



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1. Metodologi Penelitian

Metode Penelitian yang digunakan dalam identifikasi tulisan tangan *offline* dengan menggunakan algoritma *Gray Level Co-occurrence Matrix* dan *Support Vector Machine* adalah sebagai berikut.

1. Studi Literatur

Pada tahap ini, akan dilakukan pencarian dan juga pembelajaran lebih mendalam mengenai konsep yang relevan dengan permasalahan, algoritma, dan bahasa pemrograman yang dipakai dalam penelitian.

2. Pengumpulan Data

Pengumpulan data yang akan dilakukan dengan cara memberikan kepada beberapa orang untuk menuliskan 20 kata sebagai dataset. Satu orang akan menuliskan sebanyak 10 kali. Data tersebut akan dikumpulkan, kemudian akan di-*scan* menggunakan *scanner* untuk kemudian dilakukan *image processing*. Total data yang telah dikumpulkan adalah 250 data dari 25 orang.

3. Perancangan Penelitian

Pada tahapan ini, akan dirancang bagaimana proses identifikasi tulisan tangan seseorang menggunakan algoritma *Gray Level Co-occurrence Matrix* dan *Support Vector Machine* menggunakan *flowchart*.

4. Pembuatan Program

Melakukan penelitian serta penulisan kode program yang sesuai dengan perancangan penelitian yang telah dirancang sebelumnya. Pembuatan program menggunakan bahasa pemrograman Python dan pembuatan *website* menggunakan web *framework* Flask. Dengan didukung dengan hardware dan software yang dibutuhkan.

5. Uji Coba & Evaluasi

Pada tahap ini, akan dilakukan uji coba pada fitur ekstraksi ciri untuk algoritma *Gray Level Co-occurrence Matrix* (GLCM) dengan sudut yang berbeda-beda, yaitu 0° , 45° , 90° , dan 135° . Serta untuk algoritma *Histogram of Oriented Gradients* (HOG) untuk menambahkan proses *preprocessing*, sebagai fitur deskriptor. Algoritma *Support Vector Machine* (SVM) menggunakan model linear sebagai *decision boundary*. Uji coba ini dilakukan agar mendapatkan nilai dari *recall*, *accuracy*, *precision*, dan *f1-score*. Pada tahap evaluasi ini, program yang telah dirancang akan menggunakan *confusion matrix*, serta akan dilakukan proses *testing*. Jika terdapat masalah pada program, maka akan dilakukan perbaikan hingga permasalahan tersebut terselesaikan dengan baik.

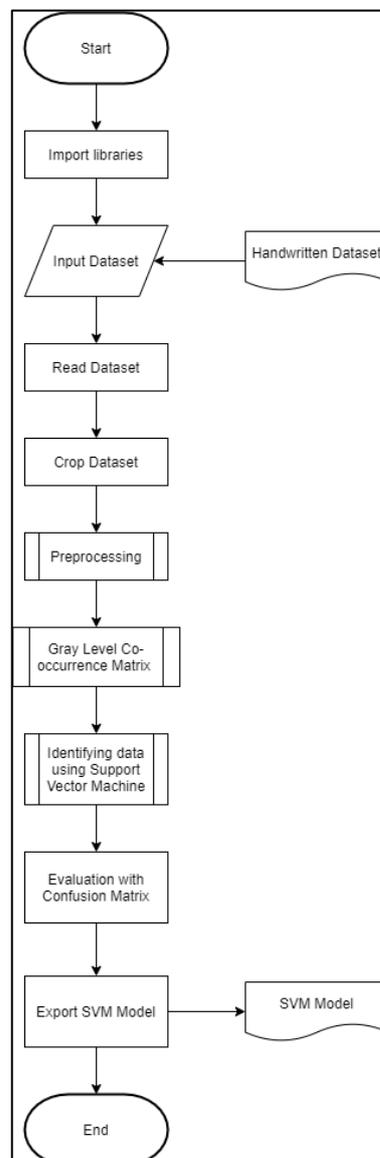
6. Penulisan Skripsi

Melalui tahapan yang terakhir ini, akan dilakukan penyusunan laporan skripsi sesuai dengan hasil penelitian yang ada. Laporan ini tersusun dari perancangan, pembuatan, hasil penelitian, serta dokumentasi selama penelitian berlangsung.

3.2. Perancangan Sistem

Perancangan sistem terdiri dari *flowchart*, perancangan struktur tabel, dan rancangan tampilan antarmuka pada aplikasi web. Perancangan *Flowchart* ini terbagi menjadi 2 bagian, yaitu *flowchart* proses identifikasi yang menggunakan algoritma *Support Vector Machine*, dan *flowchart* yang menggambarkan proses penggunaan algoritma *Support Vector Machine* ke dalam aplikasi web.

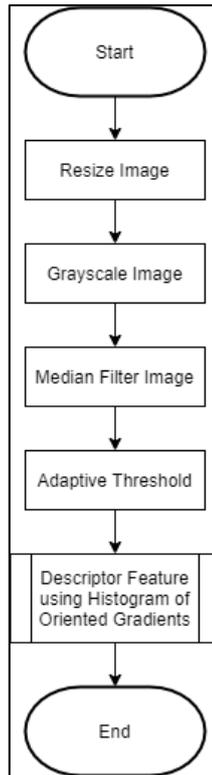
3.2.1. Flowchart Proses Identifikasi



Gambar 3.1 *Flowchart* penelitian secara umum

Pada Gambar 3.1 menggambarkan alur dari proses identifikasi tulisan tangan menggunakan algoritma *Gray Level Co-occurrence Matrix* dan *Support Vector Matrix* secara umum. Sebelum memulai proses identifikasi tersebut, yang pertama kali harus dilakukan adalah meng-*import* terlebih dahulu *library* yang dibutuhkan. Setelah *library* di-*import*, kemudian memasukkan *dataset* yang digunakan dalam penelitian ini.

Dataset berupa gambar *scan* tulisan tangan yang telah dikumpulkan dari 25 orang. *Dataset* tersebut di-*crop* menjadi 10 gambar dengan ukuran kurang lebih 1811 x 180 piksel. Setelah itu, proses akan dilanjutkan dengan *preprocessing* pada *dataset* yang dapat dilihat pada Gambar 3.2. Selanjutnya, proses ekstraksi fitur pada *dataset* yang dilakukan menggunakan algoritma *Gray Level Co-occurrence Matrix* (GLCM) yang dapat dilihat pada Gambar 3.4. Kemudian, akan dilanjutkan ke proses identifikasi *dataset* dengan menggunakan algoritma *Support Vector Machine* (SVM) yang dapat dilihat pada Gambar 3.5. Setelah semua proses tersebut, maka akan dilakukan evaluasi terhadap *dataset* menggunakan *Confusion Matrix*. Pada *Confusion Matrix* ini, terdiri dari *confusion matrix* itu sendiri serta *classification report*. Untuk memudahkan dalam proses penelitian, maka model SVM tersebut di *export* ke dalam bentuk dokumen model.pkl.

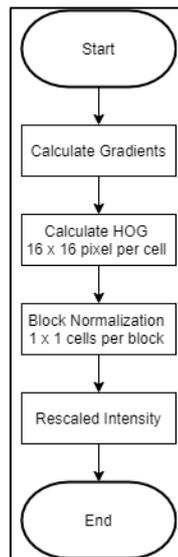


Gambar 3.2 *Flowchart Preprocessing*

Pada Gambar 3.2 ini merupakan *flowchart preprocessing* yang berisi 5 proses di dalamnya, antara lain *resize image*, *grayscale image*, *median filter image*, *adaptive threshold*, dan fitur deskriptor menggunakan algoritma *Histogram of Oriented Gradients*. Yang pertama proses *resize image*, proses ini dilakukan agar semua ukuran gambar sama, yaitu 1800 x 170 piksel. Proses *grayscale image* adalah proses dimana gambar tersebut akan berwarna antara hitam dan putih.

Kemudian proses yang ketiga, *median filter image* ini bertujuan agar menghapus *noise* yang mengganggu dalam proses *preprocessing* yang selanjutnya. *Median filter image* akan menghitung intensitas piksel dengan filter 5 x 5 piksel. Selanjutnya, *Adaptive Threshold* digunakan untuk mendapatkan *threshold* yang berbeda dalam wilayah berbeda dari gambar yang sama, sehingga memberikan hasil yang lebih baik dengan pencahayaan yang beragam. Dengan menggunakan

parameter `cv2.ADAPTIVE_THRESH_GAUSSIAN_C` ini, nilai *threshold* sama dengan jumlah *gaussian-weighted* yang dikurangi dengan konstanta *C*. Dan proses yang terakhir dalam *preprocessing* ini adalah fitur deskriptor menggunakan algoritma *Histogram of Oriented Gradients* yang dapat dilihat pada Gambar 3.3.

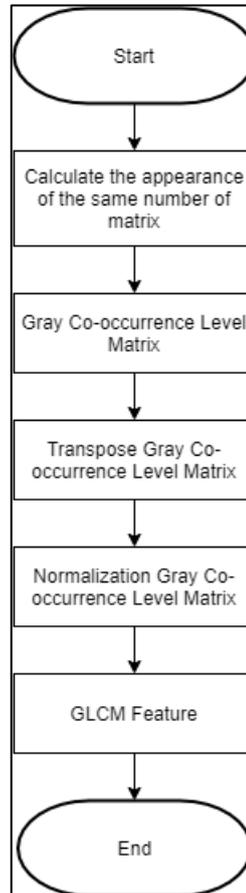


Gambar 3.3 *Flowchart* fitur deskriptor *Histogram of Oriented Gradients*

Pada Gambar 3.3 merupakan alur dari proses fitur deskriptor *Histogram of Oriented Gradients*. Algoritma *Histogram of Oriented Gradients* ini digunakan sebagai fitur deskriptor dalam penelitian ini. Pertama-tama, dengan menghitung *gradient horizontal* dan *gradient vertical* untuk dapat mengurangi bagian-bagian yang tidak penting, seperti latar belakang yang berwarna konstan. Serta akan menebalkan garis yang dibutuhkan. Perhitungan *gradient* ini telah dihitung dalam fungsi *hog()* yang dipakai dalam program.

Setelah itu, gambar akan dibagi menjadi 16×16 *pixel per cell* dan *gradient histogram* tersebut juga akan dihitung setiap 16×16 *pixel per cell*. Dan 1×1 *cell per block*-nya akan dihitung untuk setiap *block normalization*. Terakhir, gambar di-

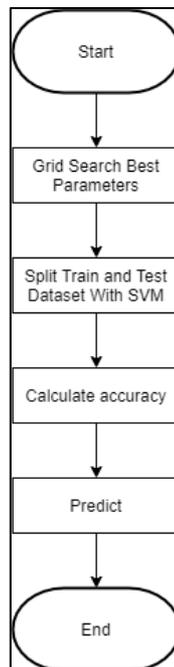
rescale intensity dengan rentang intensitas 0 sampai 10 agar gambar terlihat lebih bagus.



Gambar 3.4 *Flowchart Gray Level Co-occurrence Matrix*

Pada Gambar 3.4 menggambarkan alur dari proses *Gray Level Co-occurrence Matrix*. Algoritma *Gray Level Co-occurrence Matrix* (GLCM) ini digunakan sebagai ekstraksi fitur dalam penelitian ini. Ekstraksi fitur dimulai dengan menghitung kemunculan jumlah matriks yang sama, sehingga terbentuklah *gray co-occurrence level matrix* melalui fungsi *greycomatrix()*. Setelah *gray co-occurrence level matrix* terbentuk, matriks tersebut kemudian di-*transpose* dan dinormalisasi. *Output* yang dihasilkan ini merupakan nilai dari *histogram grayscale* yang secara bersamaan pada *offset* yang ada pada gambar.

Selanjutnya, fitur-fitur yang ada pada GLCM tersebut akan dihitung. Fitur-fitur GLCM tersebut ialah *contrast*, *dissimilarity*, *homogeneity*, *energy*, dan *correlation*. Pada setiap fitur GLCM memiliki 1 nilai dari setiap *angle* yang berbeda-beda. *Angle* tersebut terdiri dari 0° , 45° , 90° , dan 135° . *Output* dari setiap fitur GLCM inilah yang akan digabung dan kemudian akan digunakan dalam *training testing dataset*.



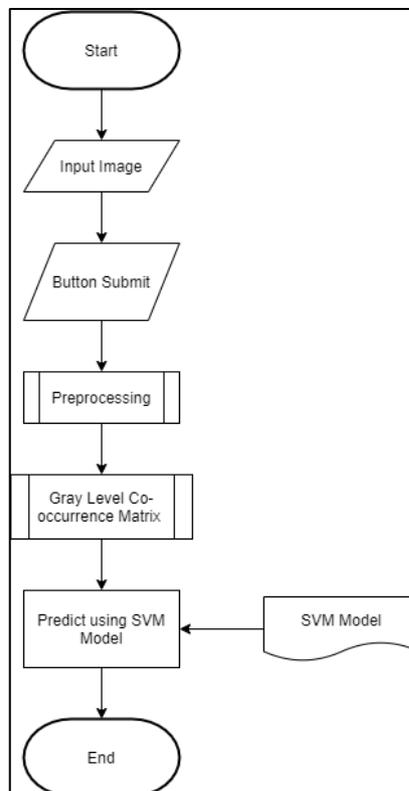
Gambar 3.5 *Flowchart Identifying data using Support Vector Machine*

Gambar 3.5 ini menunjukkan alur dari proses identifikasi menggunakan algoritma *Support Vector Machine* (SVM). Pertama-tama, menggunakan *Grid Search Best Parameters* untuk menemukan *parameter* algoritma SVM yang tepat, sehingga mendapatkan hasil akurasi yang paling bagus. *Parameter* untuk SVM, antara lain *gamma*, *C*, dan *kernel*. Untuk *parameter gamma* dan *C* terdiri dari 0.001, 0.01, 0.1, 1, 10, dan 100. Sedangkan untuk *parameter kernel* terdiri dari *linear* dan *rbf*.

Setelah menemukan *best parameters*, *dataset* tersebut akan di-*split* untuk data *training* dan data *testing* menggunakan *train test split*. Pembagian data *training* dan data *testing* juga dilakukan dengan perbandingan yang berbeda-beda, agar mendapatkan hasil akurasi yang tinggi. Perbandingan antara data *training* dan data *testing* 70:30, 75:25, dan 80:20. Dan yang terakhir, menghitung akurasi dari data *training* dan data *testing* serta memprediksi *dataset* sesuai dengan data target yaitu nama pemilik tulisan tangan tersebut.

3.2.2. Flowchart Aplikasi Web

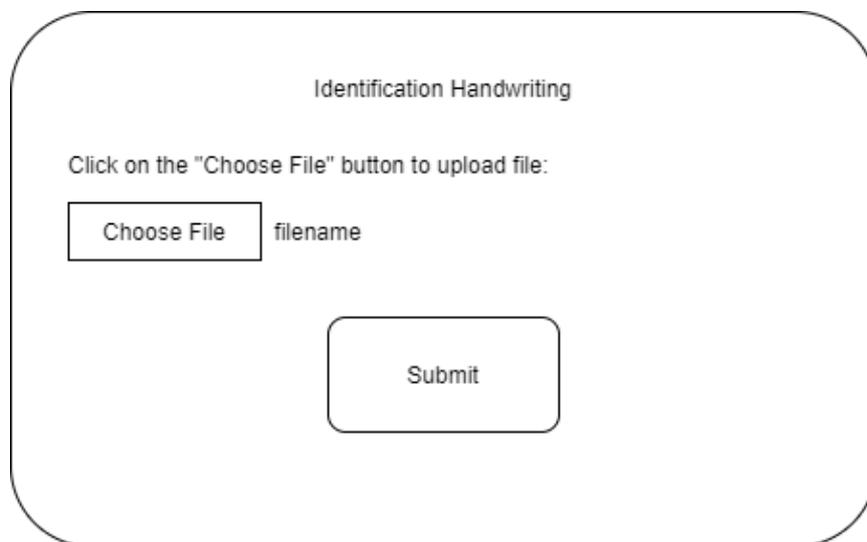
Proses identifikasi tulisan tangan menggunakan algoritma *Histogram of Oriented Gradients* (HOG) dan *Support Vector Machine* (SVM) dapat dipahami lebih mudah, maka dibuatlah sebuah *web* yang mengimplementasikan proses tersebut.



Gambar 3.6 *Flowchart Web Development*

Pada Gambar 3.6 ini merupakan *flowchart web development*. Pertama-tama, *user* meng-*upload* gambar dan kemudian menekan tombol *Submit*. Kemudian, *web* akan menjalankan proses *preprocessing* yang terdapat pada Gambar 3.2 dan proses *extraction feature* menggunakan GLCM yang terdapat pada Gambar 3.4. Proses yang terakhir adalah proses *predict* menggunakan model SVM yang telah di-*export* sebelumnya.

3.2.3. Rancangan Tampilan Antarmuka



The image shows a web interface titled "Identification Handwriting". It contains a text prompt: "Click on the 'Choose File' button to upload file:". Below this prompt is a "Choose File" button followed by a text input field labeled "filename". At the bottom center of the interface is a "Submit" button.

Gambar 3.7 Halaman Utama Aplikasi

Tampilan Antarmuka aplikasi web terbagi menjadi dua halaman, yaitu halaman utama dan halaman hasil *upload*. Pada Gambar 3.7 merupakan rancangan antarmuka untuk halaman utama. Pada halaman utama inilah, *user* dapat memasukkan gambar tulisan tangan, yang kemudian nantinya akan diprediksi siapa pemilik tulisan tangan tersebut.

Identification Handwriting

Click on the "Choose File" button to upload file:

filename

Image

Writer's Name:

Accuracy:		Precision:
Recall:		F1-Score:

Gambar 3.8 Halaman Hasil *Upload*

Pada Gambar 3.8 merupakan rancangan antarmuka pada halaman hasil *upload*. Pada halaman ini, *user* dapat melihat gambar yang telah di *upload* serta hasil proses penelitian. Hasil penelitian tersebut terdiri dari nama penulis, nilai *accuracy*, *recall*, *precision*, dan *f1-score*. Serta terdapat tombol *Back* untuk dapat kembali ke halaman utama aplikasi.