



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, 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 APLIKASI

3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan untuk merealisasikan tujuan penelitian ini adalah sebagai berikut.

1. Studi Literatur

Dilakukan studi literatur tentang grafologi, karakter berdasarkan jenis tulisan tangan, *image processing*, segmentasi, projeksi vertikal dan horizontal, deslanting, support vector machine, RBF kernel, K fold cross validation dan standar deviasi.

2. Analisis Kebutuhan

Dilakukan analisis terhadap kebutuhan-kebutuhan yang diperlukan untuk menunjang penelitian, seperti spesifikasi *software*.

3. Interview

Interview dilakukan terhadap Dr. Margaretha Purwanti, M.Si., Psikolog. Beliau memiliki pengalaman menganalisa tulisan tangan untuk seleksi masuk mahasiswa jalur bebas test.

4. Pengumpulan Sampel

Dilakukan pengumpulan sampel berupa tulisan tangan untuk proses *learning* yang akan dilakukan. Sampel dikumpulkan sebanyak 300 tulisan tangan dengan meminta 100 orang untuk menulis di atas selembar kertas HVS dimana masing-masing orang menulis sebanyak 3 lembar.

5. Perancangan Sistem

Perancangan sistem dimulai dari merancang model aplikasi untuk menunjukkan gambaran umum sistem yang akan dibangun. Dilanjutkan dengan perancangan *flowchart* untuk menunjukkan proses yang terjadi pada sistem. Database dirancang untuk menyimpan data pengguna aplikasi. Dimulai dari tahap pembuatan desain sampai dengan membuat program aplikasi.

6. Implementasi

Implementasi aplikasi identifikasi karakter berdasarkan tulisan tangan ini berbasis web. Digunakan bahasa *python* untuk mengimplementasikan *image* processing, feature extraction, dan support vector machine serta framework Django sebagai penghubung antara user interface, proses pada sistem, dan database. Data pada sistem disimpan pada database MySQL.

7. Pengujian Aplikasi dan Debug

Dilakukan pengujian terhadap tingkat akurasi, apakah aplikasi dapat berjalan dengan baik, dan *debugging* jika terdapat kesalahan pada *source code*. Pengukuran evaluasi aplikasi diukur dengan cara mencari nilai *accuracy*. Hasil analisis dari aplikasi yang telah dilatih akan dicocokkan secara manual sesuai dengan parameter yang diuji. Parameter tersebut adalah ukuran huruf, kemiringan kata, *baseline*, tekanan penulisan, spasi antar kata, ukuran margin atas, margin kiri dan margin kanan.

NUSANTARA

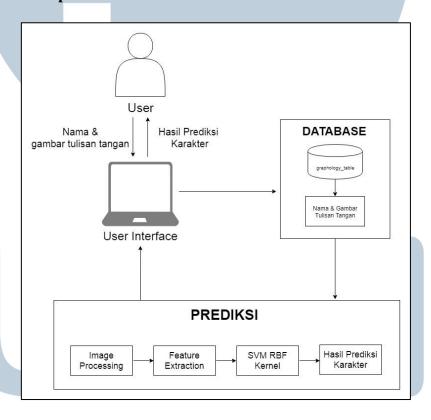
8. Konsultasi dan penulisan

Dilakukan konsultasi terhadap dosen pembimbing yang dilanjutkan dengan penulisan laporan.

3.2 Perancangan Aplikasi

Aplikasi dirancang dengan menggunakan model aplikasi, *flowchart*, struktur tabel, dan desain antarmuka.

3.2.1 Model Aplikasi



Gambar 3.1 Model Aplikasi Identifikasi Karakter Berdasarkan Tulisan Tangan

Komponen model aplikasi pada Gambar 3.1 dapat dijabarkan sebagai berikut.

NUSANTARA

1. User

User memasukkan nama dan gambar tulisan tangan digital yang telah di-*scan* sebelumnya menggunakan printer *Canon MP 287* dengan *flatbed scanner*, format warna hitam putih dan resolusi 200 dpi.

2. User Interface

User Interface berfungsi untuk menampilkan tampilan antarmuka aplikasi identifikasi karakter berdasarkan tulisan tangan. User berinteraksi dengan sistem melalui *user interface*.

3. Database

Database berfungsi untuk menyimpan informasi nama dan gambar tulisan tangan.

4. Prediksi

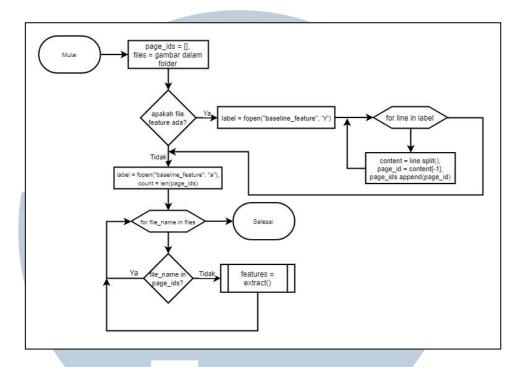
Prediksi merupakan proses yang menghasilkan karakter dari tulisan tangan yang di-input oleh *user*. Proses yang terjadi adalah *image processing*, untuk memproses gambar agar dapat diambil *feature* yang dibutuhkan. *Feature extraction* adalah proses mengambil *feature* yang dijadikan sebagai *data train* dan *data test* pada *support vector machine*. *Support vector machine* yang telah dilatih dapat digunakan untuk memprediksi karakter dari gambar tulisan tangan.

3.2.2 Flowchart

Berikut ini merupakan *flowchart* dari aplikasi identifikasi karakter berdasarkan tulisan tangan.

A. Flowchart extractFeature

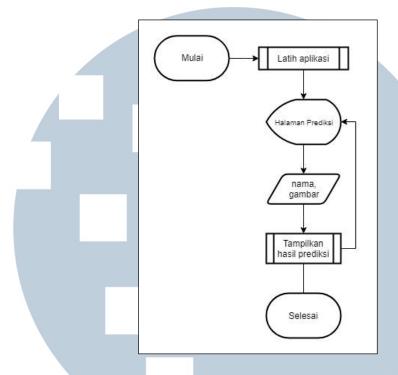
Sebelum menjalankan aplikasi, user harus generate file dataset terlebih dahulu. File dataset dibagi menjadi yaitu topMargin feature, leftMargin_feature, rightMargin_feature, baseline_feature, letterSize_feature, lineSpacing_feature dan slantAngle_feature. Tiap file mengandung feature vector dan label sesuai ilmu grafologi yang akan dijadikan input kedalam support vector mahine. Pembentukan dataset dijalankan secara terpisah dengan aplikasi. Flowchart extractFeature merupakan proses untuk mendapatkan feature vector dari gambar tulisan tangan yang telah didapatkan. files merupakan seluruh gambar tulisan tangan dalam folder. Seluruh gambar yang telah tersimpan dalam variable files akan di-extract untuk didapatkan feature vector. File yang berisi feature vector pada setiap parameter akan terbentuk dalam proses ini. Gambar 3.2 merupakan flowchart extractFeature.



Gambar 3.2 Flowchart extractFeature

B. Flowchart Aplikasi

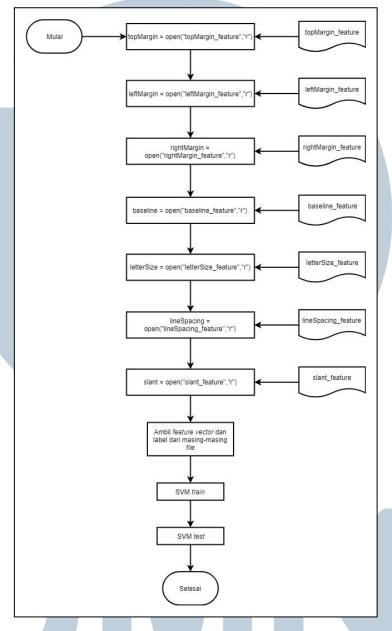
Pada saat menjalankan aplikasi, akan ditampilkan halaman prediksi. User dapat memasukkan *input* berupa nama, dan gambar tulisan tangan yang ingin dianalisa. Kemudian akan muncul hasil analisa setelah nama dan gambar tulisan tangan di-*submit*. Analisis berupa karakter dari pemilik gambar tulisan tangan berdasarkan margin atas, margin kiri, margin kanan, *baseline*, ukuran huruf, jarak antar baris, dan kemiringan tulisan. Gambar 3.3 menunjukkan *flowchart* aplikasi.



Gambar 3.3 Flowchart Aplikasi

C. Flowchart Latih Aplikasi

Pada saat menjalankan aplikasi pertama kali dijalankan, maka aplikasi akan dilatih menggunakan support vector machine. Feature vector yang akan dijadikan input ke dalam support vector machine berasal dari file yang telah dibentuk sejak awal. Kemudian untuk margin atas, margin kiri, margin kanan, baseline, ukuran huruf, jarak antar huruf dan kemiringan tulisan akan dilatih masing-masing. Gambar 3.4 merupakan gambar Flowchart Latih Aplikasi.

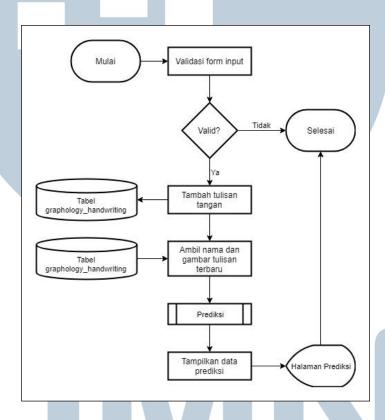


Gambar 3.4 Flowchart Latih Aplikasi

C. Flowchart Tampilkan hasil prediksi

Nama dan gambar tulisan tangan yang telah di-*input* oleh *user* akan ditambahkan ke dalam database. Kemudian nama dan gambar yang telah di-*input user* akan diprediksi. Hasil prediksi kemudian akan ditampilkan pada halaman

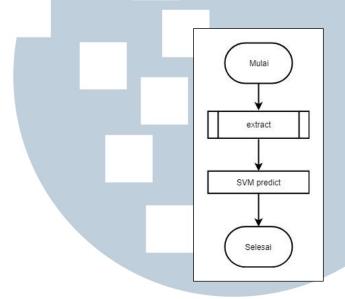
yang sama. Sistem aplikasi menggunakan *single page application* sehingga tidak berpindah halaman. Apabila *user* ingin melakukan prediksi lagi dengan gambar yang berbeda, *user* dapat mengi-*input* ulang nama dan gambar tulisan tangan pada halaman yang sama. Gambar 3.5 menunjukkan *flowchart* tampilan halaman prediksi.



Gambar 3.5 Flowchart Tampilkan hasil prediksi

C. Flowchart Prediksi

Gambar hasil *input user* akan di-*extract* supaya didapatkan *feature vector* yang akan dijadikan input ke dalam *support vector machine* untuk diprediksi. Gambar 3.6 menunjukkan *flowchart* prediksi.

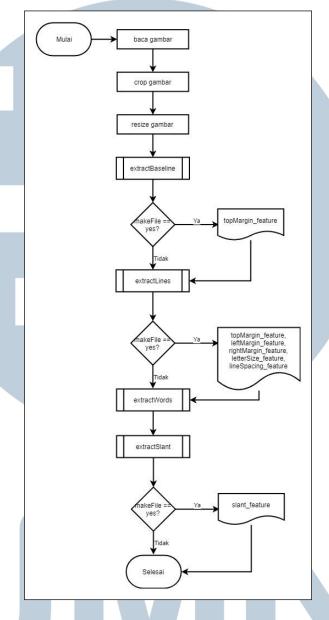


Gambar 3.6 Flowchart Prediksi

D. Flowchart extract

Dilakukan *crop* kemudian *resize* terhadap gambar yang didapat dari *input user*. Tujuan *crop* dan *resize* ini adalah agar pada saat mengekstrak *feature* didapat hasil yang konsisten. extractBaseline, extractLines, extractWords, extractSlant dilakukan untuk didapat *feature* yang akan dijadikan input ke dalam *support vector machine*. Bila dijalankan *extractFeature*, maka variabel *makeFile* akan menjadi *yes*. Variabel *makeFile* menentukan apakah akan dibentuk *file* untuk menyimpan *feature* atau tidak. Gambar 3.7 menunjukkan *Flowchart extract*.

NUSANTARA

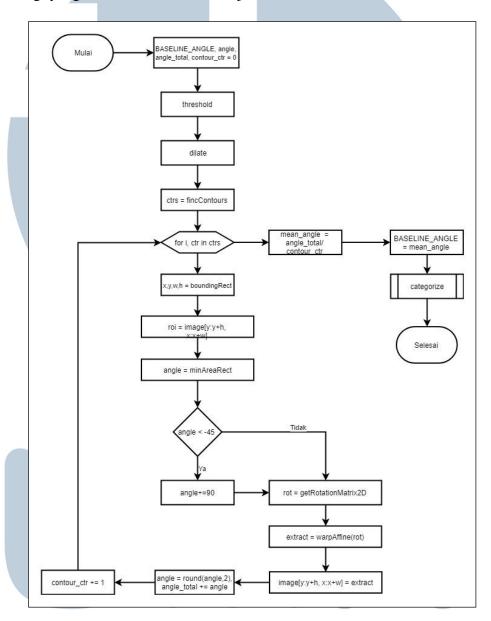


Gambar 3.7 Flowchart extract

E. Flowchart extractBaseline

BASELINE_ANGLE, angle, angle_total, dan contour_ctr diinisialisasi. Dilakukan image processing berupa threshold dan dilation terhadap gambar tulisan tangan. Angle didapat dari minAreaRect. Tiap baris tulisan tangan akan dijumlahkan angle untuk mendapatkan angle_total dan contour_ctr. BASELINE_ANGLE akan diassign dari nilai angle_total dibagi dengan contour-

_ctr. BASELINE_ANGLE akan dikategorikan dengan categorize sesuai teori grafologi yang ada. Gambar 3.8 menunjukkan Flowchart extractBaseline.

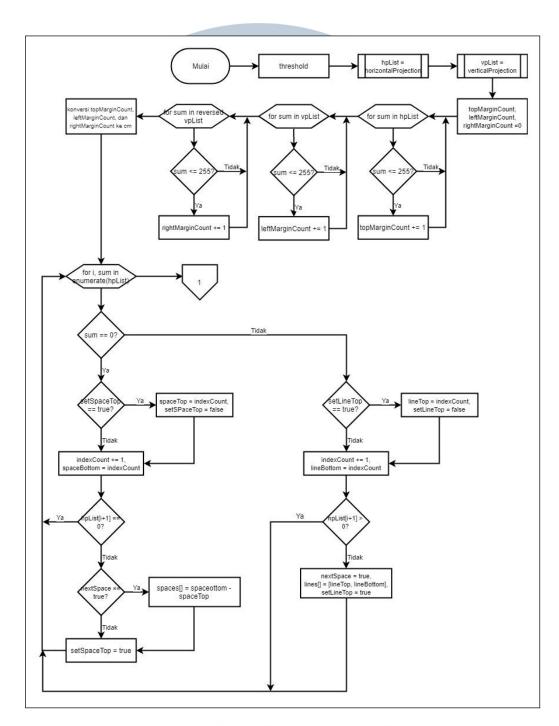


Gambar 3.8 Flowchart extractBaseline
UNIVERSITAS
MULTIMEDIA
NUSANTARA

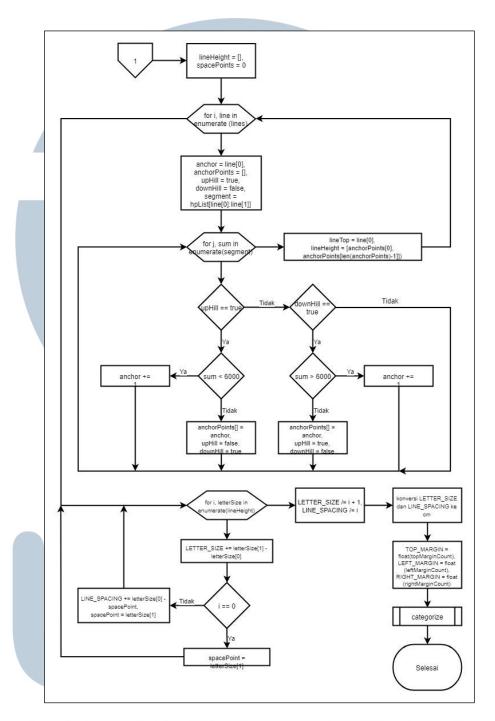
F. Flowchart extractLines

Dilakukan image processing berupa threshold. horizontalProjection dan verticalProjection dilakukan untuk mendapatkan proyeksi pixel gambar secara horizontal dan vertikal. Bila jumlah pixel pada horizontal Projection lebih kecil dari maka topMarginCount ditambah 1. Demikian juga bila verticalProjection dan reverse verticalProjection lebih kecil dari 255 maka leftMarginCount dan rightMarginCount ditambah 1. Kemudian dilakukan pengulangan lagi terhadap horizontalProjection. Bila jumlah pixel pada horizontalProjection lebih besar dari 0 maka spaceTop, spaceBottom, lineTop, dan *lineBottom* akan ditetapkan. *spaceTop* adalah titik awal dari jarak antar baris dalam satu baris dan spaceBottom adalah titik akhir sedangkan lineTop adalah titik awal dari huruf dalam satu baris dan lineBottom adalah titik akhir. spaces merupakan array yang menampung spaceTop dan spaceBottom sedangkan lines merupakan array yang menampung *lineTop* dan *lineBottom*. Kemudian dilakukan pengulangan terhadap *lines* untuk dilakukan identifikasi apakah dalam satu baris tersebut merupakan tulisan tangan atau hanya coretan kecil. Bila teridentifikasi tulisan tangan, maka akan disimpan dalam array lineHeight. LETTER_SIZE dan LINE_SPACING akan didapatkan dari lineHeight. Akan dilakukan pengelompokkan sesuai dengan teori grafologi terhadap topMarginCount, leftMarginCount, rightMarginCount, LETTER_SIZE dan LINE_SPACING dengan categorize. Gambar 3.9 dan Gambar 3.10 menunjukkan flowchart extractLines.

NUSANTARA



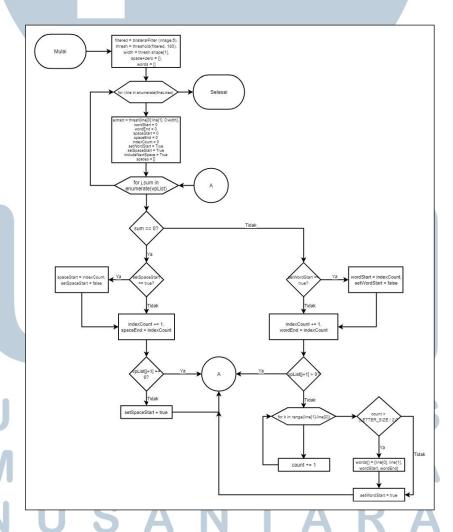
Gambar 3.9 Flowchart extractLines 1 A S MULTIMEDIA NUSANTARA



Gambar 3.10 Flowchart extractLines 2 MULTIMEDIA NUSANTARA

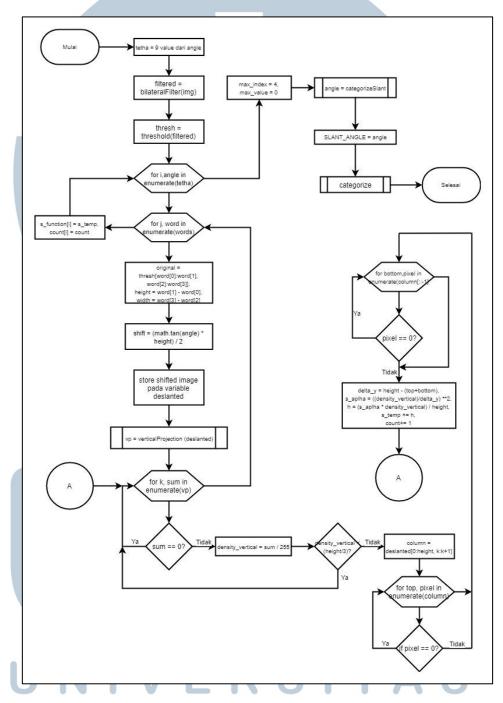
G. Flowchart extractWords

Pada *extractWords* akan dicari koordinat y1, y2, x1, dan x2 dari tiap kata. Setiap baris tulisan akan dicari titik awal dan titik akhir secara vertikal dari tiap kata. *spaceStart* merupakan titik awal kata dan *spaceBottom* merupakan titik akhir kata. Bila tinggi kata lebih kecil dari setengah ukuran huruf, maka akan diabaikan dan tidak dianggap sebagai kata. Variabel *words* merupakan array yang menampung koordinat dari y1,y2,x1, dan x2 dari tiap kata. Gambar 3.11 merupakan *flowchart extractWords*.



Gambar 3.11 Flowchart extractWords

H. Flowchart extractSlant



Gambar 3.12 Flowchart extractSlant

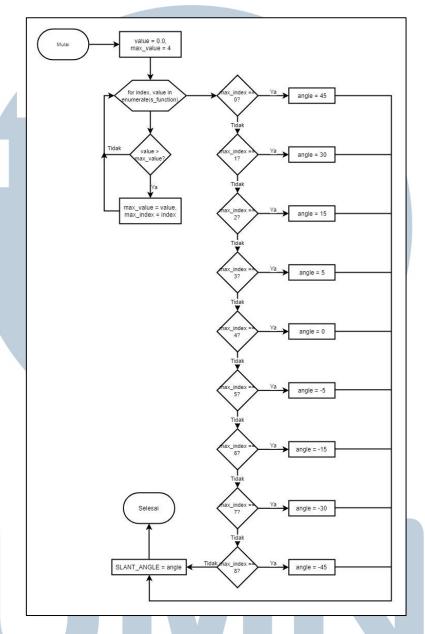
SANTARA

Pada *extractSlant* akan dicari kemiringan kemiringan huruf. Menurut Vinciarelli dan Luettin (2010) kemiringan huruf didapatkan dengan densitas vertikal (jumlah *foreground pixel* per kolom) dibagi dengan jarak antara pixel tertinggi dan pixel terendah dalam satu kolom, dimana kolom adalah huruf dalam satu kata kemudian hasil pembagian tersebut dikuadratkan. Kemudian kemiringan huruf yang telah didapat akan dikategorikan. Hasil kemiringan huruf yang telah dikategorikan akan diberi label dengan *categorize*. Pemberian label sesuai dengan teori grafologi. Gambar 3.12 merupakan *Flowchart extractSlant*.

I. Flowchart categorizeSlant

Flowchart categorizeSlant berfungsi untuk mengategorikan kemiringan huruf agar didapatkan sudut dari tiap-tiap kemiringan huruf. Bila angle bukan 0 berarti terdapat kemiringan huruf. Gambar 3.13 merupakan flowchart categorizeSlant.

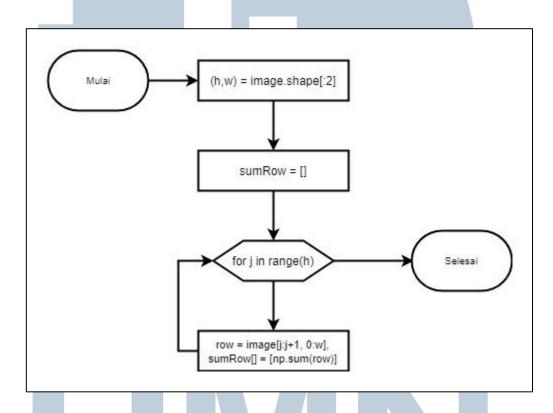




Gambar 3.13 Flowchart categorizeSlant

J. Flowchart horizontal Projection

Flowchart horizontalProjection berfungsi untuk menjumlahkan seluruh nilai pixel pada tiap baris dari gambar. Gambar 3.14 merupakan flowchart horizontalProjection.

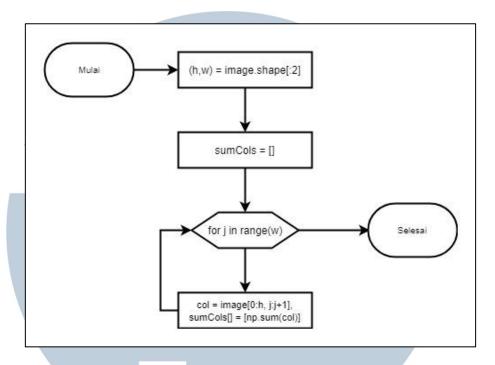


Gambar 3.14 Flowchart horizontal Projection

K. Flowchart verticalProjection

Flowchart verticalProjection berfungsi untuk menjumlahkan seluruh nilai pixel pada tiap kolom dari gambar. Gambar 3.15 merupakan flowchart verticalProjection.

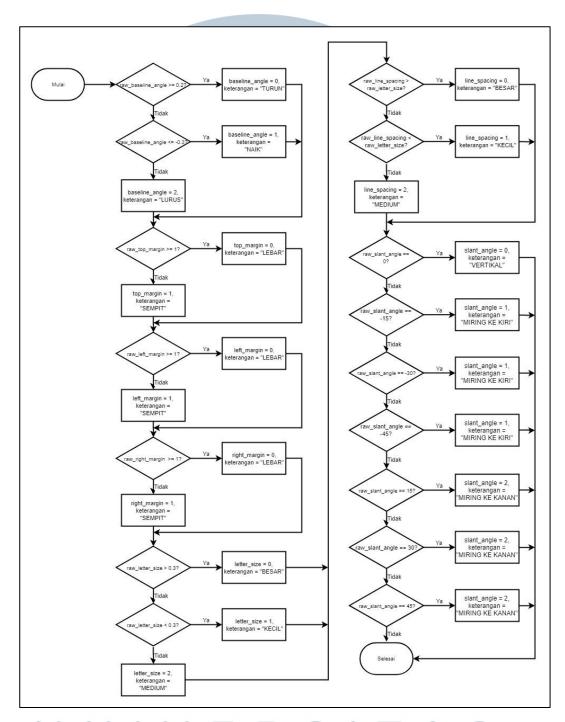
M U L T I M E D I A N U S A N T A R A



Gambar 3.15 Flowchart vertical Projection

L. Flowchart categorize

Raw feature yang telah didapat akan diberi label berdasarkan kategori pada masing-masing parameter. Margin atas, margin kiri, margin kanan, baseline, ukuran huruf, jarak antar baris, dan kemiringan huruf yang telah didapat akan dikategorikan sesuai ilmu grafologi. Gambar 3.16 merupakan flowchart categorize.



Gambar 3.16 Flowchart categorize AS

MULTIMEDIA

NUSANTARA

3.2.3 Struktur Tabel

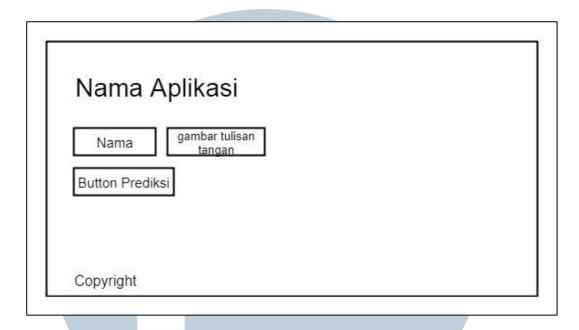
Aplikasi identifikasi karakter berdasarkan tulisan tangan memiliki 1 buah tabel. Tabel 3.1 merupakan tabel Struktur Tabel *graphology_handwriting* yang berfungsi menyimpan data *user*.

Tabel 3.1 Struktur Tabel graphology_handwriting

No.	Nama Kolom	Tipe	Panjang	Keterangan
1	id	int	11	Primary key
2	name	varchar	50	Nama user
3	handwriting_img	varchar	250	Gambar tulisan tangan

3.2.4 Desain Antarmuka

Dalam menggunakan aplikasi identifikasi karakter berdasarkan tulisan tangan, *user* akan diarahkan pada halaman utama. Gambar 3.17 merupakan Desain Halaman Utama. *User* akan memasukkan nama dan meng-*upload* gambar tulisan tangan yang telah di-*scan* sebelumnya. Setelah itu *user* dapat memprediksi karakter dengan mengklik tombol prediksi.



Gambar 3.17 Desain Halaman Utama

Setelah tombol prediksi diklik, maka *user* akan mendapatkan hasil prediksi karakter. Terdapat tujuh kategori karakter yaitu berdasarkan margin atas, margin kiri, margin kanan, *baseline*, ukuran huruf, jarak antar baris, dan kemiringan tulisan. Jika *user* ingin memprediksi karakter kembali, maka *user* dapat memasukkan nama dan meng-*upload* gambar tulisan tangan lagi pada halaman ini. Gambar 3.18 merupakan Desain Halaman Prediksi Karakter.

Nama Aplikasi

Nama

gambar tulisan

Button Prediksi

Margin Atas

Kecendrungan karakter berdasarkan margin atas

Margin Kiri

Kecendrungan karakter berdasarkan margin kiri

Margin Kanan

Kecendrungan karakter berdasarkan margin kanan

Baseline

Kecendrungan karakter berdasarkan baseline

Ukuran Huruf

Kecendrungan karakter berdasarkan ukuran huruf

Jarak Antar Baris

Kecendrungan karakter berdasarkan jarak antar baris

Kemiringan Tulisan

Kecendrungan karakter berdasarkan kemiringan tulisan

Copyright

Gambar 3.18 Desain Halaman Prediksi Karakter