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.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam pengimplementasian algoritma *Principal Component Analysis* (PCA) dan *K-Nearest Neighbor* (K-NN) untuk identifikasi tanda tangan disebutkan secara berurutan sebagai berikut.

a. Studi Literatur

Dalam studi literatur, pembelajaran terhadap berbagai teori-teori yang berhubungan dengan pengimplementasian algoritma *Principal Component Analysis* (PCA) dan *K-Nearest Neighbor* (K-NN) untuk identifikasi tanda tangan. Teori-teori tersebut antara lain algoritma *Principal Component Analysis* (PCA) dan *K-Nearest Neighbor* (K-NN), citra tanda tangan, dan pengolahan citra.

b. Analisis Kebutuhan

Tahap ini akan dimulai dari menentukan *requirements*, baik *requirement* sistem ataupun *requirement* perangkat keras yang dibutuhkan untuk dapat mengimplementasikan algotima *Principal Component Analysis* (PCA) dan *K-Nearest Neighbor* (K-NN) untuk *training* tanda tangan.

c. Pengumpulan Sampel

Mengikuti penelitian yang dilakukan oleh Yahya (2013), jumlah sampel yang akan digunakan berasal dari 10 individu dengan masing-masing 10 tanda tangan per individu yang terdiri dari data *training* dan data *testing*.

Implementasi algoritma principal..., Kazenski Alexanda, FTI UMN, 2019

d. Perancangan dan Pembuatan

Merancang aplikasi dengan membuat diagram alur / *flowchart* untuk menunjukkan alur proses dari aplikasi yang akan dibuat, dan dilakukan perancangan terhadap tampilan antarmuka dari aplikasi. Aplikasi akan dibuat dengan menggunakan bahasa pemrograman C# dengan program Microsoft Visual Studio Community 2017.

e. Pengujian Sistem

Melakukan pengujian apakah sistem yang telah dibuat dapat mengidentifikasi citra tanda tangan. Pengujian akan dilakukan sebanyak 5 kali dan mengacu pada pengolahan awal data dengan memfokuskan perhitungan jarak terdekat dengan menggunakan *Euclidean Distance*. Pada bagian pengolahan data awal, data atau sampel yang diperoleh akan dibagi menjadi data *training* dan data *testing*. Citra yang digunakan untuk data *training* dan *testing* berjumlah 100 buah, yang terdiri dari 10 individu dengan 10 tanda tangan tiap individunya. Citra tanda tangan yang telah diperoleh dibagi menjadi 2 bagian yaitu sebagai data *training* dan sebagian lagi menjadi data *testing*, dengan komposisi : 30% data *training* dan 70% data *testing*, 40% data *training* dan 60% data *testing*, 50% data *training* dan 50 %data *testing*, 60% data *training* dan 40% data *testing*, 70% data *training* dan 30% data *testing*. Pengujian sistem ini bertujuan untuk menentukan tingkat akurasi dalam proses pengenalan tanda tangan.

f. Evaluasi dan Kesimpulan

Menganalisis *output* dari sistem, *output* sistem yang akan dievaluasi adalah tingkat pengenalan tanda tangan terhadap tanda tangan yang dimasukkan ke dalam

Implementasi algoritma principal..., Kazenski Alexanda, FTI UMN, 2019

sistem. Hasil evaluasi berupa kesimpulan dari tingkat akurasi pengenalan tanda tangan oleh sistem dalam bentuk persentase.

3.2 Analisis Sistem

Sistem ini dibuat dalam bentuk aplikasi *desktop*, di dalam sistem ini dibuat sebuah aplikasi yang digunakan untuk mengidentifikasi citra uji tanda tangan dengan cara membandingkannya dengan dataset citra *training* untuk mengetahui tingkat akurasi keasliannya. Aplikasi ini menggunakan penyimpanan *file txt* untuk menyimpan data fitur *training*.

Tanda tangan dilakukan secara manual dengan menuliskannya ke dalam sebuah kertas, lalu citra tanda tangan akan diubah menjadi citra *digital* dengan menggunakan *scanner*. *Input* citra dimasukkan ke dalam sistem kemudian akan dilakukan proses *preprocessing* citra. Citra yang telah melalui proses *preprocessing* akan diambil fiturnya dengan menggunakan algortima *Principal Component Analysis* (PCA). Kemudian hasil dari proses ekstraksi fitur data *testing* akan dibandingkan dengan data set citra *training* yang telah didapatkan pada proses *training*, Proses klasifikasi citra ini dilakukan dengan menggunakan algoritma *K-Nearest Neighbor* (K-NN).

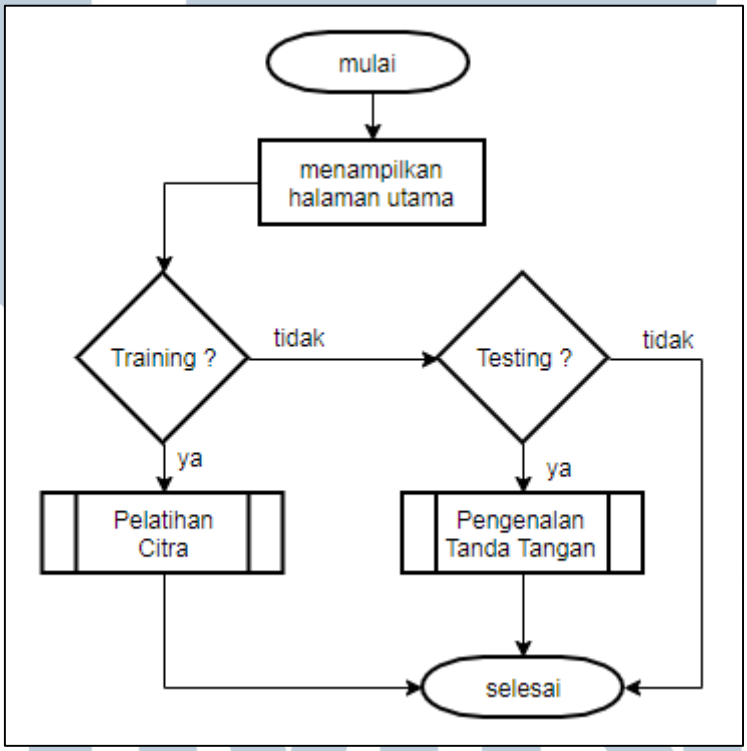
3.3 Flowchart

Flowchart akan digunakan untuk menjelaskan urutan proses yang terdapat dalam sistem yang akan dibangun. *Flowchart* dari sistem ini terdiri dari *flowchart* utama, *flowchart* pelatihan citra, *flowchart preprocessing, flowchart grayscalling* citra, *flowchart* ekstraksi fitur dengan PCA, *flowchart* pengenalan tanda tangan, dan *flowchart* proses identifikasi dengan K-NN.

Implementasi algoritma principal..., Kazenski Alexanda, FTI UMN, 2019

3.3.1 Flowchart Utama

Gambar 3.1 adalah *flowchart* utama dari sistem yang akan dibuat. *Flowchart* ini menggambarkan keseluruhan proses yang dapat dilakukan seorang *user* saat menggunakan aplikasi pengenalan tanda tangan. Ketika aplikasi dijalankan *user* dapat memilih dua menu yang terdapat pada halaman utama aplikasi, *user* dapat memilih menu *training* untuk melakukan pelatihan citra tanda tangan dan menu *testing* untuk melakukan proses pengenalan tanda tangan.



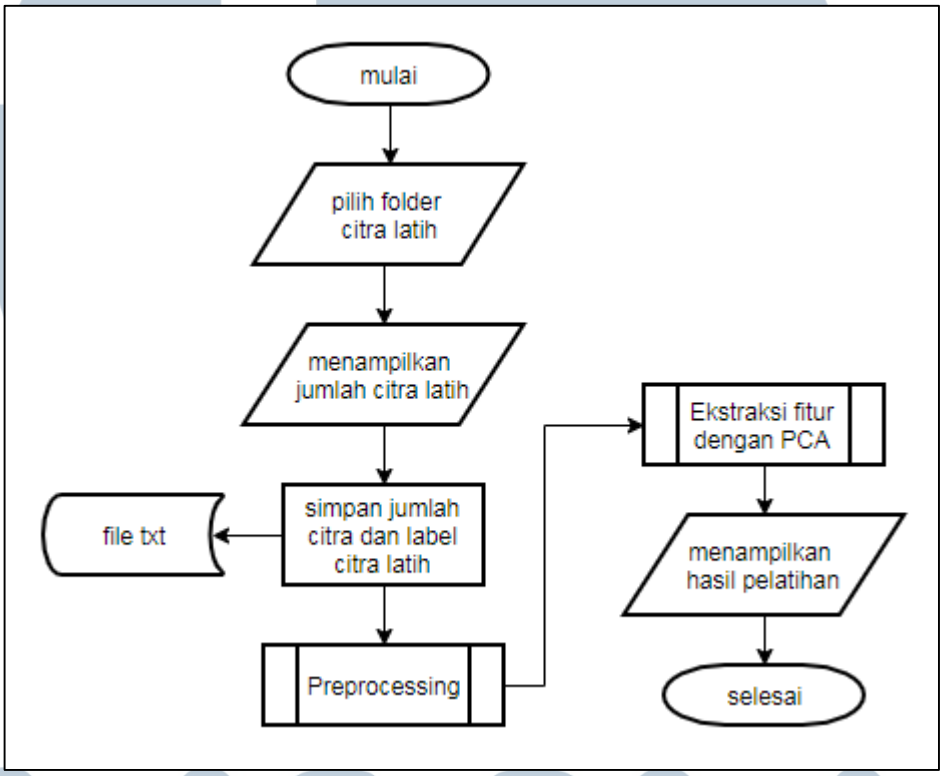
Gambar 3.1 *Flowchart* Utama

3.3.2 Flowchart Pelatihan Citra

Gambar 3.2 adalah *flowchart* pelatihan citra untuk menggambarkan submodul menu *training* di *flowchart* utama. Pelatihan citra tanda tangan dimulai dengan meminta *user* untuk memilih *folder* citra yang berisikan citra latih. Setelah memilih *folder* citra, aplikasi secara otomatis akan menghitung jumlah citra dan memberikan *label* pada tiap citra berdasarkan nama *folder* citra pelatihan dan

Implementasi algoritma principal..., Kazenski Alexanda, FTI UMN, 2019

kemudian akan disimpan ke dalam penyimpanan *file txt*. Lalu citra akan melalui proses *preprocessing* sebelum dilakukan proses ekstraksi fitur dengan PCA. Citra hasil *preprocessing* akan diekstraksi dengan PCA pada submodul eskstrasi fitur dengan PCA.

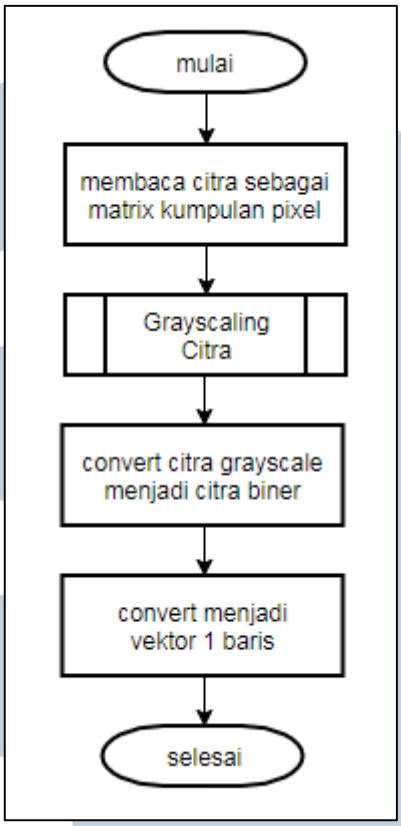


Gambar 3.2 *Flowchart* Pelatihan Citra

3.3.3 Flowchart Preprocessing

Gambar 3.3 adalah *flowchart preprocessing* untuk menggambarkan submodul yang ada pada menu *training*. Proses pertama adalah dengan mengubah *input* menjadi citra abu-abu atau *grayscale*. Lalu proses selanjutnya adalah mengubah citra *grayscale* menjadi citra biner atau citra dengan warna hitam putih dengan melakukan *thresholding* dengan nilai *threshold* 128 yang kemudian akan diubah menjadi vektor 1 baris. Vektor tersebut akan digunakan sebagai *input* untuk submodul ekstraksi fitur dengan PCA pada menu *training* dan *testing*.

Implementasi algoritma principal..., Kazenski Alexanda, FTI UMN, 2019

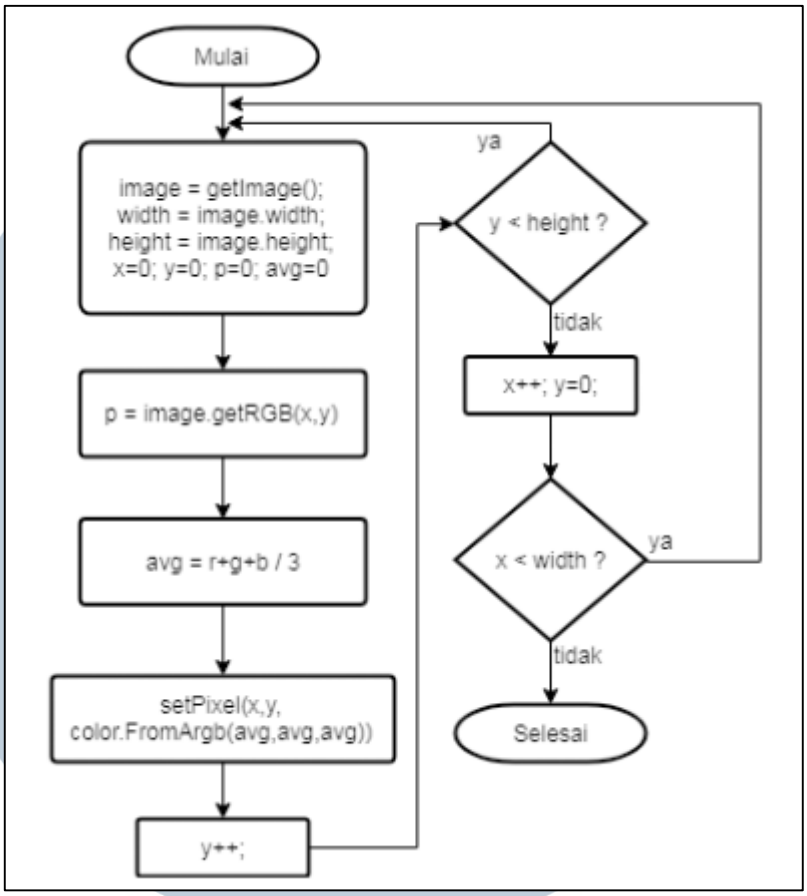


Gambar 3.3 *Flowchart Preprocessing*

3.3.4 Flowchart Grayscaling Citra

Gambar 3.4 merupakan *flowchart grayscaling* citra. Langkah yang pertama kali dilakukan adalah dengan menginisialisasi *variable* yang dibutuhkan. Lalu mendapatkan nilai RGB dari citra dengan menggunakan fungsi getRGB. Setelah nilai RGB didapatkan, maka selanjutnya dihitung nilai rata-rata / *average* dari RGB untuk mendapatkan nilai *grayscale* atau keabuannya. Lalu nilai *grayscale* yang sudah didapatkan akan dimasukkan ke dalam nilai RGB seluruh *pixel* citra, sehingga akan didapatkan citra *grayscale*.

UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 3.4 *Flowchart Grayscaling* Citra

3.3.5 Flowchart Ekstraksi Fitur dengan PCA

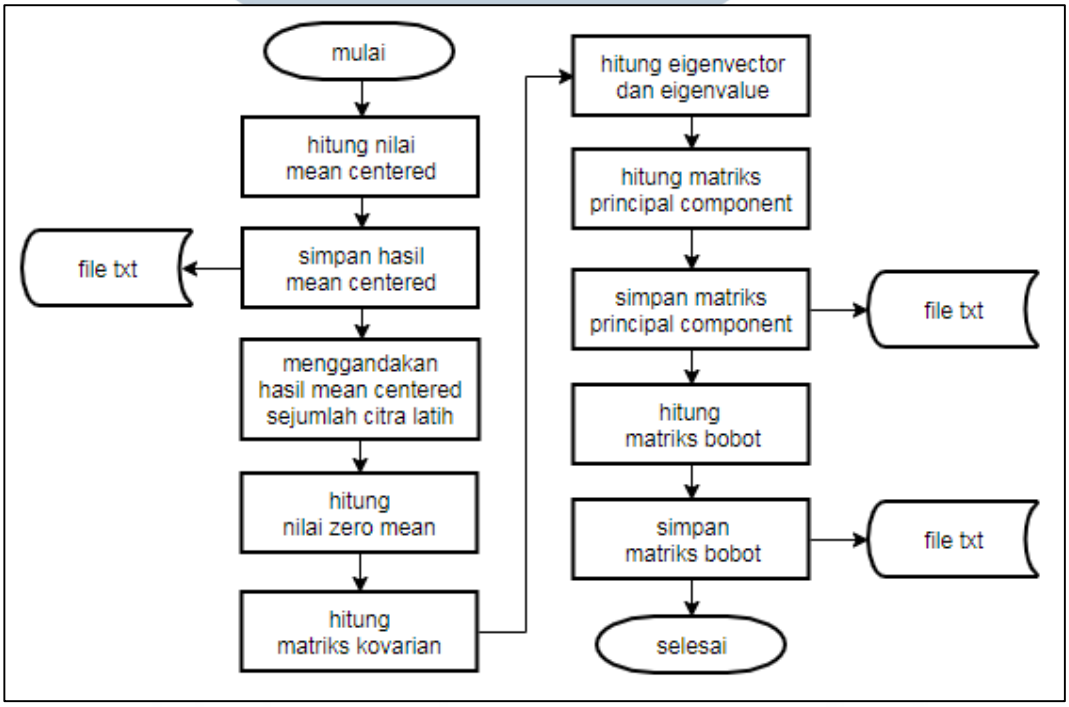
Gambar 3.5 adalah *flowchart* ekstraksi fitur dengan PCA. Langkah pertama adalah dengan mencari nilai rata-rata dari *dataset* citra latih yang dapat dihitung dengan menggunakan Persamaan 2.5, dimana seluruh data latih citra yang telah diubah ke dalam bentuk vektor 1 baris, akan dijumlahkan dan kemudian dibagi sebanyak jumlah data latih citra yang digunakan.

Kemudian akan dicari nilai dari *zero mean*. Nilai *zero mean* dari suatu sampel data akan dihitung melalui proses pengurangan nilai masing-masing data sampel dengan rata-rata data seluruh sampel. Namun, karena adanya perbedaan dimensi dari data sampel (m x n) dan rata-rata seluruh data sampel (l x n), maka perlu disamakan dimensinya dengan menggandakan rata- rata seluruh citra

Implementasi algoritma principal..., Kazenski Alexanda, FTI UMN, 2019

sebanyak m, sehingga rata-rata seluruh sampel juga memiliki dimensi yang sama. Langkah selanjutnya adalah mencari nilai dari matriks kovarian. Matriks kovarian dapat dicari dengan mengalikan hasil perhitungan *zero mean* dengan transposenya sendiri yang dapat dilihat pada persamaan 2.8.

Setelah matriks kovarian didapatkan, langkah selanjutnya adalah dengan mencari nilai dari *eigenvalue* dan *eigenvector* dari matriks kovarian. Penulis akan menggunakan *library* Accord.Net.Decomposition untuk mencari nilai dari *eigenvalue* dan *eigenvector*. Setelah nilai *eigenvalue* dan *eigenvector* didapatkan, langkah selanjutnya adalah membentuk *dataset* baru dari nilai *eigenvector* dengan *eigenvalue* terbesar. Kemudian menghitung matriks bobot dengan cara mengalikan transpose dari *principal component* dengan matriks *zero mean*.

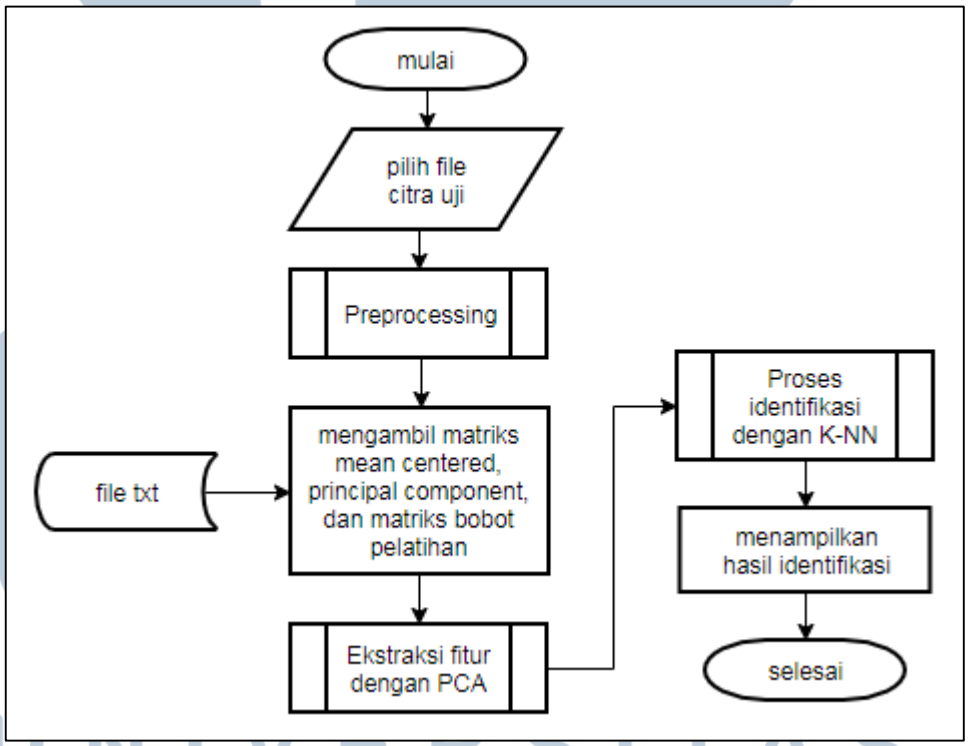


Gambar 3.5 *Flowchart* Ekstraksi Fitur dengan PCA

MULTIMEDIA NUSANTARA

3.3.6 Flowchart Pengenalan Tanda Tangan

Gambar 3.6 merupakan *flowchart* pengenalan tanda tangan. Proses *testing* diawali dengan memilih citra yang ingin diuji keasliannya. Kemudian langkah selanjutnya adalah tahap *preprocessing* seperti pada Gambar 3.3, setelah citra melalui tahap *preprocessing* citra uji diubah menjadi matriks biner. Matriks biner tersebut lalu akan melalui tahap ekstraksi dengan PCA seperti pada Gambar 3.5 agar dimensinya sama dengan dimensi vektor citra latih. Setelah dimensinya sama, citra uji tersebut lalu akan diklasifikasikan menggunakan algoritma K-NN. Lalu hasil identifikasi akan ditampilkan pada layar berupa nama pemilik citra tanda tangan.

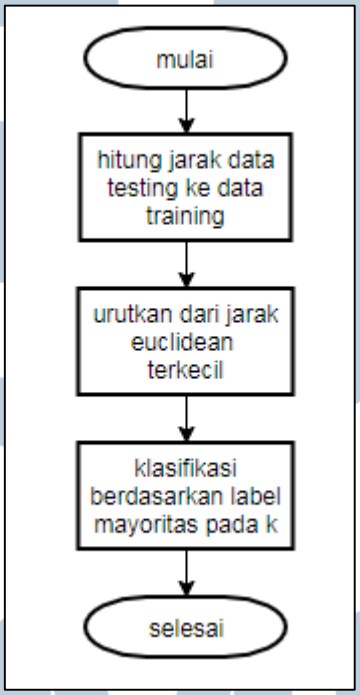


Gambar 3.6 *Flowchart* Pengenalan Tanda Tangan

UNIVERSITAS MULTIMEDIA NUSANTARA

3.3.7 Flowchart Proses Identifikasi dengan K-NN

Gambar 3.7 merupakan *flowchart* proses identifikasi dengan K-NN. Proses K-NN diawali dengan menghitung jarak antara data uji dengan semua data *training* menggunakan persamaan *euclidean distance* pada persamaan 3.1. Lalu urutkan hasil persamaan dari jarak *euclidean* terkecil. Kemudian tentukan hasil klasifikasi berdasarkan kelas yang memiliki anggota terbanyak.



Gambar 3.7 *Flowchart* Proses Identifikasi dengan K-NN

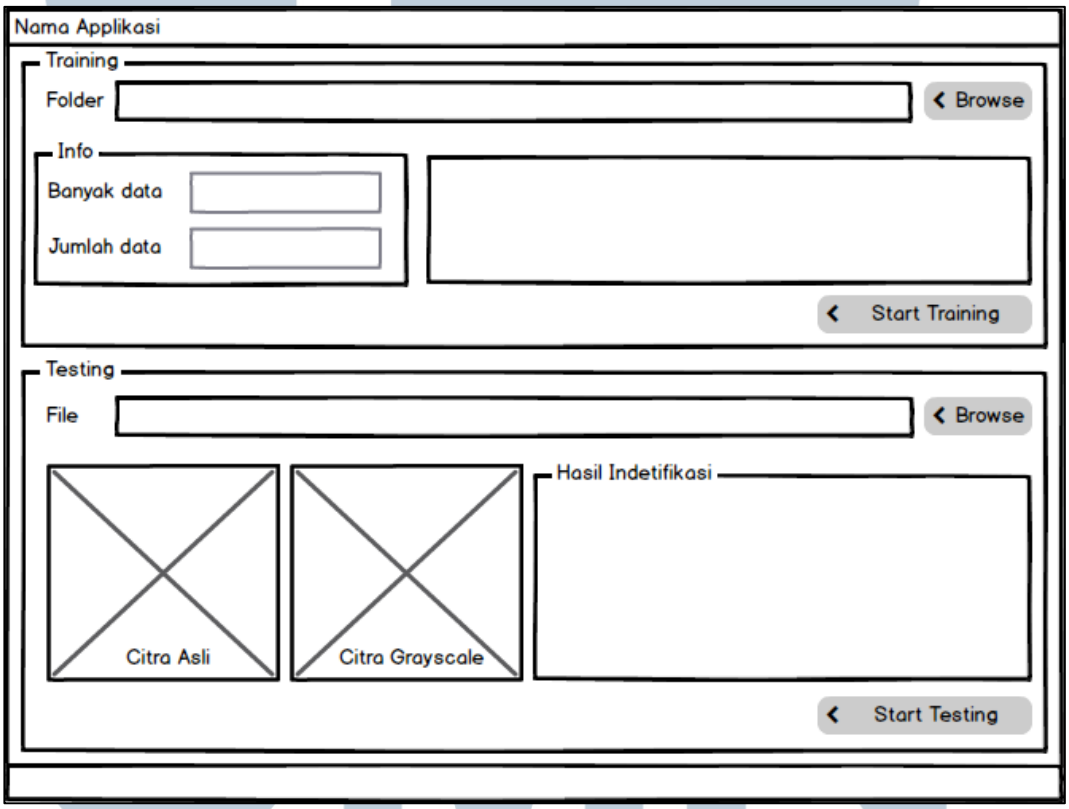
3.4 Desain Antarmuka

Gambar 3.8 merupakan gambaran dari desain mockup sistem yang akan dibuat. Aplikasi identifikasi tanda tangan terbagi atas dua fungsionalitas, yaitu menu *training* yang akan digunakan untuk melatih data latih citra yang akan digunakan sebagai data latih dan menu *testing* yang akan digunakan untuk mengidentifikasi citra uji tanda tangan. Pada menu *training*, *user* akan diminta untuk memilih *folder* dari citra yang ingin digunakan sebagai data *training*. Di

Implementasi algoritma principal..., Kazenski Alexanda, FTI UMN, 2019

dalam menu *training* terdapat menu info yang berfungsi untuk menampilkan banyak data dari *folder* yang dipilih sebelumnya. Lalu terdapat menu *start training* untuk memulai proses *training*.

Lalu terdapat menu *testing*, proses diawali dengan *user* diminta untuk memilih citra yang ingin diuji. Lalu di dalam menu *testing* juga menampilkan citra asli uji dan citra *grayscale*. Terdapat juga tombol *start testing* yang digunakan untuk memulai proses *testing*.



Gambar 3.8 Desain Antarmuka Aplikasi

UNIVERSITAS
MULTIMEDIA
NUSANTARA