IMPLEMENTASI PENGOLAHAN CITRA UNTUK PENGENALAN NOMINAL MATA UANG BERDASARKAN CIRI WARNA DAN TEKSTUR MENGGUNAKAN KLASIFIKASI SUPPORT VECTOR MACHINE (SVM)

SKRIPSI

Diajukan oleh :

NOVITA PURWANTI
NIM : 2008096015

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG

TAHUN 2023

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Novita Purwanti

NIM : 2008096015

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

Implementasi Pengolahan Citra Untuk Pengenalan Nominal Mata Uang Berdasarkan Ciri Warna Dan Tekstur Menggunakan Klasifikasi *Support Vector Machine (SVM)*

Secara keseluruhan adalah penelitian/karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 20 Juni 2024

Novita Purwanti

NIM. 2008096015

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Alamat: Jl. Prof. Dr. Hamka Km 1, Ngaliyan Semarang Telp.024-76433366 Semarang 50185
Email: fst@walisongo.ac.id, Web: http://fst.walisongo.ac.id

PENGESAHAN

Naskah skripsi berikut ini:

Judul : Implementasi Pengolahan Citra untuk Pengenalan Nominal Mata Uang berdasarkan Ciri Warna Dan Tekstur menggunakan Klasifikasi *Support Vector Machine* (SVM)

Nama : **Novita Purwanti**
NIM : 2008096015
Jurusan : Teknologi Informasi

Telah diujikan dalam Sidang Tugas Akhir oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo Semarang dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam Teknologi Informasi.

Semarang, 20 Juni 2024

DEWAN PENGUJI

Penguji I

Dr. Khotibul Umam, ST., M.Kom
NIP. 197908272011011007

Penguji II

Dr. Masy Ari Ulinuha, M.T
NIP. 198703172019031007

Penguji III

Nur Cahyo Hendro Wibowo., S.T., M.Kom
NIP. 197312222006041001

Penguji IV

Siti Nur'aini, M.Kom
NIP. 198401312018012001

KEMENTERIAN AGAMA FAKULTAS SAINS DAN TEKNOLOGI UIN WALISONGO SEMARANG REPUBLIK INDONESIA

Pembimbing I,

Dr. Masy Ari Ulinuha, M.T
NIP. 198703172019031007

Pembimbing II,

Adzhal Arwani Mahfudh, M.Kom
NIP. 199107032019031006

NOTA DINAS

NOTA DINAS

Semarang, 20 Juni 2024

Yth. Ketua Program Studi
Teknologi Informasi Fakultas
Sains dan Teknologi UIN
Walisongo Semarang

Assalamu'alaikum. wr. wb.
Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul : Implementasi Pengolahan Citra Untuk Pengenalan Nominal Mata Uang Berdasarkan Ciri Warna dan Tekstur Menggunakan Klasifikasi Support Vector Machine

Nama : **Novita Purwanti**
NIM : 2008096015
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosah.

Wassalamu'alaikum. wr. wb.

Pembimbing I,

Dr. Masy Ari Ulinuha, S.T M.T.
NIP : 19870317 201903 1 007

NOTA DINAS

NOTA DINAS

Semarang, 20 Juni 2024

Yth. Ketua Program Studi
Teknologi Informasi Fakultas
Sains dan Teknologi UIN
Walisongo Semarang

Assalamu'alaikum. wr. wb.
Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul	: Implementasi Pengolahan Citra Untuk Pengenalan Nominal Mata Uang Berdasarkan Ciri Warna dan Tekstur Menggunakan Klasifikasi Support Vector Machine
Nama	: **Novita Purwanti**
NIM	: 2008096015
Jurusan	: Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosah.

Wassalamu'alaikum. wr. wb.

Pembimbing II,

Adzhal Arwani Mahfudh,
S.Kom., M.Kom
NIP : 19910703 201903 1 006

LEMBAR PERSEMBAHAN

Puji sykur penulis panjatkan kepada Allah SWT yang telah memberikan kesehatan, rahmat dan hidayah, sehingga penulis masih diberikan kesempatan untuk menelesaikan skripsi ini sebagai salah satu syarat untuk mendapatkan gelar sarjana. Meskipun kepenulisan ini jauh dari kata sempirna, namun penulis bangga telah mencapai pada titik ini yang akhirnya skripsi ini bisa selseai diwaktu yang tepat. Karya kecil ini penulis persembahkan untuk:

1. Bapak Joko Purwanto dan Ibu Suheti selaku orangtua dari penulis.
2. Hananto Wibowo selaku kakak penulis
3. Seluruh dosen jurusan Teknologi Informasi
4. Sahabat dan teman-teman seperjuangan khususnya jurusan Teknologi Informasi Angkatan 2020
5. Almamater Universitas Islam Negeri Walisongo Semarang

MOTTO

"Karena sesungguhnya sesudah kesulitan itu ada kemudahan,
sesungguhnya sesudah kesulitan itu ada kemudahan"
Q.S Al Insyirah: 5-6

ABSTRAK

Uang merupakan alat pembayaran yang sah. Mata uang yang digunakan di Indonesia adalah rupiah. Terdapat beberapa jenis uang salah satunya adalah uang kertas. Nominal uang kertas yang digunakan di Indonesia adalah 1000, 2000, 5000, 10000, 20000, 50000, dan 100000. Pada mesin otomatis yang berhubungan dengan pembayaran menggunakan uang membutuhkan sebuah sistem yang harus mengenali dan mengklasifikasi nominal mata uang. Pada penelitian ini melakukan implementasi citra digital untuk pengklasifikasian nominal mata uang menggunakan metode support vector machine berdasarkan ciri warna (HSV) dan tekstur (LBP). Kemudian dalam pengkasifikasian nominal uang kertas terdapat 7 kelas yang didasarkan pada nominal uang kertas yang digunakan di Indonesia. Pengujian yang dilakukan di penelitian ini menghasilkan performa yang baik dengan melakukan perbandingan data training dan data testing sebesar 80:20 menghasilkan nilai akurasi sebesar 95%, precission sebesar 96%, recall sebesar 95%, dan f1 Score sebesar 95%.

Kata kunci : Uang, HSV, LBP, Support Vector Machine

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT yang memberikan rahmat dan hidayah-Nya kepada penulis serta telah memberikan kemudahan dan kelancaran sehingga penulis dapat menyelesaikan Tugas Akhir sebagai syarat kelulusan dalam menempuh Pendidikan di Progam Studi Teknologi Informasi Fakultas Sains dan Teknologi UIN Walisongo Semarang telah dapat penulis selesaikan dengan judul "Implementasi Pengolahan Citra Digital untuk Pengenalan Nominal Mata Uang berdasarkan Ciri Warna dan Tekstur menggunakan Support Vector Machine". Pada kesempatan ini penulis ingin menyampaikan sebuah ucapan terima kasih yang tak terhingga kepada beberapa pihak yang telah banyak memberikan bantuan, arahan dan juga bimbingan sehingga terselesaikannya Tugas Akhir ini. Penyusunan laporan tugas akhir ini tidak terlepas dari bantuan beberapa pihak, oleh karena itu penulis hendak mengucapkan terima kasih kepada:

1. Kedua orang tua penulis yaitu Bapak Joko Purwanto dan Ibu Suheti yang selalu memberikan dukungan dan tak lupa selalu mendoakan penulis sehingga terlesesaikannya masa pendidikan S1 dan tugas akhir ini.

2. Ketua Jurusan Prodi Teknologi Informasi UIN Walisongo Semarang, Bapak Dr. Khotibul Umam, ST., M.Kom
3. Dosen Pembimbing I sekaligus Dosen Wali Bapak Dr. Masy Ari Ulinuha, ST., M.T. yang selalu memberikan arahan dan bimbingan kepada penulis.
4. Dosen Pembimbing II Bapak Adzal Arwani Mahfudh, S,Kom., M.Kom yang selalu memberikan arahan dan bimbingan juga kepada penulis.
5. Seluruh dosen Teknologi Informasi, staf, karyawan dan dosen di lingkungan UIN Walisongo Semarang yang telah memberikan ilmu pengetahuan yang tidak ternilai selama menempuh pendidikan.
6. Semua pihak yang tidak dapat disebutkan satu per satu yang terlibat dalam penyusunan tugas akhir ini.

Akhir kata semoga segala kebaikan dan ketulusan yang diberikan mendapatkan balasan yang setimpal oleh Allah SWT. Semoga skripsi ini bisa bermanfaat bagi para pembaca dan bisa dijadikan bahan rujukan untuk melakukan penelitian selanjutnya.

Semarang, 20 Juni 2024

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengolahan citra digital merupakan salah satu pemanfaatan teknologi komputer yang digunakan untuk menyelesaikan masalah mengenai pengolahan gambar atau citra sehingga mudah diproses. Pengolahan citra juga merupakan bidang ilmu yang mempelajari tentang bagaimana suatu citra itu dibentuk, diolah, dan dianalisis sehingga menghasilkan informasi yang dapat dipahami oleh manusia, sedangkan Histogram citra adalah representasi grafik yang menyatakan distribusi nilai-nilai warna atau intensitas piksel-piksel di dalam citra. Pengolahan citra digunakan untuk mengenali suatu objek citra, memperbaiki kualitas citra, kompresi, ataupun untuk mengenali ciri dari suatu objek yang ingin diketahui. Citra merupakan salah satu bentuk informasi yang diperlukan manusia selain teks, suara dan video. Informasi yang terkandung dalam sebuah citra dapat diinterpretasikan berbeda-beda oleh manusia satu dengan yang lain. Pengolahan citra yang dimaksud adalah gambar diam ataupun gambar bergerak. (Rifqo et al., 2022)

Seiring dengan berkembangnya teknologi saat ini banyak sekali permasalah yang dapat diselesaikan menggunakan bantuan teknologi yang ada. Dan memberikan bantuan di berbagai aspek kehidupan manusia seperti halnya dalam ayat Al-Quran yaitu Surat Al-A'la ayat 8 yaitu :

وَنُيَسِّرُكَ لِلْيُسْرٰىۚ ﴿٨﴾

Arti : "Dan kami akan memberi kamu taufik ke jalan yang mudah". (Q.S Al- A'la ayat 8)

Dengan kata lain Allah SWT telah memberi manusia berbagai jalan dalam mencapai kemudahan baik dalam perintah agama maupun dunia. Salah satunya adalah kemudahan yang ada yaitu dengan perkembangan teknologi yaitu kecerdasan buatan *Artificial Intelligence* yang telah diterapkan dan membantu manusia di kehidupan sehari-hari. *Artificial Intelligence* merupakan cabang ilmu komputer yang berkaitan dengan menciptakan mesin pintar yang dapat melakukan pekerjaan layaknya manusia.

Menurut John McCarthy (1956), Kecerdasan Buatan merupakan suatu sistem komputer yang terbentuk untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan

perilaku manusia. Kecerdasaan buatan salah satunya adalah pengenalan nominal mata uang dengan memanfaatkan pengolahan citra digital. Salah satu manfaat adalah dapat diterapkan pada mesin penjual minuman otomatis, ATM yang dapat menerima setor tunai untuk tabungan, mesin penukar otomatis dan mesin otomatis lainnya. Tentu banyak kemudahan yang didapat dengan menggunakan sistem ini, diantaranya tidak perlu mengantri lama untuk mendapatkan sekaleng softdrink ataupun mengantri lama di teller untuk mendapatkan pelayanan di bank, dan berbagai kemudahan lain. (Birowo, 2020)

Menurut Solikin & Suseno Uang adalah suatu benda yang pada dasarnya dapat berfungsi sebagai: (1) alat tukar (*medium of exchange*), (2) alat penyimpan nilai (*store of value*), (3) satuan hitung (*unit of account*), dan (4) ukuran pembayaran yang tertunda (*standard for deffered payment*). (Samudra et al., 2020). Di indonesia sendiri uang yang digunakan adalah rupiah. Uang kertas rupiah merupakan uang kertas yang dicetak dan diedarkan oleh pemerintah Indonesia sebagai alat tukar yang sah.

Pengenalan nominal mata uang dilakukan dengan serangkaian pengolahan citra (*image processing*).

Pengolahan Citra bertujuan memperbaiki kualitas citra agar mudah di interpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan. Termasuk ke dalam bidang ini juga adalah pemampatan citra (*image compression*). (Rinaldi Munir, 2004).

Untuk cara kerja mesin pengenalan nominal mata uang sendiri cara kerjanya adalah seperti otak manusia. Di mana sistem itu harus dapat mengenali dan mengklasifikasikan nominal uang secara cepat dan tepat seperti yang dilakukan oleh manusia. Sistem tersebut akan digunakan pada mesin otomatis. Jika mesin menerima salah satu pecahan uang kertas, mesin itu dapat langsung mengenali nominal dari uang tersebut. Sehingga sistem ini dapat dikembangkan dan digunakan sebagai alat bantu penjualan yang melibatkan mesin. Proses pengenalan uang kertas tersebut dilihat dari gambar permukaannya atau pengenalan citra (*image recogniton*). (Izah, 2018)

Berdasarkan pemaparan diatas penelitian kali ini bertujuan untuk merancang sistem pengenalan nominal mata uang berdasarkan ciri warna dan tekstur, dimana ciri warna yang digunakan adalah fitur HSV yaitu nilai *hue, saturation, dan value*. Kemudian untuk ekstraksi tekstur menggunakan fitur *local binary pattern* yaitu nilai *mean, entropy, variance, kurtosis dan skewnes*. Selanjutnya metode klasifikasi yang digunakan dalam pengolahan citra digital deteksi nominal mata uang menggunakan *Support Vector Machine* (SVM). Dengan metode yang digunakan diharapkan dapat memberikan hasil identifikasi nominal mata uang yang akurat.

1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas, maka dapat di indentifikasi masalah sebagai berikut:

a. Pengenalan nominal mata uang bisa dilakukan dengan serangkaian pengolahan citra (*image processing*) dan proses klasifikasi.
b. Pengenalan nominal mata uang biasanya dilakukan dengan pengamatan visual oleh manusia namun ketika ingin membuat sebuah mesin yang berhubungan dengan uang seperti atm, mesin minuman otomatis dan lain sebagainya diperlukan sistem yang dapat

membantu mengidentifikasi nominal mata uang rupiah yang dimasukan kedalam mesin.

1.3 Rumusan Masalah

berdasarkan latar belakang penelitian diatas, maka dapat dirumuskan masalah adalah bagaimana merancang sistem pengenalan nominal mata uang kertas rupiah berdasarkan ciri warna dan tekstur menggunakan metode *support vector machine* (SVM)?

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, berikut merupakan tujuan dari penelitian adalah merancang sistem pengenalan nominal mata uang kertas rupiah berdasarkan ciri warna dan tekstur menggunakan metode *support vector machine* (SVM).

1.5 Batasan Masalah

Dalam penelitian ini menggunakan batasan masalah yang digunakan dalam proses penelitian agar dapat dilakukan secara jelas dan memiliki cakupan yang tidak meluas. Batasan masalah pada penelitian ini sebagai berikut:

a. Dalam penelitian ini menggunakan dataset yang diambil dari kaggle dengan jumlah 700 data yang memiliki format .png. dimana data dipisahkan 80% untuk training dan 20% untuk testing.

b. Penelitian ini untuk merancang pengenalan nominal mata uang kertas rupiah menggunakan metode *support vector machine* (SVM) yang berdasarkan ciri warna dan tekstur.
c. Simulasi penelitian ini menggunakan alat bantu Matlab 2017b
d. Simulasi percobaan pengenalan nominal mata uang menggunakan data dari dataset.

1.6 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat secara teoritis dan praktis seperti berikut ini:

a. Secara teoritis
    1. Memberikan kontribusi bagi perkembangan pengolahan citra digital di bidang perekonomian khususnya identifikasi nominal uang kertas
    2. Penelitian ini membantu mengembangkan penggunaan metode *Support Vector Machine* (SVM) dalam pengolahan gambar, khususnya untuk mengenali nominal mata uang. Hasil penelitian ini bisa menjadi acuan bagi penelitian lain yang ingin menggunakan atau mengembangkan metode SVM dalam aplikasi serupa.

b. Secara praktis

Membangun sistem yang dapat dikembangakan dan digunakan sebagai alat bantu penjualan yang melibatkan mesin. Contohnya diimplementasikan dalam perangkat kasir otomatis dan mesin penjual otomatis (*vending machine*) untuk mendeteksi dan memverifikasi uang kertas secara cepat dan akurat. Ini meningkatkan efisiensi dan mengurangi kesalahan manusia dalam transaksi tunai.

BAB II
LANDASAN PUSTAKA

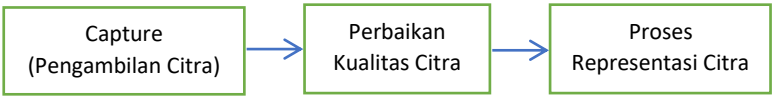
2.1 Landasan Teori

2.1.1 Pengolahan Citra

Pengolahan citra digital adalah ilmu yang me mempelajari hal-hal berkaitan dengan perbaikan kualitas terhadap suatu gambar (meningkatkan kontras, perubahan warna, restorasi citra), transformasi gambar (translasi, rotasi transformasi, skala, geometrik), melakukan pemilihan citra ciri (*feature images*) yang optimal untuk tujuan analisis, melakukan penyimpanan data yang sebelumnya dilakukan reduksi dan kompresi, transmisi data, dan waktu proses data sehingga dapat menganalisa suatu citra (Rifqo et al., 2022).

Secara umum pengolahan citra memiliki definisi pengolahan gambar berdimensi dua melalui komputer digital (Ulfah & Nurdin, 2023). Citra diolah pada tiap pixel (x,y) untuk menghasilkan citra baru yang sesuai dengan kebutuhan. Citra adalah bagian dari komponen multimedia yang memegang peranan penting dalam menyajikan informasi visual. Foto adalah contoh gambar berdimensi dua yang bisa diolah dengan mudah. Setiap foto dalam bentuk citra digital (misalnya berasal dari kamera digital) dapat diolah melalui perangkat-lunak tertentu. Proses pengolahan

citra secara diagram proses dimulai dari pengambilan citra, perbaikan kualitas citra, sampai dengan pernyataan representatif citra dicitrakan dengan gambar 2.1 berikut ini.



Gambar 2. 1 *Proses Pengolahan Citra*

2.1.2 Uang

Uang adalah segala sesuatu yang siap sedia dan pada umumnya diterima dalam pembayaran pembelian barang-barang, jasa-jasa dan untuk membayar utang. Rupiah adalah mata uang yang dikeluarkan oleh Negara Kesatuan Republik Indonesia, UU No. 23 tahun 1999 melindungi penggunaan rupiah. Bank Indonesia menjadi satu-satunya lembaga yang berwenang melakukan pengeluaran, pengedaran dan atau pencabutan rupiah. Uang merupakan alat yang paling penting dalam melakukan transaksi yang digunakan oleh manusia. Setiap uang memiliki nilai nominal yang berguna untuk menentukan nilai dari suatu barang dan jasa yang diperjual belikan (Izah, 2018).

Terdapat beberapa jenis-jenis uang salah satunya adalah uang kertas. Uang kertas adalah uang yang terbuat dari kertas dengan gambar dan cap tertentu dan merupakan alat pembayaran yang sah. Berdasarkan UU No. 23 tahun

1999 tentang Bank Indonesia, yang dimaksud dengan uang kertas adalah uang dalam bentuk lembaran yang terbuat dari bahan kertas atau bahan lainnya (Wahjudi, 2018). Secara normal, nilai nominal uang kertas sudah tertera di empat sisi uang kertas yang ditulis dengan angka, begitupun untuk mendeteksi keaslian uang kertas, pemerintah sering mensosialisasikan cara 3D (Dilihat, Diraba, Diterawang).

Uang kertas rupiah tahun emisi 2016 dan 2022 yang akan digunakan dalam penelitian ini, dengan pecahan Rp 1.000,00, Rp 2.000,00, Rp 5.000,00, Rp 10.000,00, Rp 20.000,00, Rp 50.000,00, dan Rp 100.000,00 pada sisi depan (gambar pahlawan nasional Indonesia) serta sisi belakang (tarian tradisional Indonesia). Berikut adalah gambar uang kertas rupiah tersebut.

Gambar 2. 2 Gambar Uang Kertas Rupiah

2.1.3 Ekstraksi Fitur

Ekstraksi fitur merupakan proses pengindeksan suatu database citra dengan isinya. Secara matematik, setiap ekstraksi fitur merupakan *encode* dari *vector* n dimensi yang disebut dengan *vector* fitur. Komponen *vector* fitur dihitung dengan proses citra dan teknik analisis serta digunakan untuk membandingkan citra yang satu dengan citra yang lain. Ekstraksi fitur diklasifikasikan kedalam 3 jenis yaitu *low level, middle level dan high level*. Fitur *low level* merupakan ekstraksi fitur berdasarkan isi visual seperti warna dan tekstur, fitur *middle level* merupakan ekstraksi fitur berdasrkan wilayah citra yang ditentukan dengan segmentasi, sedangkan fitur *high level* merupakan ekstraksi fitur berdasarkan informasi semantik yang terkandung dalam citra (J.R Parker, 2011).

Pemisahan ciri *(feature extraction)* merupakan bagian fundamental dari analisis citra. Fitur adalah karakteristik unik dari suatu objek. Karakteristik fitur yang baik sebisa mungkin memenuhi persyaratan berikut (Putra, 2010):

a. Dapat membedakan suatu objek dengan yang lainnya (*discrimination*)
b. Memperhatikan kompleksitas komputasi dalam memperoleh fitur. Kompleksitas komputasi yang tinggi

tentu akan menjadi beban tersendiri dalam menemukan suatu fitur.

c. Tidak terikat (*independence*) dalam arti bersifat invariant terhadap berbagai transformasi (rotasi, penskalaan, penggeseran, dan lain sebagainya).
d. Jumlah sedikit, karena fitur yang jumlahnya sedikit akan dapat menghemat waktu komputasi dan ruang penyimpanan untuk proses selanjutnya (proses pemanfaatan fitur).

Proses Ekstraksi fitur yang digunakan sebagai ciri dalam pengenalan suatu objek juga dapat dilakukan dengan menganalis tekstur objek. Tekstur objek dapat direpresentasikan dengan menggunakan persamaan matematika, sehingga hasil dari analisa tekstur dapat diukur dan dibandingkan dengan objek lainnya dalam proses pengenalan. (Muntasa, 2010)

Fitur-fitur suatu objek mempunyai peran penting untuk berbagai aplikasi berikut (Kadir & Susanto, 2013):

1. Pencarian citra: Fitur dipakai untuk mencari objek-objek tertentu yang berada di dalam database.
2. Penyederhanaan dan hampiran bentuk: Bentuk objek dapat dinyatakan dengan representasi yang lebih ringkas.

3. Pengenalan dan klasifikasi: Sejumlah fitur dipakai untuk menentukan jenis objek.

Ekstraksi Fitur ini diperlukan untuk mengetahui suatu citra. Teknik ekstraksi fitur yang digunakan adalah dengan mengekstrak citra warna RGB (*Red-Green-Blue*). Pengambilan fitur warna RGB menggunakan aplikasi *Design Graphic*. Uang yang telah dipindai di-import ke dalam aplikasi dan diseleksi pada uang kertas tersebut dan secara otomatis aplikasi mengeluarkan fitur warna RGB pada uang kertas. Setiap uang kertas dilakukan dua tahapan ekstraksi fitur, karena pada uang kertas terdapat dua sisi yang berbeda yaitu dari sisi muka dan dari sisi belakang (Pratama et al., 2020).

Ekstraksi Ciri Statistik Orde Pertama :

1. Mean (μ)

Fitur pertama yang dihitung secara statistis adalah rerata intensitas (*mean*). Komponen fitur ini dihitung berdasar persamaan:

$$\mu = \sum_{n=0}^{N} f_n \, p\,(f_n) \qquad (2.1)$$

Dimana: μ : nilai mean

f_n : nilai intensitas keabuan

$p\,(f_n)$: nilai histogram

Dalam hal ini, i adalah aras keabuan pada citra f dan p(i) menyatakan probabilitas kemunculan i dan L menyatakan nilai aras keabuan tertinggi. Rumus di atas akan menghasilkan rerata kecerahan objek.

2. *Entropy* (H)

Entropy mengindikasikan kompleksitas citra. Perhitungannya sebagai berikut:

$$H = -\sum_{n=0}^{N} p\ (f_n).^2\ \log\ p\ (f_n) \qquad (2.2)$$

Dimana: H : nilai entropy

Semakin tinggi nilai *entropy*, semakin kompleks citra tersebut. Perlu diketahui, *entropy* dan energi berkecenderungan berkebalikan. *Entropy* juga merepresentasikan jumlah informasi yang terkandung di dalam sebaran data.

3. *Variance* (σ^2)

Variance yaitu parameter yang menunjukan variasi elemen dari suatu citra.

$$\sigma^2 = \sum_{n=0}^{N} (f_n - \mu)^2\ p\ (f_n) \qquad (2.3)$$

Dimana: μ : nilai mean

4. *Kurtosis* (α^4)

Kurtosis merupakan ukuran keseragaman.

$$\alpha^4 = \frac{1}{\sigma^4} \sum_{n=0}^{N} (f_n - \mu)^4 \, p\,(f_n) - 3 \qquad (2.4)$$

5. *Skewness* (α^3)

Fitur *skewness* merupakan ukuran ketidaksimetrisan terhadap rerata intensitas. Definisinya:

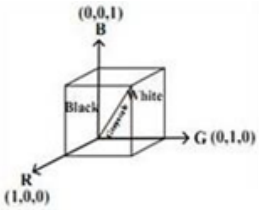
$$\alpha^3 = \frac{1}{\sigma^3} \sum_{n=0}^{N} (f_n - \mu)^3 \, p\,(f_n) \qquad (2.5)$$

Skewness sering disebut sebagai momen orde tiga ternormalisasi. Nilai negatif menyatakan bahwa distribusi kecerahan condong ke kiri terhadap rerata dan nilai positif menyatakan bahwa distribusi kecerahan condong ke kanan terhadap rerata. Dalam praktik, nilai *skewness* dibagi dengan (L-1)2 supaya ternormalisasi. (Sari & Sari, 2023)

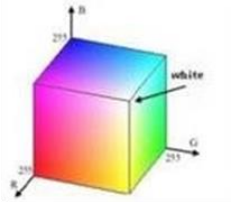
2.1.4 Ruang Warna RGB

Menurut Abdul Kadir dan Adhi Susanto (2013) ruang RGB biasanya diterapkan pada monitor CRT dan kebanyakan sistem grafika komputer. Ruang warna ini menggunakan tiga komponen dasar yaitu merah (R), hijau (G), biru (B). Setiap *pixel* dibentuk oleh tiga komponen tersebut. Model RGB biasanya disajikan dalam bentuk kubus tiga dimensi, dengan warna merah, hijau, dan biru berada di pojok sumbu. Warna hitam berada di titik asal dan warna putih berada diujung

kubus yang bersebrangan. Kubus warna secara nyata dengan resolusi 24 *bit*. Perlu diketahui dengan menggunakan 24 *bit*. Jumlah warna mencapai 16.777.216.



Gambar 2. 3 Skema Ruang Warna RGB dalam Bentuk Kubus
Sumber: Teori dan Aplikasi Pengolahan Citra



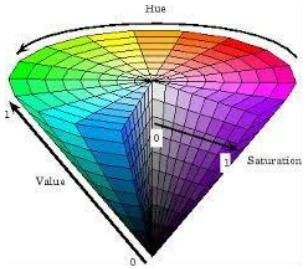
Gambar 2. 4 Kubus Warna dengan 24 bit
Sumber: Teori dan Aplikasi Pengolahan Citra (2013)

RGB biasanya digunakan karena kemudahan dalam perancangan hardware tapi sebenarnya tidak ideal untuk beberapa aplikasi. Mengingat warna merah, hijau, dan biru sesungguhnya terkolerasi sangat erat. Sangat sulit untuk beberapa algoritma pemrosesan citra (Crane, 1997). Sebagai contoh, kebutuhan untuk memperoleh warna alamiah seperti merah dengan menggunakan RGB menjadi sangat kompleks

mengingat komponen R dapat berpasangan dengan G dan B, dengan nilai berapa saja. Hal ini menjadi mudah jika menggunakan ruang warna HLS ataupun HSV.

2.1.5 Warna HSV

Menurut Kadir dan Susanto (2013) HSV merupakan contoh ruang warna yang merepresentasikan warna seperti yang dilihat oleh mata manusia. H berasal dari kata "*hue*", S berasal dari kata "*saturation*", dan V berasal dari kata "*value*". Ruang warna HSV terkadang dinamakan HSB, dengan B berasal dari kata "*brightness*".



Gambar 2. 5 Ruang Warna HSV
Sumber: Teori dan Aplikasi Pengolahan Citra (2013)

Model HSV, yang pertama kali diperkenalkan A. R. Smith pada tahun 1978, ditunjukkan di Gambar 2.5. Untuk mendapatkan nilai H, S, V, berdasarkan R, G, dan B terdapat beberapa cara.

Cara tersederhana adalah sebagai berikut:

$$r = \frac{R}{(R+G+B)}, g = \frac{G}{(R+G+B)}, b = \frac{B}{(R+G+B)} \quad (2.6)$$

$$V = \max(r, g, b) \quad (2.7)$$

$$S = \begin{cases} 0, & jika\ V = 0 \\ 1 - \dfrac{\min(r,g,b)}{v}, & V > 0 \end{cases} \quad (2.8)$$

$$H = \begin{cases} 0, jika\ S = 0 \\ \dfrac{60 * (g-b)}{S * V}, jika\ V = r \\ 60 * \left[2 + \dfrac{b-r}{S*V}\right], jika\ V = g \\ 60 * \left[4 + \dfrac{r-g}{S*V}\right], jika\ V = b \end{cases} \quad (2.9)$$

$$H = H + 360\ jika\ H < 0$$

2.1.6 Tekstur

Menurut Kadir dan Susanto (2013), Selain melibatkan fitur bentuk, tekstur banyak digunakan sebagai fitur untuk temu kembali citra. Hal ini disebabkan beberapa objek mempunyai pola-pola tertentu, yang bagi manusia mudah untuk dibedakan. Oleh karena itu, diharapkan komputer juga dapat mengenali sifat- sifat seperti itu. Dalam praktik, tekstur digunakan untuk berbagai kepentingan.

Umumnya, aplikasi tekstur dapat dibagi menjadi dua kategori. Pertama adalah untuk kepentingan segmentasi. Pada proses ini, tekstur dipakai untuk melakukan pemisahan

antara satu objek dengan objek yang lain. Kedua adalah untuk klasifikasi tekstur, yang menggunakan fitur-fitur tekstur untuk mengklasifikasi objek. Beberapa contoh aplikasi tekstur disajikan di bawah ini (Tuceryan dan Jain, 1998).

a. Inspeksi secara otomatis pada industri tekstil, pengecatan mobil, pemakaian karpet.
b. Analisis citra medis. Misalnya, tekstur digunakan untuk klasifikasi penyakit paru-paru, diagnosis leukemia, dan pembedaan tipe-tipe sel darah putih.
c. Analisis penginderaan jarak-jauh. Misalnya, tekstur dipakai untuk kepentingan klasifikasi area tanah.

Kulkarni (1994) mendefinisikan tekstur sebagai hubungan mutual antara nilai intensitas piksel-piksel yang bertetangga yang berulang di suatu area yang lebih luas daripada jarak hubungan tersebut. Namun, penjelasan seperti itu pun masih menyisakan ketidakmudahan untuk mengenali pengulangan yang terjadi pada citra. Ada suatu pengulangan yang terkadang sulit dijabarkan, tetapi mudah ditangkap oleh mata, seperti yang terdapat pada Gambar 2.6 (a) dan Gambar 2.6 (b). Hal ini berbeda dengan Gambar 2.6 (c). Citra yang disebut terakhir mempunyai sifat pengulangan yang mudah dilihat. Namun, pada ketiga gambar tersebut, jelas bahwa ada suatu tekstur yang terkandung dalam setiap citra. Tekstur pada

Gambar 2.6 (c), dari sisi keteraturan pola, adalah yang paling mudah untuk dikenali. (Kadir dan Susanto, 2013)

Gambar 2. 6 Berbagai citra yang memiliki sifat tekstur yang berbeda-beda (a) halus (b) kasar (c) teratur
Sumber: Teori dan Aplikasi Pengolahan Citra

Menurut Kadir dan Susanto (2013), berdasarkan keteraturan pengulangan pola dalam objek, tekstur dapat dikategorikan ke dalam dua bentuk: 1) tekstur teratur dan 2) tekstur tidak teratur. Tekstur buatan manusia berkecenderungan masuk dalam kategori tekstur teratur, sedangkan tekstur alamiah bersifat tidak teratur. Berdasarkan tingkat kekasaran objek, tekstur dibedakan menjadi dua: mikrotekstur dan makrotekstur (Acharya dan Ray, 2005). Apabila ukuran elemen yang menyusun pengulangan pola berukuran besar, tekstur dikatakan kasar atau bertekstur makro. Tekstur seperti itu dinamakan sebagai makrostruktur.

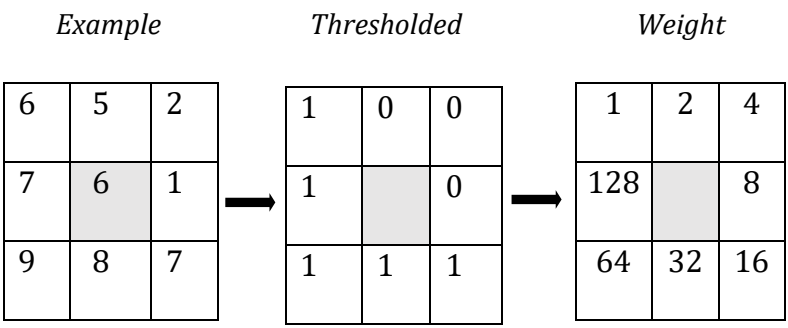
Sebaliknya, mikrostruktur mempunyai sifat elemen-elemen yang menyusun pengulangan pola berukuran kecil.

Berdasarkan perspektif matematis, tekstur dapat dibedakan ke dalam spektrum stokastis dan spektrum regular. Tekstur stokastis (atau kadang disebut tekstur statistis) adalah tekstur yang mempunyai bentuk mirip dengan derau. Tekstur regular (atau terkadang sebagai tekstur struktural) adalah tekstur yang tersusun atas pola-pola periodis. Dalam hal ini, warna/intensitas serta bentuk elemen tekstur diulang dengan interval yang sama.

2.1.7 Local Binary Pattern

Menurut (Pietikainen, Matti et al. 2011) dalam bukunya menjelaskan bahwa *Local Binary Pattern* adalah operator tekstur sederhana namun sangat efisien yang memberi label pada piksel gambar dengan mengukur lingkungan masing-masing piksel dan menganggap hasilnya sebagai bilangan biner. Metode LBP dapat dilihat sebagai pendekatan pemersatu terhadap model analisis tekstur dan struktural yang berbeda secara tradisional. Mungkin properti terpenting operator LBP dalam aplikasi dunia nyata adalah invariannya karena perubahan tingkat abu-abu monotonik yang disebabkan, misal, oleh variasi iluminasi. Hal lain yang sama pentingnya adalah kesederhanaan komputasi, yang memungkinkan untuk menganalisis gambar dalam menantang pengaturan real time.

Operator pola biner lokal asli, yang diperkenalkan oleh Ojala dkk, didasarkan pada asumsi bahwa tekstur memiliki dua aspek pelengkap lokal, dan merupakan kekuatannya. Operator bekerja di lingkungan 3 x 3, menggunakan nilai pusat sebagai ambang batas. Kode LBP diproduksi dengan mengalikan nilai ambang batas dengan bobot yang diberikan oleh piksel yang sesuai, dan menjumlahkan hasilnya. Karena lingkungannya terdiri dari 8 piksel, total 2^8 = 256 label yang berbeda dapat diperoleh tergantung pada nilai abu-abu relatif dari pusat dan piksel di lingkungan sekitar. Ukuran kontras (C) diperoleh dengan mengurangkan rata-rata tingkat abu-abu di bawah piksel tengah dari tingkat abu-abu di atas (atau sama dengan) piksel tengah. Jika semua delapan tetangga dari piksel tengah memiliki nilai yang sama (0 atau 1), nilai kontras diatur ke nol. Distribusi memiliki nilai yang sama (0 atau 1), nilai kontras diatur ke nol. Distribusi kode LBP atau distribusi dua dimensi LBP dan kontras lokal (LBP / C), digunakan sebagai fitur dalam klasifikasi atau segmentasi. Lihat gambar 2.7 untuk ilustrasi operator LBP dasar.



Gambar 2. 7 Ilustrasi Operator LBP Dasar
Sumber: *Computer Vision Using Local Binary Patterns* (2011)

Karena daya diskriminatifnya merupakan kesederhanaan komputasi, operator tekstur LBP telah menjadi pendekatan yang sangat populer dalam berbagai aplikasi. Keberhasilan besar LBP dalam berbagai masalah analisis tekstur telah menunjukkan bhwa bank filter dengan area pendukung besar tidak diperlukan untuk kinerja tinggi dalam klasifikasi tekstur, namun operator yang didefinisikan untuk lingkungan kecil seperti LBP biasanya memadai. Berikut ini merupakan rumus LBP:

$$LBP_{P,R} = \sum_{p=0}^{p-1} S(I_{P,R} - I_C)2^{P-1-P} \quad (2.10)$$

Dimana nilai S didefinisikan sebagai berikut:

$$S(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \quad (2.11)$$

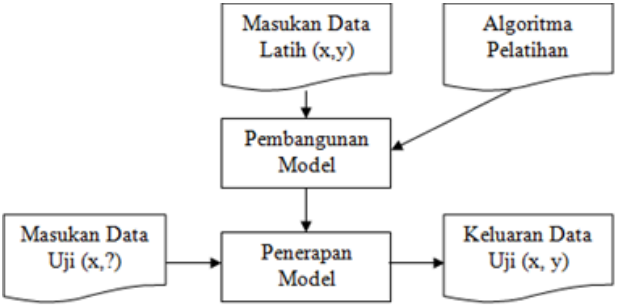
P adalah jumlah banyak tetangga, R adalah radius antara titik pusat dan titik tetangga, LBP_{PR} adalah nilai desimal hasil konversi nilai biner. I_C adalah nilai intensitas piksel pusat, I_{PR} adalah nilai intensitas piksel tetangga ke-p (p=0,1, ..., p-1) dengan radius R. Sedangkan s(x) adalah fungsi *thresholding.*

Eksploitasi informasi tekstur secara tepat dapat meningkatkan kinerja dan keandalan banyak tugas dan sistem visi komputer, membantu membuat teknologi menjadi sangat kuat dan mudah digunakan dalam aplikasi dunia nyata. (Pietikainen, Matti et al., 2011)

2.1.8 Klasifikasi

Menurut Prasetyo (2012), klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu (1) pembangunan model sebagai prototipe untuk disimpan sebagai memori dan (2) penggunaan model tersebut untuk melakukan pengenalan/ klasifikasi/ prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang sudah disimpannya.

Kerangka kerja seperti yang ditunjukkan pada gambar 2.8 meliputi dua langkah proses, yaitu induksi dan deduksi. Induksi merupakan langkah untuk membangun model klasifikasi dari data latih yang diberikan, disebut juga proses pelatihan, sedangkan deduksi merupakan langkah untuk menerapkan model tersebut pada data uji sehingga kelas yang sesungguhnya dari data uji dapat diketahui, disebut juga proses prediksi.



Gambar 2. 8 Proses Pekerjaan Klasifikasi

Sumber: Data Mining Konsep dan Aplikasi Menggunakan Matlab (2012)

Berdasarkan cara pelatihan, algoritma-algoritma klasifikasi dapat dibagi menjadi dua macam, yaitu *eager learner* dan *lazy learner*. Algoritma- algoritma yang termasuk dalam kategori *eager learner* didesain untuk melakukan pembacaan/ pelatihan/ pembelajaran pada data latih agar dapat memetakan dengan benar setiap vektor masukan ke label kelas keluarannya sehingga di akhir proses pelatihan,

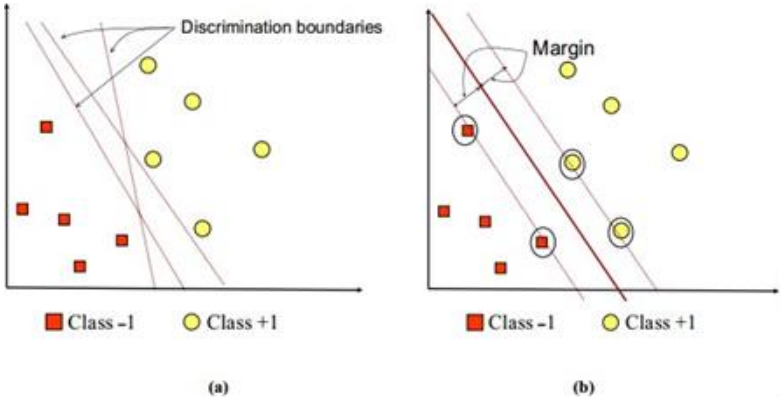
model sudah dapat memetakan semua vektor data uji ke label kelas keluarannya dengan benar. Selanjutnya, setelah proses pelatihan tersebut selesai, model (biasanya berupa bobot atau sejumlah nilai kuantitas tertentu) disimpan sebagai memori, sedangkan semua data latihnya dibuang. Proses prediksi dilakukan dengan model yang tersimpan, tidak melibatkan data latih sama sekali. Cara ini mengakibatkan proses prediksi berjalan dengan cepat, tetapi harus dibayar dengan proses pelatihan yang lama. Algoritma-algoritma klasifikasi yang masuk kategori ini, diantaranya, adalah *Artificial Neural Network* (ANN), *Support Vector Machine* (SVM), *Decision Tree,* Bayesian, dan sebagainya. (Prasetyo, 2012)

Sementara algoritma-algoritma yang masuk dalam kategori *lazy learner* hanya sedikit melakukan pelatihan (atau tidak sama sekali), hanya menyimpan sebagian atau seluruh data latih, kemudian menggunakannya dalam proses prediksi. Hal ini mengakibatkan proses prediksi menjadi lama karena model harus membaca kembali semua data latihnya agar dapat memberikan keluaran label kelas dengan benar pada data uji yang diberikan. Kelebihan algoritma seperti ini adalah proses pelatihan yang berjalan dengan cepat. Algoritma-algoritma klasifikasi yang masuk kategori ini, di antaranya, adalah *K-Nearest Neighbor* (K-NN), *Fuzzy K-Nearest Neighbor* (FK-NN), Regresi Linear, dan sebagainya. (Prasetyo, 2012)

2.1.9 Support Vector Machine (SVM)

Menurut Widodo, et al. (2013), Support Vector Machine (SVM) merupakan metode klasifikasi jenis terpandu (supervised) karena ketika proses pelatihan, diperlukan target pembelajaran tertentu. SVM muncul pertama kali pada tahun 1992 oleh Vladimir Vapnik bersama rekannya Bernhard Boser dan Isabelle Guyon.

Ide dasar SVM adalah memaksimalkan batas *hyperplane (maximal margin hyperplane)* seperti yang diilustrasikan pada gambar 2.9



Gambar 2. 9 (a) Decision Boundary yang Mungkin (b) Decision Boundary dengan Margin Maksimal

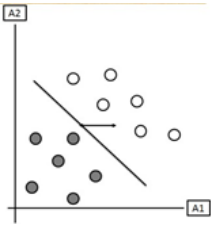
Sumber: Penerapan Data Miniengan Matlab (2013)

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. Data yang paling dekat ini disebut *support vector*. Garis solid pada gambar 2.9 (b) sebelah kanan menunjukkan *hyperplane* yang terbaik yaitu yang terletak tepat pada tengah-tengah kedua kelas, sedangkan data lingkaran dan bujur sangkar yang dilewati garis batas margin (garis putus-putus) adalah *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pelatihan pada SVM. (Prasetyo, 2012)

2.1.9.1 SVM Linier

Kasus data yang terpisah secara linear adalah kasus yang termudah. Misalnya kita memiliki data yang terdiri dari kelas orang yang membeli komputer dan yang tidak membeli komputer (X_i, Y_i), (X_2, Y_2), ..., $(X_{|D|}, Y_{|D|})$. Data tersebut kita beri notasi X dan kelasnya adalah y dimana yi hanya memiliki dua kemungkinan yakni +1 atau -1. (Widodo, et al., 2013)



Gambar 2. 10 Data yang Terpisah Secara Linear
Sumber: Penerapan Data Mining dengan Matlab (2013)

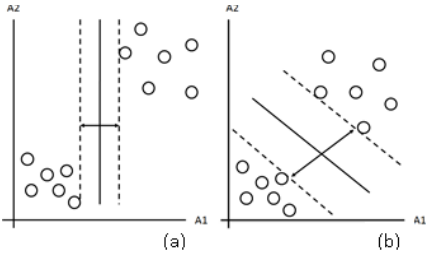
SVM memecahkan masalah klasifikasi dengan mencari *hyperplane* marjinal maksimum dimana ada jumlah tak terbatas *hyperplanes* yang harus dicari mana yang terbaik. Secara intuitif, *hyperplane* dengan margin yang lebih besar lebih akurat dalam mengklasifikasikan data dibanding margin yang lebih kecil. Inilah sebabnya mengapa (selama pembelajaran), SVM mencari *hyperplane* dengan *margin* terbesar, dikenal dengan istilah *Maximum Marginal Hyperplane* (MMH). Untuk definisi *margin*, kita dapat mengatakan bahwa jarak terpendek dari *hyperplane* ke satu sisi *margin* adalah sama dengan jarak terpendek dari *hyperplane* yang ke sisi lain dari *margin*, dimana "sisi" dari *margin* sejajar dengan *hyperplane* tersebut. (Widodo, *et al.,* 2013)

Sebuah *hyperplane* dapat ditulis sebagai

$$W . X + b = 0 \qquad (2.12)$$

Di mana W adalah vektor bobot, yaitu, W = {w1, w2, ..., wn}, n adalah jumlah atribut, dan b adalah skalar yang sering disebut sebagai bias. Jika b sebagai bobot tambahan, w0, kita dapat menulis ulang *hyperplane* pemisah sebagai

$$w_0 + w_1x_1 + w_2x_2 = 0 \qquad (2.13)$$



Gambar 2. 11 (a) Margin Kecil (b) Margin Besar
Sumber: Penerapan Data Mining dengan Matlab (2013)

Dengan demikian, setiap titik yang terletak di atas *hyperplane* pemisah memenuhi:

$$w_0 + w_1x_1 + w_2x_2 > 0 \qquad (2.14)$$

Demikian pula, setiap titik yang terletak di bawah *hyperplane* pemisah memenuhi:

$$w_0 + w_1x_1 + w_2x_2 < 0 \qquad (2.15)$$

Bobot dapat disesuaikan sehingga *hyperplanes* dapat mendefinisikan "sisi" dari margin yang ditulis sebagai:

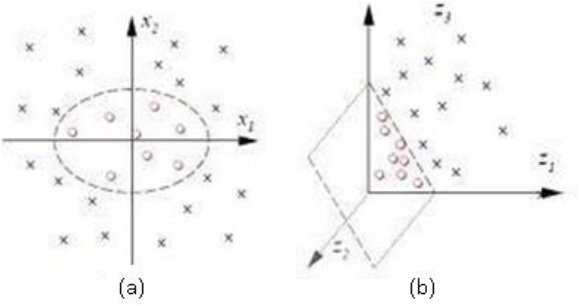
H_1: $w_0 + w_1x_1 + w_2x_2 \geq 1$ untuk yi = +1 (2.16)

H_1: $w_0 + w_1x_1 + w_2x_2 \leq 1$ untuk yi = -1 (2.17)

Artinya, setiap tupel yang jatuh pada atau di atas H_1 milik kelas +1, dan setiap tuple yang jatuh pada atau di bawah H_2 milik kelas -1. (Widodo, *et al.,* 2013)

2.1.9.2 SVM Nonlinear

Menurut (Prasetyo, 2012) untuk data yang distribusi kelasnya tidak linear biasanya digunakan pendekatan kernel pada fitur data awal set data. Kernel dapat didefinisikan sebagai suatu fungsi yang memetakan fitur data dari dimensi awal (rendah) ke fitur lain yang berdimensi lebih tinggi (bahkan jauh lebih tinggi). Pendekatan ini berbeda dengan metode klasifikasi pada umumnya yang justru mengurangi dimensi awal untuk menyederhanakan proses komputasi dan memberikan akurasi prediksi yang lebih baik.



Gambar 2. 12 Dimensi Data (a) Dalam Fitur Dimensi Rendah (b) Dalam Fitur Dimensi Tinggi

Sumber: Data Mining: Konsep dan Aplikasi Menggunakan Matlab (2012)

Algoritma pemetaan kernel ditunjukkan sebagai berikut:

$$\Phi : D^q \rightarrow D^r$$

$$x \rightarrow \Phi(x)$$

Φ merupakan fungsi kernel yang digunakan untuk pemetaan, *D* merupakan data latih, *q* merupakan set fitur dalam satu data yang lama, dan *r* merupakan set fitur yang baru sebagai hasil pemetaan untuk setiap data latih. Sementara *x* merupakan data latih, dimana $x1, x2, ..., xn \in D^q$ merupakan fitur-fitur yang akan dipetakan ke fitur berdimensi tinggi *r*, jadi untuk set data yang digunakan sebagai pelatihan dengan algoritma yang ada dari dimensi fitur yang lama *D* ke dimensi baru *r*. (Prasetyo, 2012)

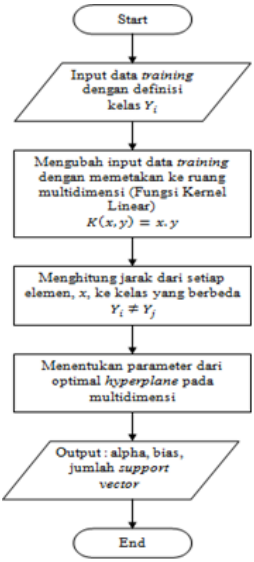
Untuk pilihan fungsi kernel yang banyak digunakan dalam aplikasi, dapat dilihat pada tabel 2.1. *x* dan *y* adalah pasangan dua data dari semua bagian data latih. Parameter σ, $c, d > 0$, merupakan konstanta. (Prasetyo, 2012)

Tabel 2. 1 fungsi kernel

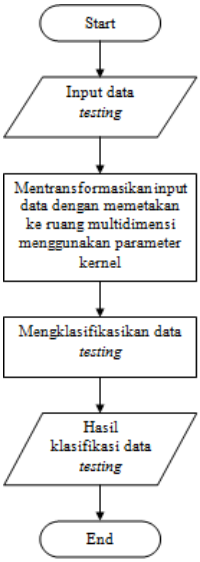
Nama Kernel	Definisi Fungsi
Linear	$K(x,y) = x.y$
Polinomial	$K(x,y) = (x.y + c)^d$
Gaussian RBF	$K(x,y) = \exp\left(\frac{-\|x-y\|^2}{2.\sigma^2}\right)$
Sigmoid (tangen hiperbolik)	$K(x,y) = \tanh(\sigma(x.y) + c)$
Invers Multikuadrik	$K(x,y) = \frac{1}{\sqrt{\|x-y\|^2 + c^2}}$

Sumber: Data Mining: Konsep dan Aplikasi Menggunakan Matlab (2012)

Dalam implementasinya, proses klasifikasi dibagi menjadi 2 tahap yaitu *training* dan *testing.*



Gambar 2. 13 Diagram Alir Training SVM



Gambar 2. 14 Diagram Alir Testing SVM

Sumber: Data Mining: Konsep dan Aplikasi Menggunakan Matlab (2012)

2.1.9.3 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.

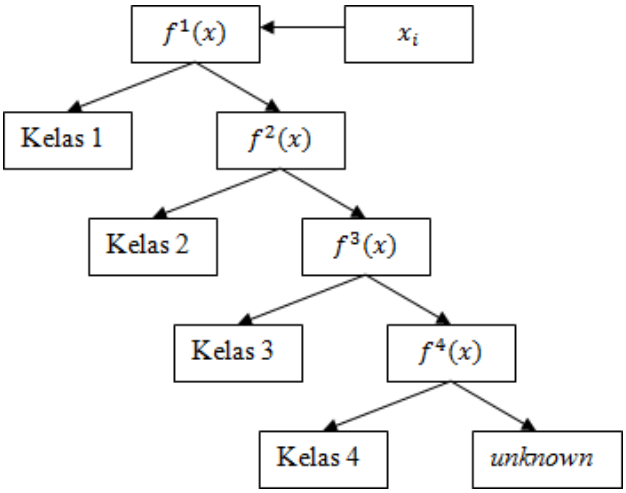
Dalam rangka untuk membuat SVM sebagai multiclass problem, berikut ini teknik yang diadopsi (Pushpalatha, K.N, et al., 2012) :

```
1.  k = 1;
2.  do
3.  Create an initial SVM problem as A x B where A=k,
    b={k+1,k+2,...n} Suppose the test vector is T
4.  Class=classify(SVM(A,B),T)
5.  Where classify is SVM classification problem.
6.  If(class==A)
7.  Break;
8.  Else K++;
9.  While(k>=(n-1));
10. Actual class=Class
```

Tabel 2. 2 Klasifikasi SVM Biner dengan Metode One-against-all

$y_i = 1$	$y_i = -1$	Hipotesis
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	$f^4(x) = (w^4)x + b^4$

Sumber : *Iris recognition system with Frequency Domain Features Optimized with PCA and SVM Classifier.* (2012)



Gambar 2. 15 Contoh Klasifikasi dengan Metode One-against-all

Sumber: *Iris recognition system with Frequency Domain Features Optimized with PCA and SVM Classifier.* (2012)

2.1.9.4 Karakteristik SVM

Menurut Prasetyo (2012), karakteristik klasifikasi SVM dapat diringkas menjadi seperti berikut:

1. SVM merupakan teknik klasifikasi yang semi-eager leaner karena selain memerlukan proses pelatihan, SVM juga menyimpan sebagian kecil data latih untuk digunakan kembali pada saat proses prediksi.
2. Untuk parameter yang sama yang digunakan dalam klasifikasi, SVM memberikan model klasifikasi yang solusinya adalah global optima, tidak seperti *Artificial Neural Network* (ANN) yang solusinya sering masuk pada wilayah lokal optima. Hal ini berarti SVM selalu memberikan model yang sama dan solusi dengan margin maksimal, sedangkan ANN memberikan model dengan nilai yang berbeda dengan margin yang tidak selalu sama.
3. Proses pelatihan yang dilakukan oleh SVM tidak sebanyak ANN, tetapi sering kali memberikan kinerja yang lebih baik daripada ANN.
4. Tidak membutuhkan pemilihan parameter-parameter seperti pada ANN. 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.
7. 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.

2.1.10 Matlab

MATLAB merupakan bahasa pemrograman yang hadir dengan fungsi dan karakteristik yang berbeda dengan bahasa pemrograman lain yang sudah ada lebih dahulu seperti DELPHI, BASIC, maupun C++. Matlab merupakan bahasa pemrograman level tinggi yang dikhususkan untuk kebutuhan komputasi, teknis, visualisasi dan pemrograman seperti komputasi matematik, analisis data, pengembangan algoritma, simulasi dan pemodelan dan grafik-grafik perhitungan. Matlab hadir dengan membawa warna yang berbeda. Hal ini karena matlab membawa keistimewaan dalam fungsi-fungsi matematika, fisika, statistik, dan

visualisasi. Matlab dikembangkan oleh mathworks, yang pada awalnya dibuat untuk memberikan kemudahan mengakses data matrik pada proyek LINPACK dan EISPASCK. Pengembangan MATLAB untuk pertama kali dilakukan oleh Universitas Nee Mexico dan Universitas Stanford pada tahun 1970. Saat ini MATLAB memiliki ratusan fungsi yang dapat digunakan sebagai problem solver mulai dari sederhana sampai masalah-masalah yang kompleks dari berbagai disiplin ilmu. (Siregar dan Sani, 2017)

MATLAB adalah perangkat lunak (*software*) interaktif yang powerful, komperhensif, dan *user friendly* yang telah menjadi standar industri untuk menyelesaikan berbagai permasalahan dalam berbagai bidang, misalnya sains, rekayasa, ekonomi, pendidikan, dan sebagainya. Salah satu keuntungan menggunakan MATLAB adalah dari segi kemudahan membuat program, dibandingkan dengan pemrograman menggunakan bahasa lain seperti Basic, C++, Pascal, atau FORTRAN. Keunggulan lain dari MATLAB dibandingkan bahasa komputasi lain adalah tersedianya fasilitas grafik, fungsi *built-in*, dan *toolbox* yang sangat banyak. MATLAB bersifat extensible, yakni memungkinkan bagi seorang pengguna untuk menulis fungsi baru yang dapat ditambahkan pada *library* ketika fungsi-fungsi yang tersedia

(*built-in*) tidak dapat melakukan tugas tertentu yang dibutuhkan.

MATLAB adalah singkatan dari Matrix Laboratory, *software* ini dibuat oleh The Mathworks.inc dan telah memasuki versi 9.14 kekuatan MATLAB terletak pada:

1. Kemudahan manipulasi struktur matriks
2. Jumlah *routine-routine powerful* yang berlimpah yang terus berkembang
3. Kekuatan fasilitas grafik tiga dimensi yang sangat memadai
4. Sistem *scripting* yang memberikan keleluasaan bagi pengguna untuk mengembangkan dan memodifikasi *software* untuk kebutuhan sendiri
5. Kemampuan *interface* (misal dengan bahasa C, *word* dan *mathematics*)
6. Dilengkapi dengan *toolbox, simulink, stateflow* dan sebagainya, serta mulai melimpahnya *source code* di internet yang dibuat dalam MATLAB (contoh *toolbox,* misalnya: *signal processing, optimization, control system, image processing, statistic, neural network design,* dan sebagainya)

Semua itu merupakan perangkat yang *powerful* untuk menyelesaikan permasalahan sains dan teknik terutama untuk wilayah dimana komputasi numerik harus dilakukan.

2.2 Kajian Penelitian yang Relevan

Berikut ini beberapa penelitian yang telah dilakukan oleh para peneliti sebelumnya terkait dengan penelitian yang akan dibuat yaitu:

Tabel 2. 3 Penelitian yang Relevan

No	Peneliti	Judul	Hasil Penelitian	Kelebihan dan Kekurangan
1	Nurul Ilmi Lailatal Fajriyah (2020)	Klasifikasi citra kupu-kupu menggunakan ciri tekstur, ruang warna hsv dan backpropagation neutral network	penelitian untuk klasifikasi citra kupu-kupu menggunakan ciri tekstur, ruang warna HSV, dan backpropagation neural network menghasilkan rata-rata akurasi tertinggi yang diperoleh yaitu 63% dengan nilai K=10, learning rate 0,1 dan maksimum epoch 100 pada neuron hidden 19.	Kelebihan:menggunakan penerapan metode hsv, lbp, dan GLCM untuk ekstraksi fitur warna dan tekstur dengan Backpropagation Neural Network yang efektif untuk klasifikasi citra kupu-kupu. Kekurangan : Gabungan dari metode HSV, Local Binary Pattern (LBP), Gray Level Cooccurrence Matrix (GLCM) dan Backpropagation Neural Network dapat digunakan pada penelitian ini. Namun dengan nilai akurasi yang masih relatif rendah

2	M. Kukuh Prayogo (2018)	Pengenalan Citra Uang Kertas dengan Metode Local Binary Pattern	Pengenalan Citra Uang Kertas dengan Metode Local Binary Pattern menggunakan 6 nominal uang kertas rupiah masing-masing tiga lembar untuk setiap nominal menggunakan segmentasi 50, 128, 200 dan 800, didapatkan akurasi pembacaan sebesar 92,857% untuk 50 segmentasi, 97,61% untuk 128 segmentasi, 92,857% untuk 200 segmentasi dan 83,333% untuk 800 segmentasi.	Kelebihan : Pengenalan Citra Uang Kertas menggunakan Metode Local Binary Pattern dengan 6 nominal uang kertas rupiah masing-masing tiga lembar untuk setiap nominal, menggunakan segmentasi 50, 128, 200 dan 800, Kekurangan : Akurasi pengenalan uang kertas dapat dipengaruhi oleh iluminasi yang tidak merata dan jumlah segmentasi pada proses Local Binary Pattern (LBP). Pencahayaan yang lebih merata dan penambahan sampel data base dapat meningkatkan akuras, Jumlah sampel yang terbatas dalam data learning mempengaruhi keakuratan pengenalan.
3	Titi Sumarni (2016)	Analisis Citra untuk Identifikasi Nilai Mata Uang Logam Rupiah	Analisis Citra untuk Identifikasi Nilai Mata Uang Logam Rupiah menghasilkan akurasi fitur area	Kelebihan : penelitian menggunakan fitur area dan ruang warna HSV, menghasilkan akurasi yang tinggi yaitu diperoleh rata-rat diatas 50%

			dan ruang warna HSV dengan analisis diskriminan two group diperoleh rata-rata diatas 50%.	Kekurangan : Meskipun berhasil mencapai akurasi tinggi, penelitian ini memiliki beberapa kekurangan. pertama, penelitian terbatas pada mata uang logam dan tidak mencakup mata uang kertas. Kedua, pengujian sistem hanya dilakukan dalam lingkungan terkontrol dan tidak mempertimbangkan kondisi mata uang logam yang rusak atau usang.
4	Millenia Mudita Chandra, Yoannita (2023)	Klasifikasi Jenis Bunga Menggunakan Metode SVM Berdasarkan Citra Dengan Fitur HSV	Klasifikasi Jenis Bunga Menggunakan Metode SVM Berdasarkan Citra Dengan Fitur HSV menghasilkan akurasi keseluruhan sebesar 63.66%.	Kelebihan : Penelitian ini berhasil mengklasifikasikan jenis bunga matahari dan daisy dengan menggunakan ekstraksi fitur HSV dan metode SVM, mencapai akurasi yang cukup baik sebesar 63.36%. Kekurangan : masih perlu peningkatan untuk mencapai akurasi yang lebih tinggi.
5	Sylviana Murni, Didit Widiyanto, Catur	Klasifikasi Citra Penyakit Daun Kopi Arabika Menggunakan	Klasifikasi Citra Penyakit Daun Kopi Arabika Menggunakan Support Vector	Kelebihan : Penelitian ini menonjolkan penerapan metode Support Vector Machine (SVM) dengan seleksi fitur Information Gain

	Nugrahaeni Puspita Dewi	Support Vector Machine (SVM) Dengan Seleksi Fitur Information Gain.	Machine (SVM) Dengan Seleksi Fitur Information Gain menghasilkan h nilai akurasi sebesar 68,30%, presisi sebesar 55,77%, dan recall sebesar 57,85%.	untuk klasifikasi citra penyakit daun kopi Arabika, kemampuannya untuk mengidentifikasi fitur-fitur yang paling relevan dari data citra, yang mana meningkatkan akurasi klasifikasi hingga 68,30%. Kekurangan : masih perlu peningkatan untuk mencapai akurasi yang lebih tinggi.

Berdasarkan tabel di atas maka peneliti akan melakukan penelitian terkait dengan identifikasi nominal mata uang rupiah berdasarkan ciri warna dan tekstur menggunakan metode *support vector machine* (SVM). Yang membedakan penelitian ini dengan penelitian sebelumnya adalah disini data citra uang yang digunakan menggunakan emisi 2016 dan 2022. Kemudian ekstraksi fitur yang digunakan adalah ciri warna yaitu fitur HSV dan ciri teksur yaitu Fitur Local Binary Pattern.

BAB III
METODOLOGI PENELITIAN

3.1 Teknik pengumpuan Data

pada penelitian kali ini data yang digunakan adalah jenis data sekunder. Data sekunder adalah data yang diperoleh atau dikumpulkan oleh orang yang melakukan penelitian dari sumber-sumber yang telah ada. Data ini digunakan untuk mendukung informasi primer yang telah diperoleh yaitu dari bahan pustaka, literatur, penelitian terdahulu, buku, dan lain sebagainya.

Data Sekunder yang digunakan dalam penelitian ini didapatkan melalui metode Studi Literatur yang merupakan suatu metode untuk pengumpulan data yang didapatkan dari sejumlah buku, majalah, atau artikel yang memiliki keterikatan terhadap masalah dan tujuan penelitian yang dimiliki (Riyono et al., 2022b). Studi Literatur juga dapat didefiniskan sebagai kegiatan dalam pengumpulan data dengan membaca, mencatat serta mengolah bahan atau instrument penelitian.

Data sekunder yang digunakan di penelitian ini berupa citra uang kertas yang bersumber dari *kaggle.com* dengan mata uang Rupiah sebagai data utama. Uang kertas yang digunakan adalah 7 jenis uang kertas yang beremisi tahun 2016 dan 2022. Nilai nominal uang kertas yang digunakan adalah Rp. I000, Rp.2000, Rp.5000, Rp.10.000, Rp.20.000, Rp.50.000, Rp. 100.000. Masing-masing nominal uang kertas akan dijadikan kelas untuk proses klasifikasi.

Pengumpulan data uang kertas dilakukan dengan uang kertas yang berbeda-beda. Pada setiap nominal dikumpulkan masing-masing l00 uang kertas yang menghasilkan data berjumlah 700 uang kertas. Selain data uang dalam keadaan baik, terdapat juga data uang kertas yang memiliki cacat seperti data uang yang tercoret, kusam, lecek dan sobek. Berikut merupakan tabel yang mendeskripsikan data yang didapatkan dari *kaggle*.

Tabel 3. 1 Deskripsi data uang rupiah

Nominal	Emisi	Bagian	Kondisi				
			baik	lecek	tercoret	sobek	Kusam
1000	2016	Depan	5	5	5	5	5
		Belakang	4	6	5	5	5
1000	2022	Depan	5	5	5	5	5
		Belakang	5	5	5	5	5
2000	2016	Depan	5	5	5	5	5
		Belakang	4	5	5	5	5
2000	2022	Depan	7	5	5	3	5
		Belakang	5	5	5	5	5
5000	2016	Depan	5	5	5	5	5

		Belakang	5	5	5	5	5
5000	2022	Depan	5	5	5	5	5
		Belakang	5	5	5	5	5
10000	2016	Depan	3	8	5	5	4
		Belakang	3	8	5	5	4
10000	2022	Depan	5	5	5	5	5
		Belakang	5	5	5	5	5
20000	2016	Depan	5	5	5	5	5
		Belakang	5	5	5	5	5
20000	2022	Depan	5	5	5	5	5
		Belakang	5	5	5	5	5
50000	2016	Depan	3	7	5	5	5
		Belakang	3	7	5	5	5
50000	2022	Depan	5	5	5	5	5
		Belakang	5	5	5	5	5
100000	2016	Depan	5	5	5	5	5
		Belakang	5	5	5	5	5
100000	2022	Depan	5	5	5	5	5
		Belakang	5	5	5	5	5

Sumber: Kaggle.com

3.2 Kebutuhan Perangkat Penelitian

Pada pelaksanaan penelitian ini tentunya dibutuhkan perangkat yang menunjang keperluan penelitian. Kebutuhan perangkat penelitian dibagi menjadi dua yaitu perangkat keras dan perangkat lunak. Deskripsi Perangkat dan Spesifikasi dijabarkan pada Tabel 3.1 untuk kebutuhan perangkat keras dan Tabel 3.2 untuk kebutuhan perangkat lunak.

1. Kebutuhan perangkat keras

Tabel 3. 2 Kebutuhan Perangkat Keras

NO	Perangkat	Spesifikasi
1	Device	HP 14s-dk0005AU
2	Processor	AMD
3	Memori (RAM)	4 GB
4	Monitor	14 inch

2. Kebutuhan perangkat lunak

Tabel 3. 3 Kebutuhan perangkat lunak

NO	Perangkat	Spesifikasi
1	Sistem Operasi	Windows 10
2	Tools	Matlab R2017b
3	Bahasa Pemrograman	Matlab

3.3 Tahapan penelitian

Tahapan penelitian yang penulis lakukan dapat digambarkan dalam diagram blok berikut:

Gambar 3. 1 Diagram blok tahapan penelitian

Di bawah ini merupakan penjelasan dari alur proses identifikasi nominal mata uang:

1. Preprocessing

   Tahap Preprocessing citra yang dilakukan adalah normalisasi ukuran citra (*resize image*). Pada proses normalisasi ukuran citra *(resize image)* bertujuan untuk memperkecil dimensi ukuran pixel citra sehingga dimensi ukuran pixel semua citra akan sama. Ukuran pixel dalam penelitian ini adalah 400 x 900 pixel.

   Contoh pada kelas 1000 terdapat gambar uang 1000 yang memiliki ukuran pixel 533x246 kemudian akan diubah menjadi 400 x 900 pixel.

Gambar 3. 2 perubahan pixel menjadi 400x900

2. Ekstraksi Fitur Warna HSV

Proses ekstraksi fitur warna dapat dilakukan setelah proses normalisasi selesai dengan cara mengubah warna RGB ke HSV. Hasil ekstraksi fitur akan disimpan dalam sebuah variabel.

Gambar 3. 3 Proses Ekstraksi Fitur Warna

untuk mentransformasikan dari RGB ke HSV. Diasumsikan koordinat R,G,B [0,1] adalah berurutan merah, hijau, biru dalam ruang warna RGB, dengan *max* adalah nilai maksimum dari nilai *red, green, blue*, dan *min* adalah nilai minimum dari nilai *red, green, blue*. Untuk memperoleh sudut hue [0,360] yang tepat untuk ruang warna HSV, menggunakan rumus seperti yang terdapat pada persamaan 2.9, rumus pada persamaan 2.7, dan 2.8 menghasilkan nilai *value* dan *saturation* dalam jangkauan RGB [0,1]. Kalikan dahulu dengan 255 untuk memperoleh nilai dengan jangkauan RGB [0,255].

Selanjutnya proses yang dilakukan untuk mendapatkan nilai *hue, saturation* dan *value* yang akan disimpan dalam variabel adalah dengan cara menghitung nilai rata-rata (*mean*) dari komponen *Hue, Saturation, dan Value* untuk seluruh gambar. Ukuran pixel yang digunakan dalam penelitian ini adalah 400x900 sehingga keseluruhan pixel yang saya miliki 360.000 pixel. Jadi dari setiap pixel tersebut akan dihitung nilai *Hue, Saturation, Value*. Kemudian ketika sudah mendapatkan keseluruhan nilai setiap pixel maka kita menghitung nilai rata-rata dari *Hue, Saturation, Value* dengan cara :

$$nilai\ H = \frac{H1 + H2 + \cdots + H360000}{360000} \tag{3.1}$$

$$nilai\ S = \frac{S1 + S2 + \cdots + S360000}{360000} \tag{3.2}$$

$$nilai\ V = \frac{V1 + V2 + \cdots + V360000}{360000} \tag{3.3}$$

Berikut merupakan contoh gambar perubahan dari RGB ke HSV.

Gambar 3. 4 RGB ke HSV

3. Ekstraksi Fitur Tekstur LBP

Proses ekstraksi fitur tekstur merupakan kelanjutan dari tahap *Preprocessing*, dimana gambar telah diubah menjadi ukuran piksel yang sesuai dan diubah menjadi grayscale. Ekstraksi fitur tekstur yang digunakan adalah dengan metode *Local Binary Pattern* (LBP). Proses ekstraksi fitur akan dihitung berdasarkan nilai ketetanggan 1, 2, 4, 8, 16, 32, 64, 128. Dimana setiap piksel yang nilainya lebih besar atau sama dengan piksel tengah akan diberikan nilai 1, sedangkan nilai piksel yang lebih kecil dari piksel tengah akan diberikan nilai 0. Setiap bilangan yang bernilai 1 kemudian dikalikan dengan perpangkatan 2. Mulai dari 20 sampai 27, sehingga nilai total yang dihasilkan adalah 256.

Berikut merupakan contoh gambar perubahan dari Grayscale ke Local Binary Pattern.

Gambar 3. 5 Grayscale ke Local Binary Pattern

Dalam tahap ini, citra akan ditransformasi menggunakan *Local Binary Pattern*. Hasil transfomasi LBP ini kemudian akan diambil nilai statistik orde

pertama, yang terdiri dari lima parameter, yaitu *mean, entropy, variance, kurtosis, dan skewness*.

Langkah yang harus dilakukan sebelum melakukan klasifikasi adalah penggabungan hasil ekstraksi fitur yaitu dengan cara cukup dengan disandingkan. Proses penyandingan data cukup dengan menambahkan data pada kolom berikutnya sebanyak fitur yang ada. Data tidak bisa dijumlahkan atau dikalikan karena memiliki jumlah fitur yang berbeda. Hasil ekstraksi yang disandingkan dalam artian jumlah masing-masing ekstraksi fitur langsung dijumlahkan pada kolom ekstraksi berikutnya.

Dalam kasus ini, data yang disandingkan terdiri dari 8 parameter, dengan 3 parameter untuk ekstraksi fitur warna, yaitu hue, saturation, dan value. 5 parameter untuk ekstraksi fitur tekstur, yaitu *mean, entropy, variance, kurtosis, dan skewness.*

4. Klasifikasi

Klasifikasi dilakukan dengan menggunakan algoritma *Support Vector Machine* (SVM) untuk mengidentifikasi nominal mata uang dari data citra. Sebelumnya, dataset yang terdiri dari 700 citra uang dibagi menjadi data latih (80%) dan data uji (20%) untuk keperluan evaluasi model. Pembagian dataset ini dilakukan secara acak,

dengan memastikan bahwa setiap subset mencerminkan distribusi kelas yang seimbang. Data latih diberi label sesuai dengan nominal uang yang sesuai, dan kemudian data uji digunakan untuk mencocokkan ciri-ciri yang ada pada data latih guna mengidentifikasi nominal mata uang dengan benar

5. Evaluasi model

   Tahap evaluasi model ini bertujuan untuk mengukur kinerja dari suatu model yang digunakan. Didalam mengevaluasi kinerja model maka dilakukan menggunakan *multiclass confusion matrix*.

   a. *Confusion matrix* merupakan tabel yang berisi hasil pengukuran performa yang berasal dari model klasifikasi *machine learning*. *Confusion matrix* ini digunakan untuk menentukan klasifikasi dalam sebuah metode apakah termasuk dalam label 1000, 2000, 5000, 10000, 20000, 50000, 100000. Didalam penelitian ini output yang dihasilkan berupa identifikasi nominal uang yang terdiri dari 7 kelas yaitu kelas 1000, 2000, 5000, 10000, 20000, 50000, dan 100000. Maka dari itu penelitian ini menggunakan *multiclass confusion matrix* 7x7. Adanya *confusion matrix* yang terdiri dari berbagai macam bagian ini dijadikan evaluasi hasil dari

prediksi yang dilakukan oleh metode *support vector machine* (SVM). berikut merupakan tabel dari multiclass confusion matrix:

Tabel 3. 4 Tabel Confusion Matrix

		Predicted Class(j)						
		1	2	5	10	20	50	100
True class (i)	1	$P_{1,1}$	$P_{1,2}$	$P_{1,5}$	$P_{1,10}$	$P_{1,20}$	$P_{1,50}$	$P_{1,100}$
	2	$P_{2,1}$	$P_{2,2}$	$P_{2,5}$	$P_{2,10}$	$P_{2,20}$	$P_{2,50}$	$P_{2,100}$
	5	$P_{5,1}$	$P_{5,2}$	$P_{5,5}$	$P_{5,10}$	$P_{5,20}$	$P_{5,50}$	$P_{5,100}$
	10	$P_{10,1}$	$P_{10,2}$	$P_{10,5}$	$P_{10,10}$	$P_{10,20}$	$P_{10,50}$	$P_{10,100}$
	20	$P_{20,1}$	$P_{20,2}$	$P_{20,5}$	$P_{20,10}$	$P_{20,20}$	$P_{20,50}$	$P_{20,100}$
	50	$P_{50,1}$	$P_{50,2}$	$P_{50,5}$	$P_{50,10}$	$P_{50,20}$	$P_{50,50}$	$P_{50,100}$
	100	$P_{100,1}$	$P_{100,2}$	$P_{100,5}$	$P_{100,10}$	$P_{100,20}$	$P_{100,50}$	$P_{100,100}$

Keterangan:

$P_{i,j}$ = Elemen dalam *confusion matrix* pada baris i dan kolom j.

i = mewakili kelas actual

j = mewakili kelas prediksi

Diagonal utama (True Postive (TP)) merupakan elemen pada diagonal utama (misalnya, $P_{1,1}$, $P_{2,2}$, $P_{5,5}$, dst.) menunjukkan jumlah prediksi yang

benar untuk setiap kelas. Misalnya, $P_{1,1}$ menunjukkan jumlah sampel yang benar-benar termasuk dalam kelas 1 dan diprediksi sebagai kelas 1.

False Positive (FP) dan False Negative (FN) merupakan elemen-elemen di luar diagonal utama menunjukkan kesalahan klasifikasi. Misalnya, $P_{1,2}$ menunjukkan jumlah sampel yang sebenarnya adalah kelas 1 tetapi diprediksi sebagai kelas 2.

False Positives (FP): Ini terjadi ketika model memprediksi kelas tertentu tetapi sebenarnya sampel tersebut berasal dari kelas lain. Misalnya, semua elemen di kolom "Prediksi 1" selain $P_{1,1}$ adalah false positives untuk kelas 1.

False Negatives (FN): Ini terjadi ketika model gagal memprediksi kelas tertentu. Misalnya, semua elemen di baris "Aktual 1" selain $P_{1,1}$ adalah false negatives untuk kelas 1.

Dalam melakukan pengukuran performa dari model yang digunakan maka diperlukannya menghitung nilai *accuracy, precision, recall, dan F1-Score*.

Accuracy adalah variable yang penggunaanya berdasarkan perhitungan persentase proporsi hasil klasifikasi yang benar.

$$Akurasi = \frac{jumlah\ prediksi\ yang\ benar}{jumlah\ total\ prediksi} \quad (3.4)$$

Precision adalah kualitas ketepatan dari sebuah informasi *actual* dengan adanya prediksi sistem. Berikut persamaannya:

$$Precision = \frac{TP}{TP + FP} \quad (3.5)$$

Recall merupakan kemampuan model untuk menangkap atau mengidentifikasi sebanyak mungkin hal yang sebenarnya positif dari total hal yang seharusnya positif. Berikut persamaannya:

$$Recall = \frac{TP}{TP + FN} \quad (3.6)$$

F1 Score adalah ukuran gabungan yang mencoba menyatukan *precision* dan *recall* menjadi satu angka untuk memberikan gambaran keseluruhan tentang kinerja model. Ini adalah cara yang bagus untuk mengevaluasi model klasifikasi karena mempertimbangkan kedua *false positives* dan *false negatives*. Berikut persamaanya:

$$F1\ Score = 2.\frac{Precision\ x\ Recall}{Precision + Recall} \quad (3.7)$$

BAB IV
HASIL DAN PEMBAHASAN

Pada bab ini akan menjelaskan terkait dengan hasil dan pembahasan dari sebuah sistem yang telah dibangun. Dengan demikian maka diperlukannya pengujian terhadap sistem dengan melalui berbagai tahapan yaitu dengan adanya tahap preprocessing, tahap ekstraksi fitur warna HSV, Ekstraksi Fitur tekstur LBP, tahap klasifikasi menggunakan metode *support vector machine* (SVM), dan yang terakhir adalah evaluasi model menggunakan *confusion matrix*. Didalam penelitian ini menggunakan data uang 1000, 2000, 5000, 10000, 20000, 50000, dan 100000 yang didapatkan dari kaggle.

4.1 Preprocessing

Tahap *Preprocessing* citra yang dilakukan adalah normalisasi ukuran citra (*resize image*). Pada proses normalisasi ukuran citra *(resize image)* bertujuan untuk memperkecil dimensi ukuran pixel citra sehingga dimensi ukuran pixel semua citra akan sama. Ukuran pixel dalam penelitian ini adalah 400 x 900 pixel.

4.2 Ekstraksi Fitur Warna HSV

Untuk mentransformasikan dari RGB ke HSV. Diasumsikan kordinat-kordinat R, G, B [0,1] adalah berurutan merah, hijau, biru dalam ruang warna RGB, dengan max adalah nilai maksimum dari nilai *red, green, blue,* dan min adalah nilai minimum dari nilai *red, green, blue.* Untuk memperoleh sudut *hue* [0,360] yang tepat untuk ruang warna HSV, menggunakan rumus seperti yang terdapat pada persamaan 2.9:

Rumus pada persamaan 2.7, 2.8 menghasilkan nilai *value* dan *saturation* dalam jangkauan RGB [0,1]. Kalikan dahulu dengan 255 untuk memperoleh nilai jangkauan RGB [0,255]. Misalnya ingin ditransformasikan RGB (227, 225, 167) ke dalam bentuk HSV, maka langkahnya adalah sebagai berikut:

Setiap nilai RGB (227, 225, 167) diubah dalam jangkauan [0,1] dengan membagi setiap nilai dengan 255:

Menjadi

$$\left(\frac{227}{255}, \frac{225}{255}, \frac{167}{255} = (0.890, 0.882, 0.655)\right)$$

RGB (0.890, 0.882, 0.655) ini yang akan ditransformasikan ke bentuk HSV.

Max = nilai R (*red*) = 0.890

Min = nilai B (*blue*) = 0.655

Max – Min = 0.890 – 0.655 = 0.235

Untuk menghitung nilai hue, gunakan rumus yang terdapat dalam persamaan 2.9

H (*hue*) = $\frac{60° \text{ x } (g-b)}{max-min}$

$= \frac{60° \text{ x } (0.882-0.655)}{0.235} = \frac{60° \text{ x } (0.227)}{0.235} = 57{,}96°$

Untuk menghitung nilai *value*, gunakan rumus pada persamaan 2.7

V (*value*) = max

= 0.890

Untuk menghitung nilai *saturation*, gunakan rumus pada persamaan 2.8

S (*saturation*) = $\frac{max-min}{V}$

$= \frac{0.235}{0.890} = 0.264$

Sehingga nilai RGB (227, 225, 167) ditransformasikan menjadi HSV (57,96°, 0.264, 0.890) dengan jangkauan RGB [0,1].

4.3 Ekstraksi Fitur Tekstur LBP

Untuk melakukan proses local binary pattern perlu dilakukan perubahan dari gambar asli ke greyscale proses tersebut dilakukan dengan menghitung nilai greyscale perpixel.

Gambar 4. 1 Gambar RGB dan Gambar Grayscale

Berikut merupakan sebagian pixel RGB dari gambar diatas:

Pixel (1, 1): R=227, G=225, B=167

Pixel (2, 1): R=227, G=225, B=169

Pixel (3, 1): R=224, G=221, B=166

Pixel (4, 1): R=219, G=218, B=163

Pixel (5, 1): R=217, G=216, B=159

Pixel (6, 1): R=221, G=220, B=164

Selanjutnya kita menghitung greyscale dengan rumus luminasi atau kecerahan untuk setiap pixel:

Pixel (1,1) ➔ I = 0.2989 x R + 0.5870 x G + 0.1140 x B

= 0.2989 x 227 + 0.5870 x 225 + 0.1140 x 167

= 67.850 + 132,075 + 19,038 = 218,963

Pixel (2,1) ➔ I = 0.2989 x R + 0.5870 x G + 0.1140 x B

= 0.2989 x 227 + 0.5870 x 225 + 0.1140 x 169

= 67.850 + 132,075 + 19,266 = 219,191

Pixel (3,1) ➔ I = 0.2989 x R + 0.5870 x G + 0.1140 x B

= 0.2989 x 224 + 0.5870 x 221 + 0.1140 x 166

= 66.953 + 129,727 + 18,924 = 215,604

Pixel (4,1) ➔ I = 0.2989 x R + 0.5870 x G + 0.1140 x B

= 0.2989 x 219 + 0.5870 x 218 + 0.1140 x 163

= 65.459 + 127,966 + 18,582 = 212,007

Pixel (5,1) ➔ I = 0.2989 x R + 0.5870 x G + 0.1140 x B

= 0.2989 x 217 + 0.5870 x 216 + 0.1140 x 159

= 64.861 + 126,792 + 18,126 = 209,779

Pixel (6,1) ➔ I = 0.2989 x R + 0.5870 x G + 0.1140 x B

= 0.2989 x 221 + 0.5870 x 220 + 0.1140 x 164

= 66.056 + 129,140 + 18,696 = 213,892

Kemudian Ilustrasi perhitungan *Local binary pattern* (LBP) algoritmanya adalah sebagai berikut:

1. Tentukan default pola yang digunakan untuk menyusun pola biner

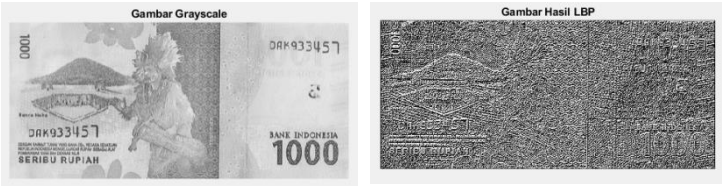
   Ada 2 kemungkinan cara penyusunan, yaitu:

   a. Searah jarum jam(*clockwise*)
   b. Berlawanan jarum jam (*counter-clockwise*)
2. Tentukan default mask atau blok data saat perhitungan LBP

Diasumsikan dalam kasus ini *default mask* adalah 3 Sehingga titik pusat dari masing-masing blok adalah posisi kedua.

Defaultmask = 3;

Titik pusat = *ceil*((*defaultmask*+1)/2);



Gambar 4. 2 Gambar Grayscale dan Local Binary Pattern

Sebagian pixel grayscale dari gambar tersebut, matriksnya adalah:

219	221	223	220	221	220
219	221	219	222	223	222
216	214	220	222	219	221
212	214	221	217	219	221
210	216	218	213	221	219
214	215	218	220	220	218

Dari matriks tersebut dapat dihitung seperti berikut:

- Perhitungan pertama

219	221	223	220	221	220
219	221	219	222	223	222
216	214	220	222	219	221
212	214	221	217	219	221

210	216	218	213	221	219
214	215	218	220	220	218

219	221	223		0	0	1
219	221	219	➔	0		0
216	214	220		0	0	0

Cara menghitung matriks tersebut gunakan persamaan 2.10 dan 2.11, dimana setiap nilai pixel dibandingkan dengan pixel tengah. Apabila nilai pixel > pixel tengah maka bernilai biner 1, apabila nilai pixel ≤ pixel tengah maka bernilai 0. Lalu masing-masing dihitung, seperti pada perhitungan dibawah ini.

$00100000 \rightarrow 0(2^7) + 0(2^6) + 1(2^5) + 0(2^4) +$

$0(2^3) + 0(2^2) + 0(2^1) + 0(2^0)$

$\rightarrow 0 + 0 + 32 + 0 + 0 + 0 + 0 + 0 = 32$

Angka 00100000 ini diperoleh dari hasil membandingkan setiap nilai pixel dengan pixel tengah, dan diurutkan sesuai dengan arah jarum jam sesuai dengan persamaan 2.11.

Lalu angka 000000 tersebut dihitung menggunakan rumus pada persamaan 2.10 sehingga menghasilkan nilai LBP sebesar 0.

- Perhitungan kedua: Gunakan rumus yang terdapat pada persamaan 2.10 dan 2.11 cara perhitungan kedua sama dengan cara perhitungan pertama.

219	221	223	220	221	220
219	221	219	222	223	222
216	214	220	222	219	221
212	214	221	217	219	221
210	216	218	213	221	219
214	215	218	220	220	218

221	223	220
221	219	222
214	220	222

➔

1	1	1
1		1
0	1	1

$11111101 \rightarrow 1(2^7) + 1(2^6) + 1(2^5) + 1(2^4) +$

$1(2^3) + 1(2^2) + 0(2^1) + 1(2^0)$

➔ 128 + 64 + 32 + 16 + 8 + 4 + 0 + 1 = 253

- Perhitungan ketiga: Gunakan rumus yang terdapat pada persamaan 2.10 dan 2.11 cara perhitungan

kedua sama dengan cara perhitungan pertama dan kedua.

219	221	223	220	221	220
219	221	219	222	223	222
216	214	220	222	219	221
212	214	221	217	219	221
210	216	218	213	221	219
214	215	218	220	220	218

223	220	221		1	0	0
219	222	223	➔	0		1
220	222	219		0	0	0

10010000 ➔ $1(2^7) + 0(2^6) + 0(2^5) + 1(2^4) +$

$0(2^3) + 0(2^2) + 0(2^1) + 0(2^0)$

➔ 128 + 0 + 0 + 16 + 0 + 0 + 0 + 0 = 144

Setelah mendapatkan nilai LBP dan citra asli telah ditransformasikan menjadi citra LBP, maka selanjutnya gunakan rumus pada persamaan 2.1 sampai 2.5 untuk mendapatkan nilai statistik orde pertama, yaitu meliputi *mean, entropy, variance,*

kurtosis, dan skewness. berikut ini merupakan sebagian pixel LBP dari gambar 4.2

0	0	0	0	0	0
0	32	253	144	0	33
0	249	180	32	249	192
0	188	32	251	180	0
0	56	72	255	0	67
0	122	20	32	64	201

langkah pertama sebelum menghitung mean, entropy, variance, kurtosis dan skewness adalah mengubah matriks menjadi datar sebagai berikut:

[0, 0, 0, 0, 0, 0, 0, 32, 253, 144, 0, 33, 0, 249, 180, 32, 249, 192, 0, 188, 32, 251, 180, 0, 0, 56, 72, 255, 0, 67, 0, 122, 20, 32, 64, 201]

Selanjutnya dilakukan perhitungan distribusi frekuensi dan probabilitas yaitu dengan menghitung nilai unik:

a. 0: 14 kali
b. 32: 4 kali
c. 33, 253, 144, 249, 180, 188, 251, 56, 72, 255, 67, 122, 20, 64, 192, 201: 1 kali masing-masing

Total jumlah elemen = 36

Distribusi probabilitas

$$p(0) = \frac{14}{36}, p(32) = \frac{4}{36}, p(33) = p(253) = p(144) = \cdots = \frac{1}{36}$$

Menghitung mean (μ):

$$\mu = \sum_{n=0}^{N} f_n . p(f_n)$$

$$\mu = \left(0.\frac{14}{36}\right) + \left(32.\frac{4}{36}\right) + \left(33.\frac{1}{36}\right) + \cdots = 80.67$$

Menghitung entropy (H):

$$H = -\sum_{n=0}^{N} (p\ (f_n).^2\ \log\ p\ (f_n))$$

$$H = -\left(\left(\frac{14}{36}\right)^2 log\left(\frac{14}{36}\right) + \left(\frac{4}{36}\right)^2 log\left(\frac{4}{36}\right) + \cdots\right) = 0.23$$

Menghitung variance (σ^2)

$$\sigma^2 = \sum_{n=0}^{N} (f_n - \mu)^2\ p\ (f_n)$$

$$\sigma^2 = \left((0 - 80.67)^2 . + \frac{14}{36}\right) + \left((32 - 80.67)^2 . + \frac{4}{36}\right) + \cdots = 8813.89$$

Menghitung kurtosis (α^4)

$$\alpha^4 = \frac{1}{\sigma^4} \sum_{n=0}^{N} (f_n - \mu)^4\ p\ (f_n) - 3$$

$$\alpha^4 = \frac{1}{(8813.89)^4}\left((0 - 80.67)^4 . \frac{14}{36} + (32 - 80.67)^4 . \frac{4}{36} + \cdots\right) - 3$$

$$= -0.988$$

Menghitung skewness (α^3)

$$\alpha^3 = \frac{1}{\sigma^3}\sum_{n=0}^{N} (f_n - \mu)^3 \, p\,(f_n)$$

$$\alpha^3 = \frac{1}{8813.89^3}\left((0 - 80.67)^3.\frac{14}{36} + (32 - 80.67)^3.\frac{4}{36} + \cdots\right)$$

$$= 0.785$$

4.4 Klasifikasi

Untuk melakukan klasifikasi citra pengenalan nominal mata uang, penulis menggunakan metode *Support Vector Machine* (SVM). Metode SVM yang digunakan adalah *One-Against-All*. Klasifikasi dengan metode *One-Against-All* dilakukan berdasarkan banyaknya jumlah kelas (grup), yakni akan terdapat 7 kali klasifikasi. Saat data uji diinputkan, SVM akan membandingkan data uji dengan semua grup dari data latih hingga data uji tersebut masuk ke dalam suatu grup yang spesifik. Klasifikasi ini menentukan jenis uang dari inputan yang dimasukkan, antara uang 1000, 2000, 5000, 10000, 20000, 50000, dan 100000. Adapun langkah-langkahnya sebagai berikut :

1. Inisialisasi permasalahn SVM sebagai A X B :
   Misalkan ada 7 kelas, yaitu: Kelas 1 = 1000, Kelas 2 = 2000, Kelas 3 = 5000, Kelas 4 = 10000, Kelas 5 = 20000, Kelas 6 = 50000, Kelas 7 = 100000. Untuk setiap kelas k, buat model SVM yang memisahkan

kelas k (label positif) dengan semua kelas lainnya (label negatif). Jadi akan ada 7 model SVM yang dilatih:

a. Model SVM 1: Kelas 1 (1000) vs Semua Kelas lainnya
b. Model SVM 2: Kelas 2 (2000) vs Semua Kelas lainnya
c. Model SVM 3: Kelas 3 (5000) vs Semua Kelas lainnya
d. Model SVM 4: Kelas 4 (10000) vs Semua Kelas lainnya
e. Model SVM 5: Kelas 5 (20000) vs Semua Kelas lainnya
f. Model SVM 6: Kelas 6 (50000) vs Semua Kelas lainnya
g. Model SVM 7: Kelas 7 (100000) vs Semua Kelas lainnya

Saat data uji diinputkan, setiap model SVM akan memberikan prediksi apakah data tersebut termasuk dalam kelas yang spesifik atau tidak.

2. Input

Data citra uang kertas sebanyak 700 buah citra dengan masukan parameter fitur warna HSV sebagai atribut, yaitu *hue, saturation,* dan *value.*

Juga termasuk parameter statistik orde pertama dari Local Binary Pattern (LBP) yaitu, *mean, entropy, variance, kurtosis, dan skewness*.

3. Proses Klasifikasi

   Untuk setiap data uji x, gunakan setiap model SVM untuk memprediksi apakah x termasuk dalam kelas yang spesifik. Proses ini dilakukan dengan 7 kali klasifikasi (satu untuk setiap model SVM). Misalnya, jika x adalah kelas 1000 (kelas 1): Jika x diklasifikasikan sebagai 1000 oleh Model SVM 1, maka x adalah kelas 1 (1000). Jika x tidak diklasifikasikan sebagai 1000 oleh Model SVM 1, lanjutkan ke Model SVM 2. Ulangi proses ini hingga menemukan kelas yang tepat untuk x.

4. Langkah-langkah klasifikasi lebih rinci:

   a. Langkah 1: buat inisialisasi permaslahan SVM sebagai A x B, dimana $A = k$ dan $B = \{k + 1, k + 2, \ldots, n\}$ dan $k = 1$, lalu kelas = klasifikasi (SVM(A,B),T).

   b. Langkah 2: input data $x = \{x1, x2, \ldots, xn\}$.

      Data citra uang kertas sebanyak 700 buah citra dengan masukan parameter fitur warna HSV sebagai atribut, yaitu *hue, saturation, dan value*. Juga termasuk parameter statistik orde

pertama dari *Local Binary Pattern* (LBP) yaitu, *mean, entropy, variance, kurtosis, dan skewness*. Jumlah kelas $k = 7$ terdiri atas:

kelas 1 = 1000, kelas 2 = 2000, kelas 3 = 5000, kelas 4 = 10000, kelas 5 = 20000, kelas 6 = 50000, kelas 7 = 100000.

c. Langkah 3: Apabila $x1 = 1$, maka $x1$ merupakan jenis klasifikasi A, dimana nilai A merupakan $k = 1$.

d. Langkah 4: Apabila $x1 = -1$, maka $x1$ akan diuji dengan B, dimana nilai B merupakan $k + 1, k + 2, \ldots, n\}$.

e. Langkah 5: Proses klasifikasi akan berhenti saat x menemukan kelas aktual yang sesuai dengan nilai T. Sebaliknya, apabila x belum menemukan kelas yang sesuai maka akan kembali ke langkah 4.

4.5 Evaluasi Model

Tujuan dilakukannya evaluasi model yaitu untuk nilai dari performa pada model yang digunakan. Perhitungan performa ini meliputi nilai *accuracy, precision, recall dan F1 Score* berdasarkan multiclass confusion matrix. Pada tabel 4.1 dibawah ini

menunjukkan hasil dari *multiclass confusion matrix* 7x7.

Tabel 4. 1 Tabel hasil confusion matrix 7x7

		Predicted Class(j)						
		1	2	5	10	20	50	100
True class (i)	1	19	0	0	0	1	0	0
	2	1	18	0	1	0	0	0
	5	0	0	19	0	0	0	0
	10	0	0	0	20	0	0	0
	20	0	0	0	0	20	0	0
	50	0	0	0	0	0	20	0
	100	0	0	0	0	2	1	17

Setelah didapatkan nilai dari *multiclass confusion matrix* maka akan dilakukan proses perhitungan performa dari model yang digunakan secara manual. Hasil perhitungan akurasi yang dilakukan secara manual akan dijelaskan pada proses dibawah ini.

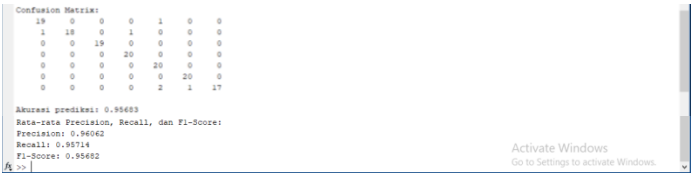
$$Accuracy = \frac{T1 + T2 + T5 + T10 + T20 + T50 + T100}{jumlah\ keseluruhan\ data} \times 100$$

$$= \frac{19+18+19+20+20+20+17}{139} \times 100$$

$$= \frac{133}{139} \times 100$$

= 0.956 x 100

= 95%



Berdasarkan gambar 4.2 dapat diketahui bahwa nilai akurasi sebesar 0.956 atau 95%. Kemudian rata-rata nilai *precision* adalah 0.960 atau 96%, rata-rata *recall* adalah 0.957 atau 95%, dan rata-rata *F1- score* adalah 0.956 atau 95%. namun untuk mendapatkan nilai rata-rata dari *precission*, *recall*, dan *f1 score* perlu dilakukan perhitungan performa model dahulu. Yaitu setelah diketahui nilai akurasinya maka selanjutnya adalah melakukan perhitungan terhadap performa model yang lainnya seperti *precission*, *recall*, dan juga *F1 Score* berdasarkan nilai *true positif*, nilai *false positif*, juga nilai *false negatif*. Karena dalam menentukan nilai *true positif*, nilai *false positif*, juga nilai *false negatif* pada *multiclass confusion matrix* 7x7 sangat sulit maka solusi yang diberikan untuk mempermudah pencarian masing-masing nilai yaitu

memecah kolom 7x7 menjadi 2x2 terlebih dahulu yang akan ditunjukan pada tabel dibawah ini:

a. Kelas 1000

Tabel 4. 2 perhitungan kelas 1000

True/Predict	1000	Bukan 1000
1000	TP = 19	FN = 0 + 0 + 0 + 1 + 0 + 0 =1
Bukan 1000	FP = 1 + 0 + 0 + 0 + 0 + 0 = 1	TN = 118

Untuk perhitungan *Precision* menggunakan persamaan dibawah ini:

$Precision = \frac{TP}{TP+FP}$

$Precision\ 1000 = \frac{19}{19+1}$

$Precision\ 1000 = \frac{19}{20}$

$Precision\ 1000 = 0.95 = 95\%$

Untuk perhitungan *Recall* menggunakan persamaan dibawah ini:

$Recall = \frac{TP}{TP+FP}$

$Recall\ 1000 = \frac{19}{19+1}$

$Recall\ 1000 = \frac{19}{20}$

$Recall\ 1000 = 0.95 = 95\%$

Untuk perhitungan *F1-Score* menggunakan persamaan dibawah ini:

$$\text{F1-Score} = 2 \times \frac{Precision \times recall}{Precision + recall}$$

$$\text{F1-Score 1000} = 2 \times \frac{0.95 \times 0.95}{0.95 + 0.95}$$

$$\text{F1-Score 1000} = 2 \times \frac{0.9025}{1.9}$$

$$\text{F1-Score 1000} = 2 \times 0.475$$

$$\text{F1-Score 1000} = 0.95 = 95\%$$

b. Kelas 2000

Tabel 4. 3 perhitungan kelas 2000

True/Predict	2000	Bukan 2000
2000	TP = 18	FN = 1 + 0 + 1 + 0 + 0 + 0 = 2
Bukan 2000	FP = 0 + 0 + 0 + 0 + 0 + 0 = 0	TN = 119

Untuk perhitungan *Precision* menggunakan persamaan dibawah ini:

$$\textit{Precision} = \frac{TP}{TP+FP}$$

$$\textit{Precision}\ 2000 = \frac{18}{18+0}$$

$$\textit{Precision}\ 2000 = \frac{18}{18}$$

$$\textit{Precision}\ 2000 = 1 = 100\%$$

Untuk perhitungan *Recall* menggunakan persamaan dibawah ini:

$$Recall = \frac{TP}{TP+FP}$$

$$Recall\ 2000 = \frac{18}{18+2}$$

$$Recall\ 2000 = \frac{18}{20}$$

$$Recall\ 2000 = 0.90 = 90\%$$

Untuk perhitungan *F1-Score* menggunakan persamaan dibawah ini:

$$\text{F1-Score} = 2 \times \frac{Precision \times recall}{Precision + recall}$$

$$\text{F1-Score 2000} = 2 \times \frac{1 \times 0.90}{1 + 0.95}$$

$$\text{F1-Score 2000} = 2 \times \frac{0.9}{1.95}$$

$$\text{F1-Score 2000} = 2 \times 0.461$$

$$\text{F1-Score 2000} = 0.923 = 92\%$$

c. Kelas 5000

Tabel 4. 4 perhitungan kelas 5000

True/Predict	5000	Bukan 5000
5000	TP = 19	FN = 0 + 0 + 0 + 0 + 0 + 0 = 0
Bukan 5000	FP = 0 + 0 + 0 + 0 + 0 + 0 = 0	TN = 120

Untuk perhitungan *Precision* menggunakan persamaan dibawah ini:

Precision = $\frac{TP}{TP+FP}$

Precision 5000 = $\frac{19}{19+0}$

Precision 5000 = $\frac{19}{19}$

Precision 5000 = $1 = 100\%$

Untuk perhitungan *Recall* menggunakan persamaan dibawah ini:

Recall = $\frac{TP}{TP+FP}$

Recall 5000 = $\frac{19}{19+0}$

Recall 5000 = $\frac{19}{19}$

Recall 5000 = $1 = 100\%$

Untuk perhitungan *F1-Score* menggunakan persamaan dibawah ini:

F1-Score = $2 \times \frac{Precision \times recall}{Precision + recall}$

F1-Score 5000 = $2 \times \frac{1 \times 1}{1 + 1}$

F1-Score 5000 = $2 \times \frac{1}{2}$

F1-Score 5000 = 2×0.5

F1-Score 5000 = $1 = 100\%$

d. Kelas 10000

Tabel 4. 5 perhitungan kelas 10000

True/Predict	10000	Bukan 10000
10000	TP = 20	FN = 0 + 0 + 0 + 0 + 0 + 0 = 0
Bukan 10000	FP = 0 + 1 + 0 + 0 + 0 + 0 = 1	TN = 118

Untuk perhitungan *Precision* menggunakan persamaan dibawah ini:

Precision = $\frac{TP}{TP+FP}$

Precision 10000 = $\frac{20}{20+1}$

Precision 10000 = $\frac{20}{21}$

Precision 10000 = 0.95 = 95%

Untuk perhitungan *Recall* menggunakan persamaan dibawah ini:

Recall = $\frac{TP}{TP+FP}$

Recall 10000 = $\frac{20}{20+0}$

Recall 10000 = $\frac{20}{20}$

Recall 10000 = 1 = 100%

Untuk perhitungan *F1-Score* menggunakan persamaan dibawah ini:

F1-Score = $2 \times \frac{Precision \times recall}{Precision + recall}$

F1-Score 10000 = $2 \times \frac{0.95 \times 1}{0.95 + 1}$

F1-Score 10000 = $2 \times \frac{0.95}{1.95}$

F1-Score 10000 = 2×0.487

F1-Score 10000 = $0.974 = 97\%$

e. Kelas 20000

Tabel 4. 6 perhitungan kelas 20000

True/Predict	20000	Bukan 20000
20000	TP = 20	FN = 0 + 0 + 0 + 0 + 0 + 0 = 0
Bukan 20000	FP = 1 + 0 + 0 + 0 + 0 + 2 = 3	TN = 116

Untuk perhitungan *Precision* menggunakan persamaan dibawah ini:

Precision = $\frac{TP}{TP+FP}$

Precision 20000 = $\frac{20}{20+3}$

Precision 20000 = $\frac{20}{23}$

Precision 20000 = $0.86 = 86\%$

Untuk perhitungan *Recall* menggunakan persamaan dibawah ini:

Recall = $\frac{TP}{TP+FP}$

Recall 20000 = $\frac{20}{20+0}$

Recall 20000 = $\frac{20}{20}$

Recall 20000 = $1 = 100\%$

Untuk perhitungan *F1-Score* menggunakan persamaan dibawah ini:

F1-Score = $2 \times \frac{Precision \times recall}{Precision + recall}$

F1-Score 20000 = $2 \times \frac{0.86 \times 1}{0.86 + 1}$

F1-Score 20000 = $2 \times \frac{0.86}{1.86}$

F1-Score 20000 = 2×0.462

F1-Score 20000 = $0.924 = 92\%$

f. Kelas 50000

Tabel 4. 7 peritungan kelas 50000

True/Predict	50000	Bukan 50000
50000	TP = 20	FN = 0 + 0 + 0 + 0 + 0 + 0 = 0
Bukan 50000	FP = 0 + 0 + 0 + 0 + 0 + 1 = 1	TN = 118

Untuk perhitungan *Precision* menggunakan persamaan dibawah ini:

Precision = $\frac{TP}{TP+FP}$

Precision 50000 = $\frac{20}{20+1}$

Precision 50000 = $\frac{20}{21}$

Precision 50000 = 0.95 = 95%

Untuk perhitungan *Recall* menggunakan persamaan dibawah ini:

Recall = $\frac{TP}{TP+FP}$

Recall 50000 = $\frac{20}{20+0}$

Recall 50000 = $\frac{20}{20}$

Recall 50000 = 1 = 100%

Untuk perhitungan *F1-Score* menggunakan persamaan dibawah ini:

F1-Score = $2 \times \frac{Precision \times recall}{Precision + recall}$

F1-Score 50000 = $2 \times \frac{0.95 \times 1}{0.95 + 1}$

F1-Score 50000 = $2 \times \frac{0.95}{1.95}$

F1-Score 50000 = 2×0.487

F1-Score 50000 = 0.974 = 97%

g. Kelas 100000

Tabel 4. 8 perhitungan kelas 100000

True/Predict	100000	Bukan 100000
100000	TP = 17	FN = 0 + 0 + 0 + 0 + 2 + 1 = 0
Bukan 100000	FP = 0 + 0 + 0 + 0 + 0 + 0 = 0	TN = 119

Untuk perhitungan *Precision* menggunakan persamaan dibawah ini:

$$Precision = \frac{TP}{TP+FP}$$

$$Precision\ 10000 = \frac{17}{17+0}$$

$$Precision\ 10000 = \frac{17}{17}$$

$$Precision\ 10000 = 1 = 100\%$$

Untuk perhitungan *Recall* menggunakan persamaan dibawah ini:

$$Recall = \frac{TP}{TP+FP}$$

$$Recall\ 10000 = \frac{17}{17+3}$$

$$Recall\ 10000 = \frac{17}{20}$$

$$Recall\ 10000 = 0.85 = 0.85\%$$

Untuk perhitungan *F1-Score* menggunakan persamaan dibawah ini:

$$\text{F1-Score} = 2 \times \frac{Precision \times recall}{Precision + recall}$$

$$\text{F1-Score } 10000 = 2 \times \frac{1 \times 0.85}{1 + 0.85}$$

$$\text{F1-Score } 10000 = 2 \times \frac{0.85}{1.85}$$

$$\text{F1-Score } 10000 = 2 \times 0.459$$

$$\text{F1-Score } 10000 = 0.918 = 91\%$$

Hasil ringkasan dari perhitungan diatas dapat dilihat pada tabel 4.9 dibawah ini:

Tabel 4. 9 hasil perhitungan performa model

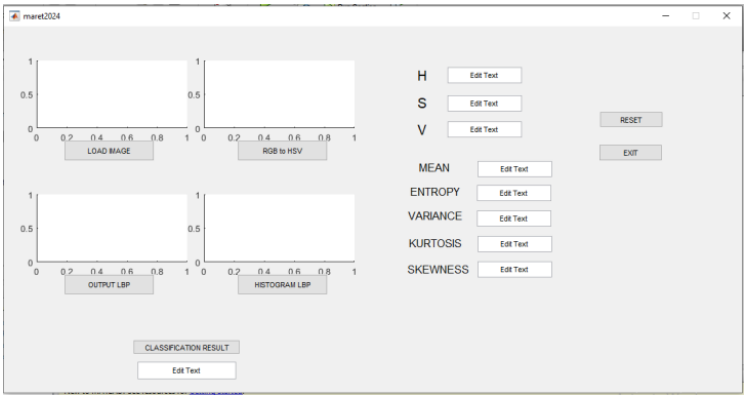
Kelas	Precision	Recall	F1-Score
1000	0.95	0.95	0.95
2000	1	0.90	0.92
5000	1	1	1
10000	0.95	1	0.97
20000	0.86	1	0.92
50000	0.95	1	0.97
100000	1	0.85	0.91

Berdasarkan tabel 4.9 diatas, untuk precision kelas 1000 sebesar 95%, *precission* kelas 2000 sebesar 100%, *precision* kelas 5000 sebesar 100%, *precision* kelas 10000 sebesar 95%, *precision* kelas 20000 sebesar 86%, *precision* kelas 50000 sebesar 95%, *precision* kelas 100000 sebesar 100%. Untuk *recall* kelas 1000 sebesar 95%, *recall* kelas 2000 sebesar 90%, *recall* kelas 5000 sebesar 100%, *recall* kelas 10000 sebesar 100%, *recall* kelas 20000 sebesar 100%, *recall* kelas 50000 sebesar 100%, *recall* kelas 100000 sebesar 85%. Untuk *F1-Score* kelas 1000 sebesar 95%, *F1-Score* kelas 2000 sebesar 92%, *F1-Score* kelas 5000 sebesar 100%, *F1-Score* kelas 10000 sebesar 97%, *F1-*

Score kelas 20000 sebesar 92%, *F1-Score* kelas 50000 sebesar 97%, *F1-Score* kelas 100000 sebesar 91%.

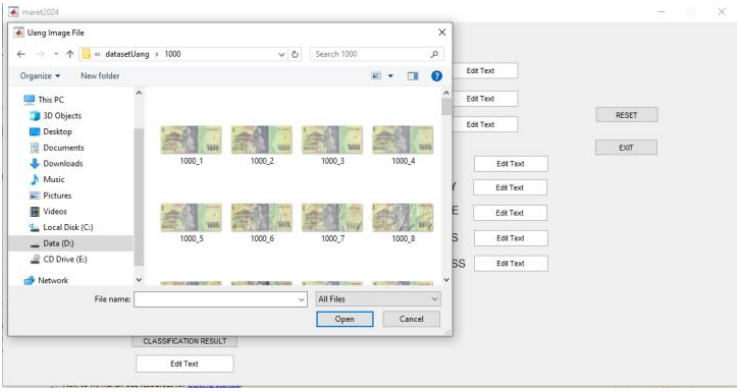
4.6 Tampilan GUI MATLAB

1. Tampilan antarmuka awal



Gambar 4. 3 Antarmuka awal

2. Tampilan antarmuka input citra



Gambar 4. 4 antarmuka input citra

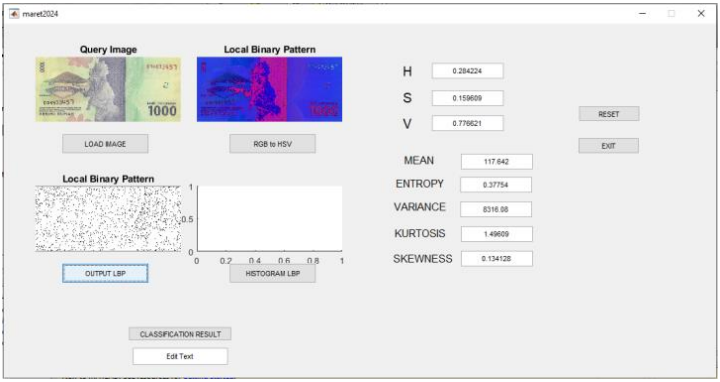
3. Tampilan antarmuka hasil input citra

Gambar 4. 5 antarmuka hasil input citra

4. Tampilan antarmuka Ekstraksi fitur warna RGB ke HSV

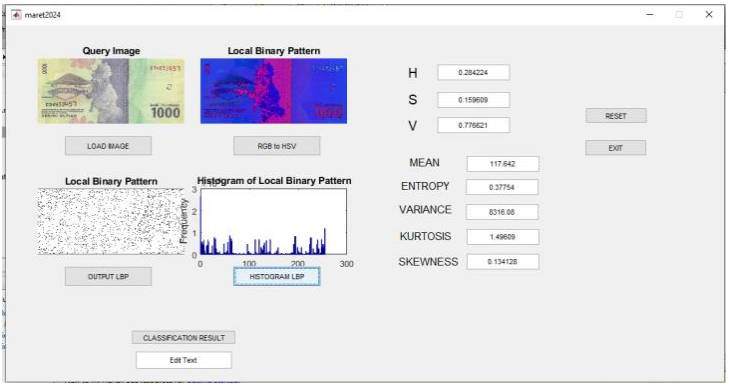
Gambar 4. 6 antarmuka Ekstraksi fitur warna RGB ke HSV

5. Tampilan antarmuka ekstraksi fitur tekstur Local Binnary Pattern



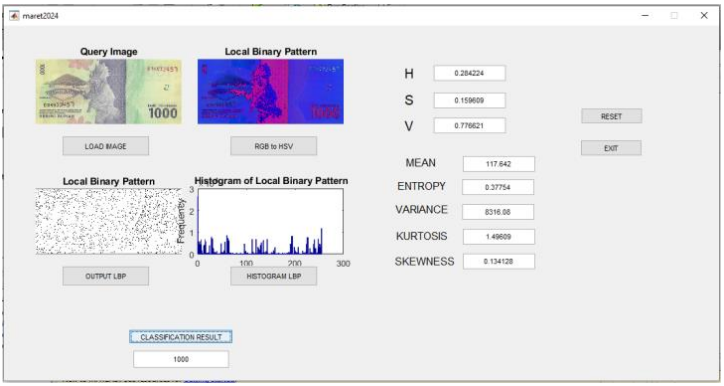
Gambar 4. 7 antarmuka ekstraksi fitur tekstur Local Binnary Pattern

6. Tampilan antarmuka histogram *local binary pattern*



Gambar 4. 8 antarmuka histogram local binary pattern

7. Tampilan antarmuka klasifikasi *support vector machine*



Gambar 4. 9 antarmuka klasifikasi support vector machine

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pembahasan dari penelitian yang dilakukan oleh peneliti dapat disimpulkan sebagai berikut:

1. Penggunaan metode *support vector machine* dalam melakukan klasifikasi pada pengenalan nominal mata uang rupiah emisi 2016 dan 2022 dengan menggunakan 700 data dapat menghasilkan klasifikasi yang cukup baik. Kemudian penggunaan nilai fitur warna hsv yaitu *hue, saturation, value* dan fitur tekstur lbp yaitu berupa *mean, variance, entropy, skewness, kurtosis* dapat digunakan sebagai acuan dalam proses klasifikasi pengenalan nominal mata uang. Proses yang diawali dengan processing berupa input gambar RGB yang kemudian dirubah dari RGB menjadi hsv. langkah selanjutnya merubah citra input menjadi *grayscale* lalu dilakukan proses ekstraksi fitur tekstur local binary pattern. Dari kedua hasil tersebut dilakukan proses klasifikasi menggunakan metode *Support Vector Machine* (SVM) dilanjutkan dengan pengujian serta

evaluasi yang menghasilkan 7 kelas yaitu kelas 1000, kelas 2000, kelas 5000, kelas 10000, kelas 20000, kelas 50000, kelas 100000.

2. Hasil penerapan support vector machine pada pengenalan nominal mata uang rupiah berdasarkan ciri warna dan tekstur mendapat akurasi sebesar 0.956 atau 95%, *precision* sebesar 0.960 atau 96%, *recall* sebesar 0.957 atau 95%, dan *F1- score* sebesar 0.956 atau 95%.
3. Untuk simulasi pengenalan nominal mata uang dalam penelitian ini terbatas hanya dari gambar yang dipakai didataset. Karena apabila ingin menggunakan data baru berupa uang gambar maka perlu disesuaikan spesifikasi gambar agar sistem dapat mendeteksi uang tersebut.

5.2 Saran

Berdasarkan hasil penelitian yang sudah digunakan, peneliti menyadari bahwa masih terapat kekurangan sehingga pada penelitian berikutnya bisa dikembangkan kembali. Adapun saran yang bisa diberikan seperti dibawah ini:

1. Menerapkan penggunaan algoritma klasifikasi yang lainya seperti *Decision Tree*, *K-means*

Clustering, *Random Forest* atau beberapa algoritma lainya agar menghasilkan performa yang dapat dilakukan perbandingan untuk mengetahui klasifikasi terbaik

2. Pada pengenalan nominal mata uang ini berdasarkan nilai ciri warna dan tekstur menggunakan klasifikasi SVM yang diproses menggunakan matlab peneliti menyarankan untuk mencoba menggunakan metode teknik matching yang dapat menggunakan *jupyter* atau *google collab*. Membandingkan kinerja berbagai model ini dapat memberikan pemahaman yang lebih mendalam tentang model mana yang lebih efektif untuk pengenalan nominal mata uang.
3. Pada penelitian selanjutnya mungkin bisa ditambahkan untuk pengenalan nominal uang dengan data baru yang tidak hanya diambil dari dataset bisa dengan menggunakan pengenalan nominal mata uang secara *realtime*.

DAFTAR PUSTAKA

Birowo. (2020). Pengolahan Citra Untuk Pengenalan Nilai Nominal Pada Mata Uang Kertas Dengan Metode EigenFace. *Jurnal Inovasi Dan Sains Teknik Elektro*, *1*(1), 20–27. http://ejournal.bsi.ac.id/ejurnal/index.php/insantek

Darma, Putra (2010). *Pengolahan Citra Digital.* Yogyakarta: Andi.

Izah, R. N. (2018). *Klasifikasi Nominal Uang Kertas Rupiah Tahun Emisi 2017 Dengan Algoritma Convolutional Neural Network Menggunakan Mxnet*. *372*(2), 2499–2508. http://www.ncbi.nlm.nih.gov/pubmed/7556065%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC394507%0Ahttp://dx.doi.org/10.1016/j.humpath.2017.05.005%0Ahttps://doi.org/10.1007/s00401-018-1825-z%0Ahttp://www.ncbi.nlm.nih.gov/pubmed/27157931

Kadir, A., & Susanto, A. (2013). *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andi.

Kiswanto. (2012). *Identifikasi Citra Untuk Mengidentifikasi Jenis Daging Sapi Dengan Menggunakan Tramsformasi Wavelet Haar.* Universitas Diponegoro, 1(2), 73-79.

https://doi.org/10.21456/vol1iss2pp73-79

Munir, Rinaldi. 2004. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung. Penerbit: Informatika.

Muntasa, A., & Purnomo, H. M. (2010). *Konsep Pengolahan Citra Digital dan Ekstraksi Fitur*. Yogyakarta: Graha Ilmu.

Parker, J.R. (2011). *Algorithm For Image Processing And Computer Vision*. Canada: Willey Publishing, Inc.

Pratama, A. R., Mustajib, M., & Nugroho, A. (2020). Deteksi Citra Uang Kertas dengan Fitur RGB Menggunakan K-Nearest Neighbor. *Jurnal Eksplora Informatika*, *9*(2), 163–172. https://doi.org/10.30864/eksplora.v9i2.336

Rifqo, M. H., Darnita, Y., Deslianti, D., & Zen, W. (2022). Application of Hough Transformation Method for Value Analysis Rupiah Coins. *Jurnal Komputer, Informasi Dan Teknologi (Jkomitek)*, *2*(2), 247–258. https://doi.org/10.53697/jkomitek.v2i2.855

Samudra, D. A., Via, Y. V., & Puspaningrum, E. Y. (2020). Deteksi Nominal Mata Uang Dengan Template Matching Currency Nominal Detection With Template Matching. *Jurusan Teknik Informatika, Fakultas Ilmu Komputer UPN "Veteran" Jawa Timur*, 1–14.

Sari, W. S., & Sari, C. A. (2023). Multi-SVM Dalam Identifikasi Bunga Berbasis Ekstraksi Ciri Orde Satu. *InComTech :*

Jurnal Telekomunikasi Dan Komputer, *13*(1), 18. https://doi.org/10.22441/incomtech.v13i1.15012

Ulfah, J., & Nurdin, N. (2023). Implementasi Metode Deteksi Tepi Canny Untuk Menghitung Jumlah Uang Koin Dalam Gambar Menggunakan Opencv. *Jurnal Informatika Dan Teknik Elektro Terapan*, *11*(3), 420–426. https://doi.org/10.23960/jitet.v11i3.3147

Wahjudi, D. (2018). *Sejarah Uang*. *13*, 1–34.

DAFTAR LAMPIRAN

LAMPIRAN 1 : Source Code GUI

```
1. function pushbutton1_Callback(hObject, eventdata,
   handles)
2. % hObject    handle to pushbutton1 (see GCBO)
3. % eventdata  reserved - to be defined in a future
   version of MATLAB
4. % handles    structure with handles and user data
   (see GUIDATA)
5. % Clear the axes
6. axes(handles.axes1);
7. cla;
8. axes(handles.axes2);
9. cla;
10. axes(handles.axes3);
11. cla;
12. axes(handles.axes4);
13. cla;
14.
15. % Clear the edit boxes
16. set(handles.edit1, 'string', '');
17. set(handles.edit2, 'string', '');
18. set(handles.edit3, 'string', '');
19. set(handles.edit4, 'string', '');
20. set(handles.edit5, 'string', '');
21. set(handles.edit6, 'string', '');
22. set(handles.edit7, 'string', '');
23. set(handles.edit8, 'string', '');
24. set(handles.edit9, 'string', '');
25.
26. % Clear the handles data
27. handles.ImgData1 = [];
28. handles.ImgData2 = [];
29. handles.ImgData3 = [];
30.
31. % Update handles structure
32. guidata(hObject, handles);
33.
34. % --- Executes on button press in pushbutton2.
35. function pushbutton2_Callback(hObject, eventdata,
   handles)
36. % hObject    handle to pushbutton2 (see GCBO)
37. % eventdata  reserved - to be defined in a future
   version of MATLAB
38. % handles    structure with handles and user data
   (see GUIDATA)
39.
40.
```

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
%clear all
%close all
%mengambil data citra dapat berformat .bmp,.jpg,dan
.gif
clc
[filename,pathname]=uigetfile({'*.*';'*.bmp';'*.jpg';
'*.gif'},'Uang Image File');
%membaca citra
I = imread([pathname,filename]);
%mengubah ukuran menjadi 400x900 px
I = imresize(I,[400,900]);
I2 = imresize(I,[400,900]);
axes(handles.axes1);
imshow(I2);title('Query Image');
ss = ones(400,900);
axes(handles.axes2);
imshow(ss);
axes(handles.axes3);
%menampilkan citra yang diinput
imshow(ss);
handles.ImgData1 = I;
guidata(hObject,handles);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata,
handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data
(see GUIDATA)
% Access the local binary pattern image data
localBinaryPatternImage = handles.ImgData3;

% Compute the histogram using the hist function
[counts, edges] = hist(localBinaryPatternImage(:),
max(localBinaryPatternImage(:)) + 1);

% Plot the histogram
axes(handles.axes4);
bar(edges, counts, 'hist');
title('Histogram of Local Binary Pattern');
xlabel('Pixel Value');
ylabel('Frequency');
```

```
84.
85. % --- Executes on button press in pushbutton5.
86. function pushbutton5_Callback(hObject, eventdata,
    handles)
87. % hObject    handle to pushbutton5 (see GCBO)
88. % eventdata  reserved - to be defined in a future
    version of MATLAB
89. % handles    structure with handles and user data
    (see GUIDATA)
90. I3 = handles.ImgData1;
91. I4 = rgb2hsv(I3);
92. I5 = imresize(I4,[400,900]);
93. H = I4(:,:,1);
94. S = I4(:,:,2);
95. V = I4(:,:,3);
96. nilai_H = mean2(H);
97. nilai_S = mean2(S);
98. nilai_V = mean2(V);
99. set(handles.edit2,'string',nilai_H);
100.      set(handles.edit3,'string',nilai_S);
101.      set(handles.edit4,'string',nilai_V);
102.      axes(handles.axes2);
103.      imshow(I5);title('RGB to HSV');
104.      handles.ImgData2 = I4;
105.      guidata(hObject,handles);
106.      % --- Executes on button press in pushbutton6.
107.      function pushbutton6_Callback(hObject,
    eventdata, handles)
108.      % hObject    handle to pushbutton6 (see GCBO)
109.      % eventdata  reserved - to be defined in a
    future version of MATLAB
110.      % handles    structure with handles and user
    data (see GUIDATA)
111.      Img = handles.ImgData1;
112.      input = rgb2gray(Img);
113.      [rows columns numberOfColorBands] =
    size(input);
114.      %Display the original gray scale image
115.      %preallocate/instantiate array for the local
    binary pattern.
116.      localBinaryPatternImage = zeros(size(input));
117.      for row = 2 : rows - 1
118.        for col = 2 : columns - 1
119.           centerPixel = input(row,col);
120.           pixel7=input(row-1,col-1)>centerPixel;
121.           pixel6=input(row-1,col)>centerPixel;
122.           pixel5=input(row-1,col+1)>centerPixel;
123.           pixel4=input(row,col+1)>centerPixel;
124.           pixel3=input(row+1,col+1)>centerPixel;
125.           pixel2=input(row+1,col)>centerPixel;
126.           pixel1=input(row+1,col-1)>centerPixel;
127.           pixel0=input(row,col-1)>centerPixel;
```

```
            localBinaryPatternImage(row,col)=uint8(...
                pixel7*2^7+pixel6*2^6+...
                pixel5*2^5+pixel4*2^4+...
                pixel3*2^3+pixel2*2^2+...
                pixel1*2+pixel0);
          end
        end
        title('Local Binary Pattern');
        grid on
        axes(handles.axes3);
        imshow(localBinaryPatternImage);
        guidata(hObject,handles);
        set(imshow(localBinaryPatternImage));

        %%Extract Features
        %Derive Statistic
        Mean = mean2(localBinaryPatternImage);
        Entropy = entropy(localBinaryPatternImage(:));
        Variance = var(localBinaryPatternImage(:));
        Kurtosis =
    kurtosis(localBinaryPatternImage(:));
        Skewness =
    skewness(localBinaryPatternImage(:));

        I7 =
    imresize(localBinaryPatternImage,[400,900]);
        axes(handles.axes3);
        imshow(I7);title('Local Binary Pattern');
        set(handles.edit5,'string',Mean);
        set(handles.edit6,'string',Entropy);
        set(handles.edit7,'string',Variance);
        set(handles.edit8,'string',Kurtosis);
        set(handles.edit9,'string',Skewness);
        guidata(hObject,handles);

        % Store the local binary pattern image data in
    handles
        handles.ImgData3 = localBinaryPatternImage;
        guidata(hObject, handles);

        % --- Executes on button press in pushbutton7.
        function pushbutton7_Callback(hObject,
    eventdata, handles)
        % hObject    handle to pushbutton7 (see GCBO)
        % eventdata  reserved - to be defined in a
    future version of MATLAB
        % handles    structure with handles and user
    data (see GUIDATA)

        % Load the trained SVM model
        load('trainedSVMModel.mat');

```

```
173.      % Access the local binary pattern image data
174.      localBinaryPatternImage = handles.ImgData3;
175.
176.      % Extract LBP features
177.      Mean_LBP = mean2(localBinaryPatternImage);
178.      Entropy_LBP = entropy(localBinaryPatternImage);
179.      Variance_LBP =
    mean2(var(double(localBinaryPatternImage)));
180.      Kurtosis_LBP =
    kurtosis(double(localBinaryPatternImage(:)));
181.      Skewness_LBP =
    skewness(double(localBinaryPatternImage(:)));
182.
183.      % Access the HSV image data
184.      HSV_Image = handles.ImgData2;
185.      H_Channel = HSV_Image(:,:,1);
186.      S_Channel = HSV_Image(:,:,2);
187.      V_Channel = HSV_Image(:,:,3);
188.
189.      % Extract HSV features
190.      Mean_H = mean2(H_Channel);
191.      Mean_S = mean2(S_Channel);
192.      Mean_V = mean2(V_Channel);
193.
194.      % Combine features into one vector
195.      features = [Mean_H, Mean_S, Mean_V, Mean_LBP,
    Entropy_LBP, Variance_LBP, Kurtosis_LBP,
    Skewness_LBP];
196.
197.      % Initialize an array to store predictions
198.      predictions = zeros(size(features, 1), 1);
199.
200.      % Predict the currency denomination for each
    feature vector
201.      for i = 1:size(features, 1)
202.          % Initialize an array to store scores
203.          scores = zeros(size(SVMModels, 1), 1);
204.          % Predict the score for each class using
    each SVM model
205.          for j = 1:size(SVMModels, 1)
206.              [~, score] = predict(SVMModels{j},
    features(i, :));
207.              scores(j) = score(2); % Take the score
    for the positive class
208.          end
209.          % Determine the class with the highest
    score
210.          [~, max_score_idx] = max(scores);
211.          % Assign the predicted nominal based on the
    index
212.          predictions(i) = nominals(max_score_idx);
213.      end
```

```
214.
215.       % Display the predicted result
216.       set(handles.edit1, 'string',
   num2str(predictions));
217.
218.
219.       % Update GUI
220.       guidata(hObject,handles);
221.
```

LAMPIRAN 2: Source Code Data Training

```
1.  % Inisialisasi variabel untuk menyimpan fitur dan
    label
2.  Training_Data = [];
3.  Train_Label = [];
4.  file_names = {}; % Variabel untuk menyimpan nama file
5.  file_names_training = {}; % Variabel untuk menyimpan
    nama file data training
6.  file_names_testing = {}; % Variabel untuk menyimpan
    nama file data testing
7.
8.  % Tentukan ukuran dimensi gambar yang diinginkan
9.  desired_height = 400;
10. desired_width = 900;
11.
12. % Tentukan direktori tempat gambar-gambar mata uang
    disimpan
13. parent_dir = 'D:\Kuliah\semester
    7\citrasvmtrain\datasetUang\';
14.
15. % Daftar nominal mata uang
16. nominals = [1000, 2000, 5000, 10000, 20000, 50000,
    100000];
17.
18. % Loop melalui setiap nominal mata uang
19. for nominal_idx = 1:length(nominals)
20.     nominal = nominals(nominal_idx);
21.     % Tentukan direktori tempat gambar-gambar mata
    uang dengan nominal tertentu disimpan
22.     dir_path = fullfile(parent_dir,
    num2str(nominal));
23.     % Loop melalui setiap gambar untuk setiap nominal
    mata uang
24.     for i = 1:100
25.         % Baca gambar
26.         file_name = sprintf('%d_%d.png', nominal, i);
27.         img = imread(fullfile(dir_path, file_name));
28.
```

```
        % Simpan nama file
        file_names = [file_names; {file_name}];

        % Resize gambar ke dimensi yang diinginkan
        img_resized = imresize(img, [desired_height,
desired_width]);

        % Ekstraksi fitur LBP dari gambar
        img_gray = rgb2gray(img_resized);
        features_lbp =
calculateLBPFeatures(img_gray);

        % Hitung mean, entropy, variance, kurtosis,
dan skewness dari fitur LBP
        mean_lbp = mean2(features_lbp);
        entropy_lbp = entropy(features_lbp(:));
        variance_lbp = var(features_lbp(:));
        kurtosis_lbp = kurtosis(features_lbp(:));
        skewness_lbp = skewness(features_lbp(:));

        % Ekstraksi fitur dari gambar menggunakan
nilai HSV
        img_hsv = rgb2hsv(img_resized);
        H = img_hsv(:,:,1);
        S = img_hsv(:,:,2);
        V = img_hsv(:,:,3);

        % Hitung nilai rata-rata dari masing-masing
saluran HSV
        mean_H = mean2(H);
        mean_S = mean2(S);
        mean_V = mean2(V);

        % Gabungkan fitur HSV dan LBP
        features_combined = [mean_H, mean_S, mean_V,
mean_lbp, entropy_lbp, variance_lbp, kurtosis_lbp,
skewness_lbp];

        % Tambahkan fitur ke dalam Training_Data
        Training_Data = [Training_Data;
features_combined];

        % Tambahkan label ke dalam Train_Label
        Train_Label = [Train_Label; nominal];
    end
end

% Memastikan bahwa label dan data sesuai urutannya
assert(size(Training_Data, 1) == size(Train_Label,
1), 'Ukuran data dan label tidak cocok.');

% Pembagian Data Latih dan Data Uji:
```

```
training_ratio = 0.8; % Proporsi data yang akan digunakan untuk pelatihan
num_samples = size(Training_Data, 1);

% Pembagian data secara stratifikasi
cv = cvpartition(Train_Label, 'HoldOut', 1-training_ratio);  % 20% untuk data uji

% Mengambil data latih dan data uji berdasarkan pembagian stratifikasi
Training_Data_Latih = Training_Data(cv.training, :);
Train_Label_Latih = Train_Label(cv.training);
Testing_Data_Uji = Training_Data(cv.test, :);
Test_Label_Uji = Train_Label(cv.test);

% Menyimpan nama file yang masuk ke dalam data training dan testing
train_indices = find(cv.training);
test_indices = find(cv.test);

for i = 1:numel(train_indices)
    file_names_training = [file_names_training; file_names{train_indices(i)}];
end

for i = 1:numel(test_indices)
    file_names_testing = [file_names_testing; file_names{test_indices(i)}];
end

% Menampilkan nama file yang masuk ke dalam data training dan testing
disp('Nama file untuk data training:');
disp(file_names_training);

    disp('Nama file untuk data testing:');
    disp(file_names_testing);

    % Memastikan bahwa setiap kelas memiliki sampel dalam data latih
    unique_labels = unique(Train_Label_Latih);
    for label = unique_labels'
        num_samples_in_class = sum(Train_Label_Latih == label);
        fprintf('Jumlah sampel dalam kelas %d: %d\n', label, num_samples_in_class);
        if num_samples_in_class == 0
            error('Kelas %d tidak memiliki sampel yang sesuai.', label);
        end
    end

```

```
113.      % Simpan data ke dalam file .mat
114.      save('Training_Data.mat',
    'Training_Data_Latih', 'Train_Label_Latih',
    'Testing_Data_Uji', 'Test_Label_Uji',
    'file_names_training', 'file_names_testing');
```

LAMPIRAN 3: Source Code SVM

```
1.  % Muat data yang telah disimpan
2.  data = load('Training_Data.mat');
3.
4.  % Mendapatkan daftar kelas unik
5.  nominals = unique(data.Train_Label_Latih);
6.  num_classes = numel(nominals);
7.
8.  % Inisialisasi seluruh model SVM
9.  SVMModels = cell(num_classes, 1);
10.
11. % Melatih model SVM untuk setiap kelas menggunakan
    metode one-vs-all
12. for i = 1:num_classes
13.     % Mengambil label untuk kelas saat ini
14.     binary_labels = (data.Train_Label_Latih ==
    nominals(i));
15.
16.     % Melatih model SVM
17.     SVMModels{i} = fitcsvm(data.Training_Data_Latih,
    binary_labels, 'KernelFunction', 'rbf',
    'Standardize', true);
18.
19.     % Tampilkan jumlah sampel dalam kelas saat ini
    dan lainnya
20.     disp(['Jumlah sampel dalam kelas ',
    num2str(nominals(i)), ': ',
    num2str(sum(binary_labels))]);
21.     disp(['Jumlah sampel di luar kelas ',
    num2str(nominals(i)), ': ',
    num2str(sum(~binary_labels))]);
22. end
23.
24. % Simpan model SVM yang telah dilatih ke dalam file
    .mat
25. save('trainedSVMModel.mat', 'SVMModels', 'nominals');
```

LAMPIRAN 4: Source Code Prediksi SVM

```
1.  % predict_svm_model.m
```

```
% Muat model SVM yang telah dilatih
load('trainedSVMModel.mat');

% Muat data yang telah disimpan (untuk mendapatkan
data uji)
data = load('Training_Data.mat');

% Data uji yang akan digunakan untuk prediksi
Testing_Data_Uji = data.Testing_Data_Uji;

% Inisialisasi array untuk menyimpan skor keputusan
num_classes = numel(nominals);
scores = zeros(size(Testing_Data_Uji, 1),
num_classes);

% Hitung skor keputusan untuk setiap kelas
for i = 1:num_classes
    [~, score] = predict(SVMModels{i},
Testing_Data_Uji);
    scores(:, i) = score(:, 2); % Ambil skor untuk
kelas positif
end

% Tentukan kelas dengan skor tertinggi
[~, max_score_idx] = max(scores, [], 2);
predicted_labels = nominals(max_score_idx);

% Evaluasi akurasi
true_labels = data.Test_Label_Uji;

% Hitung Confusion Matrix
confusionMat = confusionmat(true_labels,
predicted_labels);

% Hitung Akurasi dari Confusion Matrix
accuracy = sum(diag(confusionMat)) /
sum(confusionMat(:));

% Hitung Precision, Recall, dan F1-Score untuk setiap
kelas
precision = zeros(num_classes, 1);
recall = zeros(num_classes, 1);
f1score = zeros(num_classes, 1);

for i = 1:num_classes
    TP = confusionMat(i, i); % True Positives
    FP = sum(confusionMat(:, i)) - TP; % False
Positives
    FN = sum(confusionMat(i, :)) - TP; % False
Negatives

    precision(i) = TP / (TP + FP);
```

```
    recall(i) = TP / (TP + FN);
    f1score(i) = 2 * (precision(i) * recall(i)) / (precision(i) + recall(i));
end

% Hitung rata-rata precision, recall, dan F1-score (macro-average)
avg_precision = mean(precision);
avg_recall = mean(recall);
avg_f1score = mean(f1score);

% Tampilkan Confusion Matrix dan Akurasi
disp('Confusion Matrix:');
disp(confusionMat);

disp(['Akurasi prediksi: ', num2str(accuracy)]);

% Tampilkan Precision, Recall, dan F1-Score untuk setiap kelas
for i = 1:num_classes
    disp(['Kelas ', num2str(nominals(i)), ':']);
    disp(['Precision: ', num2str(precision(i))]);
    disp(['Recall: ', num2str(recall(i))]);
    disp(['F1-Score: ', num2str(f1score(i))]);
end

% Tampilkan rata-rata Precision, Recall, dan F1-Score
disp('Rata-rata Precision, Recall, dan F1-Score:');
disp(['Precision: ', num2str(avg_precision)]);
disp(['Recall: ', num2str(avg_recall)]);
      disp(['F1-Score: ', num2str(avg_f1score)]);
```