fitur bentuk, ekstraksi fitur tekstur, ekstraksi fitur warna (Doavers,2018).

2.4 *Gray Level Co-Occurrence Matrix* (GLCM)

Gray Level Co-Occurrence Matrix (GLCM) adalah suatu metode yang digunakan untuk analisis tekstur/ekstraksi ciri. GLCM merupakan suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra.

Koordinat pasangan piksel memiliki jarak d dan orientasi sudut θ . Jarak direpresentasikan dalam piksel dan sudut direpresentasikan dalam derajat. Orientasi sudut terbentuk berdasarkan empat arah suduh yaitu 0° , 45° , 90° , dan

135° dan jarak antar piksel sebesar 1 piksel (Arry Supriyanto et al., 2018)

Tahapan yang dilakukan pada perhitungan GLCM adalah sebagai berikut :

1. Pembentukan matriks awal GLCM dari pasangan dua piksel yang berjajar sesuai dengan arah 0°, 45°, 90°, dan 135°.
2. Membentuk matriks yang simetris dengan menjumlahkan matriks awal GLCM dengan nilai transposnya.
3. Menormalisasi matriks GLCM dengan membagi setiap elemen matriks dengan jumlah pasangan piksel.
4. Ekstraksi ciri, yaitu :

$$Contrast = \sum_{i_1} \sum_{i_2} (i_1 - i_2)^2 p(i_1, i_2) \quad (1)$$

$$Homogeneity = \sum_{i_1} \sum_{i_2} \frac{p(i_1, i_2)}{1 + |i_1 - i_2|} \quad (2)$$

$$Energy = \sum_{i_1} \sum_{i_2} p^2(i_1, i_2) \quad (3)$$

$$Correlation = \sum_i \sum_j \frac{(i - \mu_i)(j - \mu_j)p(i, j)}{\sigma_i \sigma_j} \quad (4)$$

2.5 Klasifikasi

Klasifikasi/*clustering* adalah algoritma yang mampu mengklasifikasikan/meng-*cluster* objek berdasarkan pada karakteristik ciri-ciri yang diberikan. *Output* dari algoritma ini adalah objek yang akan dikelompokkan ke dalam kelas yang mempunyai kesamaan ciri, misalnya sebuah objek masuk dalam kelas apel Amerika atau apel Malang maupun sebuah objek masuk dalam kelas daging sapi atau daging babi atau daging ayam. Jadi, dapat dikatakan bahwa klasifikasi merupakan pengelompokkan data (parameter hasil dari ekstraksi ciri) berdasarkan pada kesamaan ciri yang dimiliki oleh data tersebut. Algoritma klasifikasi yang biasa digunakan adalah *k-nearest neighbor*, *naive bayes*, jaringan saraf tiruan, *Support Vector Machine (SVM)*, *decision trees* dan lain-lain. Algoritma *clustering* yang biasa digunakan adalah *k-means clustering*, *fuzzy c-means clustering* dan lain-lain

Teknik pembelajaran pada algoritma pengenalan pola tergantung pada tipe label *output* yang digunakan. Jika label *output* data pembelajaran diketahui, maka

disebut sebagai pembelajaran *supervised* dan pembelajaran seperti ini digunakan untuk keperluan klasifikasi. Jika label *output* data pembelajaran tidak diketahui, maka disebut sebagai pembelajaran *unsupervised* dan pembelajaran seperti ini digunakan untuk keperluan *clustering* (Andono et al., 2017).

2.6 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang fitur (*feature space*) berdimensi tinggi, dilatih dengan algoritma pembelajaran yang didasarkan pada teori optimasi dengan mengimplementasikan *learning* bias yang berasal dari teori pembelajaran statistik. Tujuannya adalah untuk menemukan *hyperplane* optimal yang memaksimalkan margin pada masing-masing kelas. Teori yang mendasari SVM sendiri sudah berkembang sejak 1960-an, tetapi baru diperkenalkan oleh Vapnik, Boser dan Guyon pada tahun 1992 dan sejak itu SVM berkembang dengan pesat. SVM adalah salah satu teknik yang relatif baru dibandingkan dengan teknik lain, tetapi memiliki performansi yang lebih baik di berbagai bidang aplikasi seperti *bioinformatics*, pengenalan tulisan tangan, klasifikasi teks dan lain sebagainya (Hafizh, 2020).

2.6.1 Konsep Support Vector Machine (SVM)

Ide dasar SVM adalah memaksimalkan batas *hyperplane* (*maximal margin hyperplane*). Konsep klasifikasi dengan SVM dapat dijelaskan secara sederhana sebagai usaha untuk mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas data pada ruang *input*. *Hyperplane* (batas keputusan) pemisah terbaik antara kedua kelas dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. Margin adalah jarak antara *hyperplane* tersebut dengan data terdekat dari masing-masing kelas. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pelatihan pada SVM (Hafizh, 2020).

2.6.2 Support Vector Machine (SVM) Multiclass

SVM pada dasarnya didesain untuk klasifikasi biner (dua kelas). Namun, penelitian lebih lanjut untuk mengembangkan SVM sehingga bisa mengklasifikasi

data yang memiliki lebih dari dua kelas, terus dilakukan. Ada dua pilihan untuk mengimplementasikan *multiclass* SVM yaitu dengan menggabungkan beberapa SVM biner atau menggabungkan semua data yang terdiri dari beberapa kelas ke dalam sebuah bentuk permasalahan optimasi. Namun, pada pendekatan yang kedua permasalahan optimasi yang harus diselesaikan jauh lebih rumit. Pada penelitian ini, pendekatan *multiclass* SVM yang penulis gunakan adalah metode klasifikasi “*one-against-all*”. Pada metode ini, dibangun k buah model SVM biner, dengan k adalah banyak kelas. Setiap model klasifikasi ke-i dilatih dengan menggunakan keseluruhan data, untuk mencari solusi permasalahan. SVM melakukan klasifikasi dua kelas antara satu kelas dengan kelas-kelas lain yang dipandang sebagai satu kelas. Kelas untuk suatu sampel data dapat langsung ditentukan dengan metode ini. Ketika sampel data tidak dimasukkan ke dalam kelompok yang berisi sekumpulan kelas, tetapi ke dalam suatu kelas yang spesifik, maka kelas tersebut adalah kelas dari sampel data yang bersangkutan (Hafizh, 2020).

Tabel 2. 3 Klasifikasi SVM Biner dengan Metode One-against-all
(Sumber : Hafidz, 2020)

$y_i = 1$	$y_i = -1$	Hipotesis Kernel
Kelas 1	Bukan Kelas 1	$f_1(x) = (w^1)x + b^1$
Kelas 2	Bukan Kelas 2	$f_2(x) = (w^2)x + b^2$
Kelas 3	Bukan Kelas 3	$f_3(x) = (w^3)x + b^3$
Kelas 4	Bukan Kelas 4	$f_4(x) = (w^4)x + b^4$

2.6.3 Karakteristik *Support Vector Machine* (SVM)

Karakteristik klasifikasi SVM dapat diringkas menjadi seperti berikut :

1. SVM bisa dikatakan sebagai teknik klasifikasi yang semi-eager learner karena selain memerlukan proses pelatihan, SVM juga menyimpan sebagian kecil data latih untuk digunakan kembali pada saat proses prediksi.
2. SVM selalu memberikan model yang sama dan solusi dengan margin maksimal.

3. Proses pelatihan yang dilakukan oleh SVM tidak sebanyak ANN, tetapi sering kali memberikan kinerja yang lebih baik daripada ANN.
4. Dalam penggunaannya SVM hanya menentukan fungsi kernel yang harus digunakan (untuk kasus data yang distribusi kelasnya tidak dapat dipisahkan secara linear).
5. SVM membutuhkan komputasi pelatihan dan prediksi yang rumit karena dimensi data yang digunakan dalam proses pelatihan dan prediksi lebih besar daripada dimensi yang sesungguhnya.
6. Untuk set data berjumlah besar, SVM membutuhkan memori yang sangat besar untuk alokasi matriks kernel yang digunakan.


Penggunaan matriks kernel mempunyai keuntungan lain, yaitu kinerja set data dengan dimensi besar tetapi jumlah datanya sedikit akan lebih cepat karena ukuran data pada dimensi baru berkurang banyak (Hafizh, 2020).



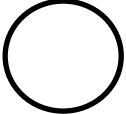
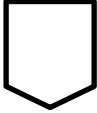


2.7 Bagan Alir (*Flowchart*)

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian (Murdock, 2018).

Flowchart adalah rangkaian gambar yang menunjukkan aliran proses data/algorithm yang dibuat. Suatu *flowchart* terdiri atas gabungan dari beberapa simbol yang dihubungkan dengan anak panah (Fayeldi & Murniasih, 2016).

Tabel 2. 4 Simbol-simbol *flowchart*
(Sumber : Fayeldi & Murniasih, 2016)

Simbol	Arti	Keterangan
	<i>Input/Output</i>	Simbol <i>input/output</i> digunakan untuk mewakili data <i>input/output</i> .

	Proses	Simbol proses digunakan untuk mewakili suatu proses.
	Garis Alir	Simbol garis alir (<i>flow lines</i>) digunakan untuk menunjukkan arus dari proses.
	Penghubung	Simbol penghubung (<i>connector symbol</i>) digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang masih sama.
	Penghubung	Simbol penghubung (<i>connector symbol</i>) digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman lainnya.
	Keputusan	Simbol keputusan (<i>decision symbol</i>) digunakan untuk suatu penyeleksian kondisi di dalam program.
	Proses Terdefinisi	Simbol proses terdefinisi (<i>predifined process symbol</i>) digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain.

2.8 Pseudocode

Pseudocode atau kode semu dapat di artikan sebagai deskripsi dari algoritma pemrograman yang dituliskan secara sederhana dibandingkan dengan sintaksis bahasa pemrograman. Tujuannya, agar lebih mudah dibaca dan dipahami manusia. *Pseudocode* bukanlah sebuah bahasa pemrograman, karena sebuah bahasa pemrograman harus memiliki aturan dalam penulisan kodenya. Sementara *pseudocode* sendiri tidak memiliki aturan yang spesifik atau baku dalam penulisannya, karena itu *pseudocode* tidak dikategorikan sebagai bahasa pemrograman. (Setiawan, 2021)

Pseudocode adalah notasi algoritmik yang lebih menyerupai bahasa pemrograman tingkat tinggi, misalnya Pascal. Agar suatu algoritma dalam bentuk *Pseudocode* mudah dipahami, maka algoritma tersebut perlu ditulis dalam struktur tertentu. Struktur dari suatu algoritma adalah sebagai berikut (Fayeldi & Murniasih, 2016) :

A. Header

Header memuat nama dan informasi mengenai algoritma yang ditulis.

B. Deklarasi

Deklarasi memuat definisi berbagai variabel dan tipe data yang digunakan dalam algoritma tersebut.

C. Deskripsi

Deskripsi memuat langkah-langkah penyelesaian masalah dengan menggunakan algoritma tersebut.

Pseudocode Algoritma Klasifikasi Jenis Motif Songket dalam penelitian ini:

Program Ekstraksi_Fitur_Jenis_Motif_Songket

Deklarasi

citra_motif_batik : BLOP;

Grayscale : *float*;

GLCM, fitur, ekstraksi_ciri : Int;

Deskripsi

Read (citra_motif_batik);

Grayscale $\leftarrow 0.299R + 0.587G + 0.144 B$

GLCM $\leftarrow 0^\circ, 45^\circ, 90^\circ, 135^\circ$

Write (*Contras, Homogenety, Energy, Entropy*)

Write (ekstraksi ciri)

Program Klasifikasi_Jenis_Motif_Songket_dengan_SVM

Deklarasi

Fitur_glcm, normalisasi_data, nilai_output_y, nilai_weight_w, nilai_bias_b,

nilai_hyperplane: int;

Deskripsi

Read (input data fitur);

normalisasi_data;

penentuan nilai_output_y $\leftarrow y := 1$ untuk kelas pertama, $y := -1$ untuk kelas berikutnya

penentuan nilai_weight_(w) $\leftarrow y_i (w \cdot x_i + b) \geq 1$

penentuan nilai_bias_(b) $\leftarrow -20w_1 + 379w_2 + 4w_3 + 2b = 2$

nilai_hyperplane $\leftarrow f(x) = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + b = 0$

Write (klasifikasi motif songket)

2.9 Matlab

Matlab merupakan sebuah singkatan dari *Matrix Laboratory*. Matlab dikenalkan untuk pertama kali oleh University of Stanford. Matlab awalnya hanya digunakan untuk keperluan analisis numerik, aljabar linear, dan matriks. Akan tetapi, saat ini kemampuan dan fitur yang dimiliki oleh Matlab sudah jauh lebih lengkap dengan ditambahkan beragam toolbox (Fayeldi & Murniasih, 2016).

2.9.1 Manfaat dari Matlab

Beberapa manfaat yang didapatkan dari Matlab antara lain sebagai berikut :

- Perhitungan metematika.
- Komputasi numerik.
- Simulasi dan pemodelan.
- Visualisasi dan analisis data.
- Pembuatan grafik untuk keperluan sains dan teknik.
- Pengembangan aplikasi berbasis *General User Interface*.

2.9.2 Penulisan Instruksi pada Matlab

Beberapa hal penting yang harus di perhatikan dalam penulisan instruksi pada Matlab, yaitu sebagai berikut :

- Variabel bersifat *case sensitive*, artinya Matlab akan membedakan adanya huruf besar dan kecil dalam penamaan variabel tersebut.
- Panjang nama variabel maksimum 31 karakter.

- c. Penamaan variabel harus selalu diawali dengan huruf, tidak boleh dengan bilangan, ataupun simbol (Fayeldi & Murniasih, 2016).

2.9.3 Kelengkapan pada sistem Matlab

Fitur-fitur Matlab sudah banyak dikembangkan, dan lebih kita kenal dengannya *toolbox*. Sangat penting bagi seorang pengguna Matlab, *toolbox* mana yang mendukung untuk *learn* dan *apply* teknologi yang sedang dipelajarinya. *Toolbox* ini merupakan kumpulan dari fungsi-fungsi Matlab (*M-files*) yang telah dikembangkan ke suatu lingkungan kerja Matlab untuk memecahkan masalah dalam kelas *particular*. Area-area yang sudah bisa dipecahkan dengan *toolbox* saat ini meliputi pengolahan sinyal, sistem kontrol, *neural networks*, *fuzzy logic*, *wavelets*, dan lain-lain (Mara Doli Nst et al., 2017)

Sebagai sebuah sistem, Matlab tersusun dari 5 bagian utama :

a. *Development Environment*.

Merupakan sekumpulan perangkat dan fasilitas yang membantu anda untuk menggunakan fungsi-fungsi dan file-file Matlab. Beberapa perangkat ini merupakan sebuah *graphical user interfaces* (GUI). Termasuk didalamnya adalah *Matlab desktop* dan *Command Window*, *command history*, sebuah editor dan *debugger*, sebuah editor dan *debugger* dan untuk melihat help, workspace, files dan *search path*.

b. *Matlab Mathematical Function Library*.

Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti: sum, sin, cos, dan complex arith-metic, sampai dengan fungsi-fungsi yang lebih kompleks seperti *matrix inverse*, *matrix eigenvalues*, *Bessel functions*, dan *fast Fourier transforms*.

c. *Matlab Language*.

Merupakan suatu *high-level matrix/array language* dengan *control flow statements*, *functions*, *data structures*, *input/output*, dan fitur-fitur *object-oriented programming*. Ini memungkinkan bagi kita untuk melakukan kedua hal baik "pemrograman dalam lingkup sederhana " untuk mendapatkan

hasil yang cepat, dan "pemrograman dalam lingkup yang lebih besar" untuk memper-oleh hasil-hasil dan aplikasi yang kompleks.

d. *Graphics*.

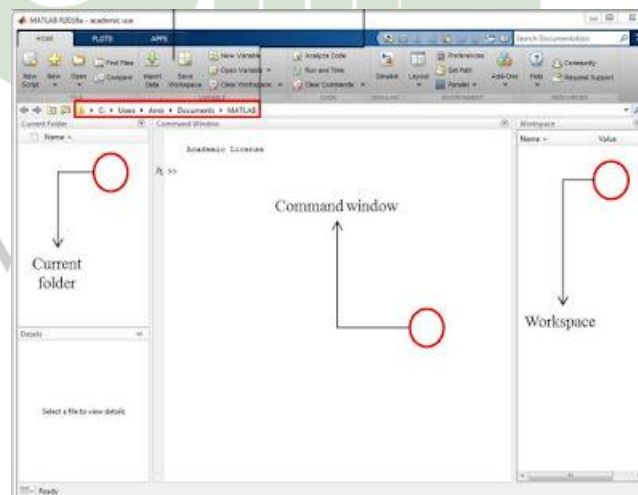
Matlab memiliki fasilitas untuk menampilkan vektor dan matrik sebagai suatu grafik. Di dalamnya melibatkan *high-level functions* (fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan data tiga dimensi, *image pro-cessing*, *animation*, dan *graphics presentation*. Ini juga melibatkan *fungsi level rendah* yang memungkinkan bagi anda untuk membiasakan diri untuk memunculkan grafik mulai dari bentuk yang sederhana sampai dengan tingkatan *graphical user interfaces* (GUI) pada aplikasi Matlab anda

e. *Matlab Application Program Interface (API)*

Merupakan suatu *library* yang memungkinkan program yang telah anda tulis dalam bahasa C dan Fortran mampu berinteraksi dengan Matlab. Ini melibatkan fasilitas untuk pemanggilan routines dari Matlab (*dynamic linking*), pemanggilan Matlab sebagai sebuah *computational engine*, dan untuk membaca dan menuliskan MAT-files.

2.9.4 Desktop Dasar Matlab

Saat menjalankan Matlab, akan tampak tampilan dasar berikut :



Gambar 2. 2 Tampilan Layar Utama Matlab

(Sumber : Mara Doli Nst et al., 2017)

Tampilan desktop terdiri atas beberapa panel berikut :

- a. *Current Folder* : Panel ini berfungsi untuk mengakses file. *Window* ini menampilkan isi dari direktori kerja saat menggunkan Matlab. Direktori dapat diganti sesuai dengan tempat direktori kerja yang diinginkan. *Default* dari alamat berada dalam folder *works* tempat program files Matlab berada.
- b. *Command Window* : Panel ini sebagai tempat untuk menuliskan perintah, ditandai dengan simbol `>>`.
- c. *Workspace* : Panel ini untuk mengeksplorasi data yang akan dibuat atau impor dari file. *Workspace* berfungsi untuk menampilkan seluruh variabel-variabel yang sedang aktif pada saat pemakaian Matlab. Apabila variabel berupa data matriks berukuran besar maka user dapat melihat isi dari seluruh data dengan melakukan double klik pada variabel tersebut. Matlab secara otomatis akan menampilkan *window* “*array editor*” yang berisikan data pada setiap variabel yang dipilih *user*.

2.10 Penelitian Terdahulu

Beberapa penelitian mengenai klasifikasi telah banyak dilakukan oleh para peneliti, berbagai metode baik ekstraksi fitur telah diujicobakan. Antara lain sebagai berikut:

1. Johan Wahyudi & Ihdahubbi Maulida yang melakukan penelitian “Pengenalan Pola Citra Kain Tradisional Menggunakan GLCM dan KNN”, data yang digunakan dalam peneletian ini sebanyak 50 motif kain sasirangan, sistem pengenalan pola sasirangan masih belum dapat mengklasifikasikan data sesuai dengan kelas gambar kain. Hal tersebut disebabkan dari pengambilan data yang tidak konsisten sehingga mempengaruhi kualitas gambar yang akhirnya menghasilkan akurasi sebesar 63%.
2. Mhd Furqan, dkk, yang melakukan penelitian “Klasifikasi Daun Bugenvil Menggunakan *Gray Level Co-Occurrence Matrix* dan *K-Nearest Neighbor*” dengan menggunakan data 60 citra daun yaitu 45 citra data latih dan 15 citra data uji dan memiliki nilai akurasi 87%.
3. Neneng, dkk, yang melakukan penelitian “*Support Vector Machine* untuk Klasifikasi Citra Jenis Daging Berdasarkan Tekstur Menggunakan Ekstraksi

Ciri *Gray Level Co-Occurrence Matrices* (GLCM)” dengan data yang digunakan yaitu citra daging kambing, daging kerbau, daging kuda dan sapi dengan cara pengambilan 20 cm, 30 cm, 40 cm dan menghasilkan akurasi sebesar 87.5% .

4. Danar Putra Pamungkas yang melakukan penelitian “Ekstraksi Citra menggunakan Metode GLCM dan KNN untuk Identifikasi Jenis Anggrek (*Orchidaceae*)”, data citra yang digunakan sejumlah 90 gambar, dimana 30 gambar *Phalaenopsis*, 30 gambar *Dendrobium*, dan 30 gambar bunga lain untuk pengujiannya dan menghasilkan tingkat keberhasilan mencapai rata-rata 77%.
5. Adrian Budi Prawira, dkk melakukan penelitian “Penerapan Metode *Gray Level Co-Occurrence Matrix* dan Algoritma *Support Vector Machine* pada Klasifikasi Tanaman Bidara Berdasarkan Tekstur Daun” dengan data 50 citra Bidara Arab dan 50 citra Bidara Cina dan menghasilkan rata-rata akurasi 80%.



UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN



UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN