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 PENELITIAN DAN PERANCANGAN

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam membangun dan mengimplementasikan aplikasi pelatihan dan pengenalan tulisan tangan dengan menggunakan algoritma *Backpropagation* dan *Generalized Hough Transform* adalah sebagai berikut.

1. Studi Literatur

   Pada tahap ini, akan dilakukan pencarian dan pembelajaran berbagai literatur dan konsep-konsep yang berhubungan dengan rumusan masalah, teori-teori yang berkaitan dengan teknologi dan bahasa pemrograman yang disesuai, algoritma yang digunakan, dan desain aplikasi yang akan dibangun.

2. Pengumpulan Data

   Pada tahap ini, akan dilakukan pengumpulan data-data dari artikel, jurnal, ataupun karya ilmiah lainnya tentang algoritma yang dibahas. Pada tahap ini juga dilakukan perancangan desain aplikasi.

3. Perancangan Jaringan Syaraf Tiruan

   Pada perancangan jaringan syaraf tiruan akan dirancang bagaimana bentuk dari jaringan syaraf yang digunakan untuk pelatihan dan pengenalan tulisan, seperti jumlah layar, penggunaan *hidden layer*, jumlah *hidden layer*, jumlah neuron pada *hidden layer*, jumlah keluaran, dan lain sebagainya untuk keperluan jaringan agar mendapatkan hasil yang akurat.

4. Perancangan Aplikasi

   Aplikasi ini akan dibuat dengan menggunakan algoritma yang telah dipilih (*Generalized Hough Transform* dan *Backpropagation*) dan bahasa pemrograman C# dengan menggunkan .NET *framework* dan Matlab R2008b. Aplikasi yang akan dibuat merupakan sebuah aplikasi yang dapat digunakan untuk mengenali tulisan tangan seseorang dari gambar yang sudah di*scan* sebelumnya, mencocokkannya pada *database* yang ada, dan memberikan hasil pemilik tulisan tangan tersebut.

5. Implementasi

   Pada tahap implementasi ini, penulis akan melakukan penulisan kode program sesuai dengan rancangan yang sudah dilakukan pada tahap sebelumnya. Selain penulisan kode, pada tahap ini penulis juga membuat antarmuka yang diperuntukkan kepada pengguna aplikasi ini.

6. Uji Coba dan Evaluasi

   Di dalam tahap ini, akan dilakukan uji coba terhadap aplikasi yang dibuat, apakah sudah bekerja dengan baik atau belum. Jika terdapat masalah, akan dilakukan perbaikan sehingga dapat berjalan dengan baik.

7. Penulisan Skripsi

   Pada tahap ini, akan dilakukan penyusunan laporan yang memuat dokumentasi mengenai perancangan, pembuatan, serta hasil dari aplikasi yang telah dibuat dalam suatu karya ilmiah.

3.2 Perancangan Jaringan Syaraf Tiruan

Perancangan jaringan syaraf tiruan yang akan dilakukan digunakan pada dua tahap, yaitu pada tahap pelatihan dan tahap pengenalan. Pada tahap pelatihan jaringan syaraf tiruan, akan digunakan algoritma *Backpropagation* untuk mendapatkan bobot yang nantinya disimpan ke dalam *database.* Jaringan akan terus mengubah nilai bobot sampai kriteria tertentu terpenuhi, dalam aplikasi ini adalah nilai kesalahan yang lebih kecil dari toleransi kesalahan.

Tahap selanjutnya adalah tahap pengenalan, aliran pemrosesan dari jaringan hanya dapat bergerak maju, berbeda dengan tahap pelatihan. Bobot yang disimpan dalam *database* akan diambil dan kemudian diberikan kepada jaringan syaraf tiruan sebagai bobot untuk pengenalan sehingga didapatkan hasil keluaran.

Kedua tahap di atas, membutuhkan masukkan yang didapatkan dari hasil ekstraksi ciri menggunakan algoritma *Generalized Hough Transform.* Jika ingin menggunakan fungsi aktivasi sigmoid biner, data harus ditransformasikan dulu karena rentang keluaran fungsi aktivasi sigmoid adalah [0,1] (Siang, 2009). Tapi, akan lebih baik jika ditransformasikan ke interval yang lebih kecil, misal interval [0.1, 0.9] (Siang, 2009). Ini mengingat fungsi sigmoid merupakan fungsi asimtotik yang nilainya tidak pernah mencapai 0 ataupun 1 (Siang, 2009). Hasil dari ekstraksi ciri akan diubah menjadi bilangan antara [0.1,0.9] dengan menggunakan normalisasi karena hasil yang didapatkan merupakan bilangan ratusan. Normalisasi dilakukan dengan menggunakan rumus normalisasi berikut (Siang, 2009).

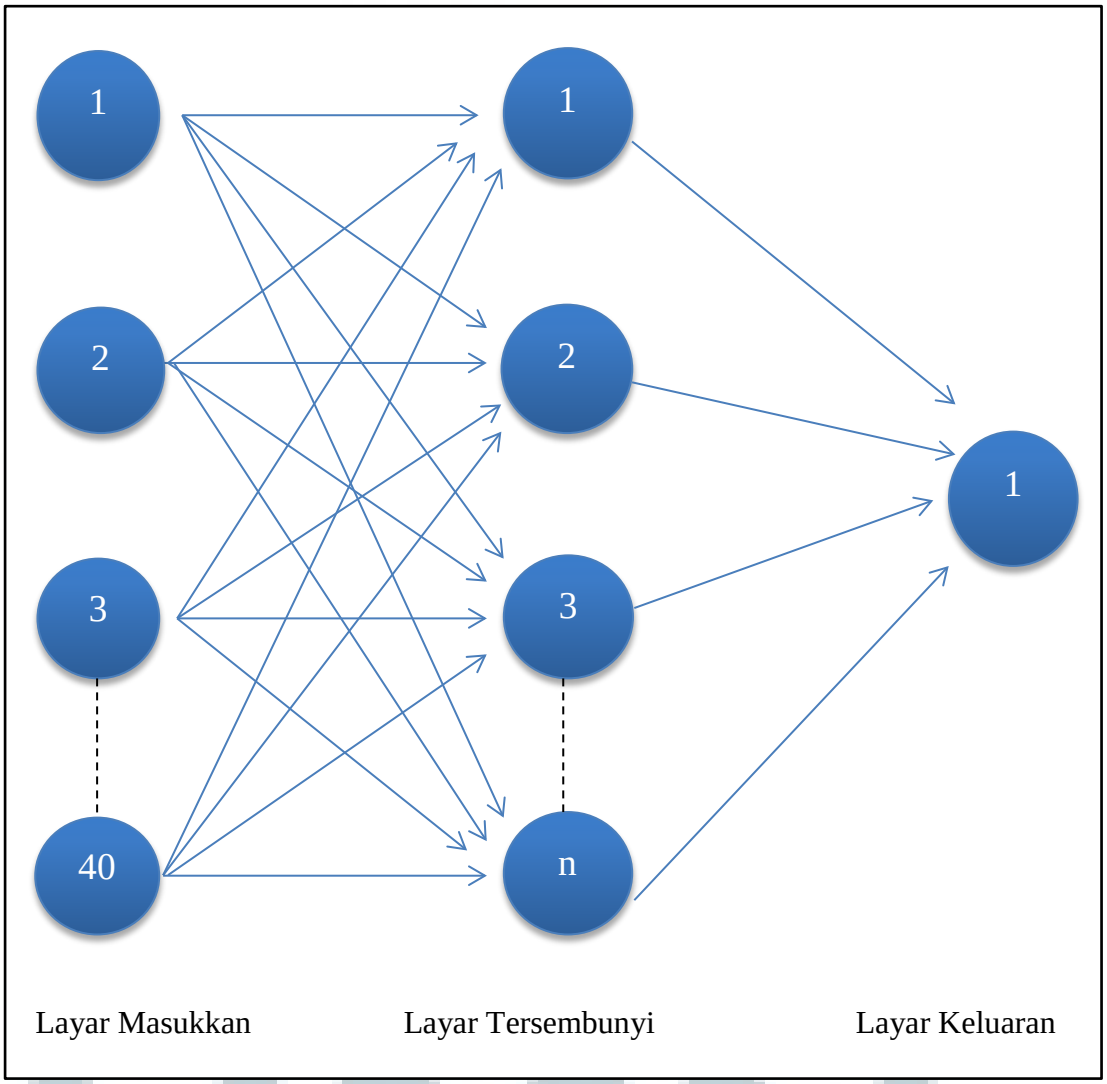
$$x' = \frac{0.8\ (x - \text{nilai terkecil})}{\text{nilai terbesar} - \text{nilai terkecil}} + 0.1$$

……. Rumus 3.1

Berikut ini adalah tahap – tahap perancangan jaringan syaraf tiruan yang akan digunakan dalam aplikasi ini.

1. Jaringan sayaraf tiruan terdiri dari tiga buah layar seperti yang dijelaskan pada Gambar 3.1.
2. Jumlah masukkan pada layar masukkan (*input layer)* pada aplikasi ini ditetapkan sesuai dengan hasil dari hasil ekstraksi ciri menggunakan algoritma *Generalized Hough Transform* adalah empat puluh buah masukkan.
3. Layar tersembunyi (*hidden layer)* yang dicoba berjumlah satu buah dan jumlah neuron pada layar tersembunyi akan ditentukan setelah proses pencarian arsitektur jaringan optimum.
4. Antara layar tersembunyi dan keluaran terdapat koneksi satu arah yang memiliki bobot yang nilainya awalnya dipilih secara acak antara -0.5 – 0.5.
5. Jumlah keluaran yang ada pada jaringan merupakan satu buah, dengan nilai target 1.
6. Data pelatihan akan berjumlah sepuluh buah dan jumlah data pengenalan adalah dua puluh buah.

Gambar 3.1 di bawah ini merupakan gambar dari perancangan jaringan syaraf tiruan pada aplikasi yang akan dibuat.



Gambar 3.1 Perancangan Jaringan Syaraf Tiruan

3.3 Perancangan Aplikasi

Pada perancangan fungsional aplikasi pelatihan dan pengenalan tulisan tangan ini, dilengkapi dengan *data flow diagram* (DFD), *flowchart* dan struktur tabel.

3.3.1 Flowchart

Flowchart merupakan sebuah diagram yang menggambarkan alur proses atau kerja dari aplikasi. *Flowchart* digunakan agar pembaca lebih memahami alur proses aplikasi dibandingkan dengan tulisan. Pada tahap perancangan aplikasi ini, terdapat *flowchart* yang digunakan untuk memvisualisasikan aliran proses atau kerja dari aplikasi, terdapat *flowchart* pelatihan jaringan dan pengenalan.

A. Flowchart Pelatihan Jaringan

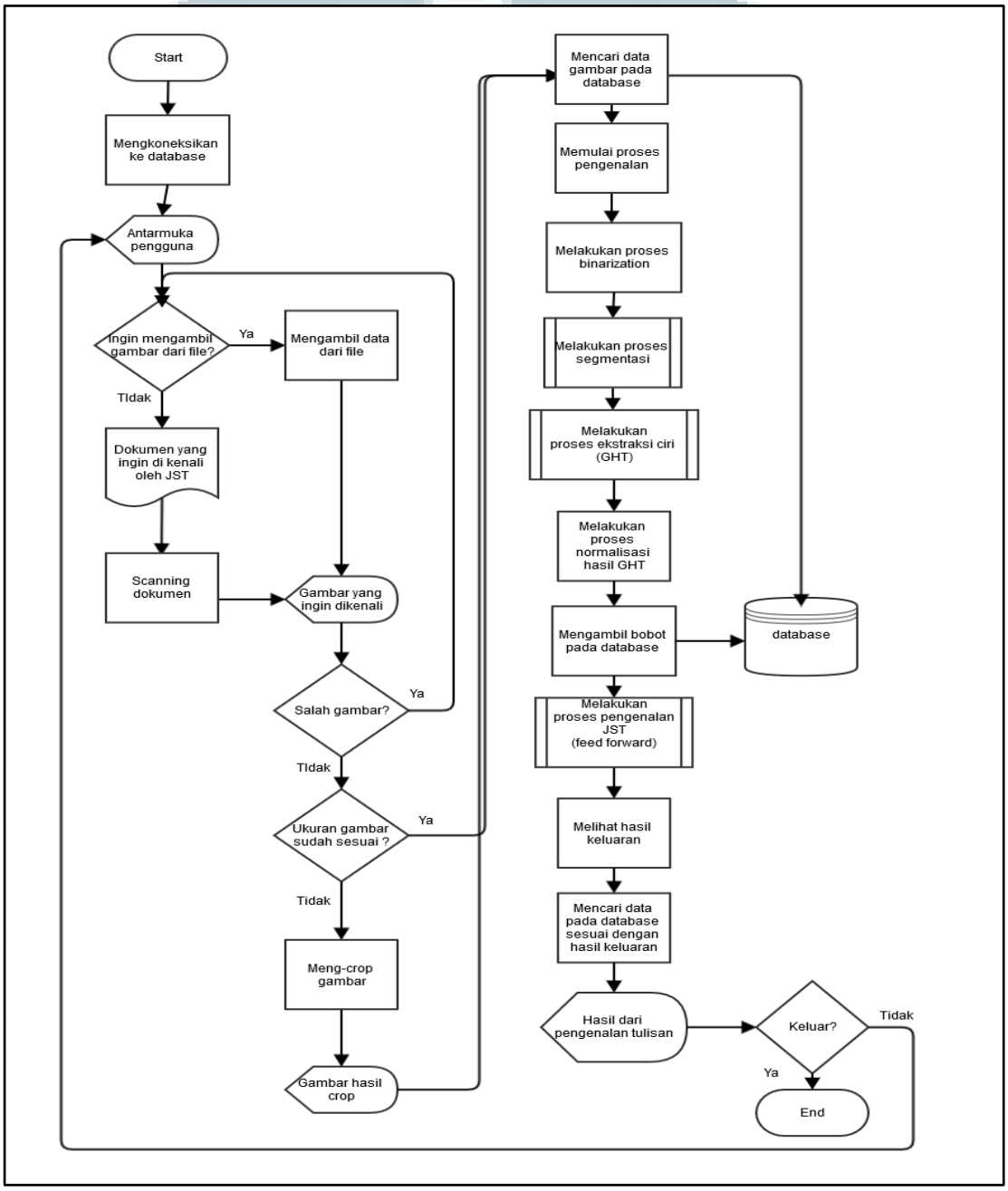
Proses pelatihan jaringan merupakan hal yang sangat penting dalam aplikasi ini yang digunakan untuk mencari hasil bobot yang nantinya akan dipakai dalam proses pengenalan tulisan tangan. Pelatihan jaringan menggunakan algoritma Backpropagation, yang akan memproses ulang masukkan dengan bobot yang berbeda-beda setiap iterasinya untuk menemukan bobot yang paling sesuai sampai jumlah iterasi mendekati atau sama dengan maksimal iterasi. *Flowchart* ini menggambarkan proses yang dilakukan seorang pengguna dari awal sampai akhir saat pengguna tersebut ingin menggunakan aplikasi ini untuk melatih jaringan.

Gambar 3.2 *Flowchart* Pelatihan Jaringan

B. Flowchart Pengenalan Tulisan Tangan

Flowchart kedua merupakan *flowchart* yang menggambarkan proses pengenalan jaringan. Berbeda dengan proses pelatihan jaringan, proses

pengenalan jaringan tidak melakukan iterasi. Proses pengenalan hanya mengambil data bobot dari *database* dan kemudian memakai bobot tersebut untuk *feed forward* saja. Hasil dari pengenalan akan ditentukan apakah tulisan tersebut asli atau tidak.



Gambar 3.3 *Flowchart* Pengenalan Tulisan Tangan

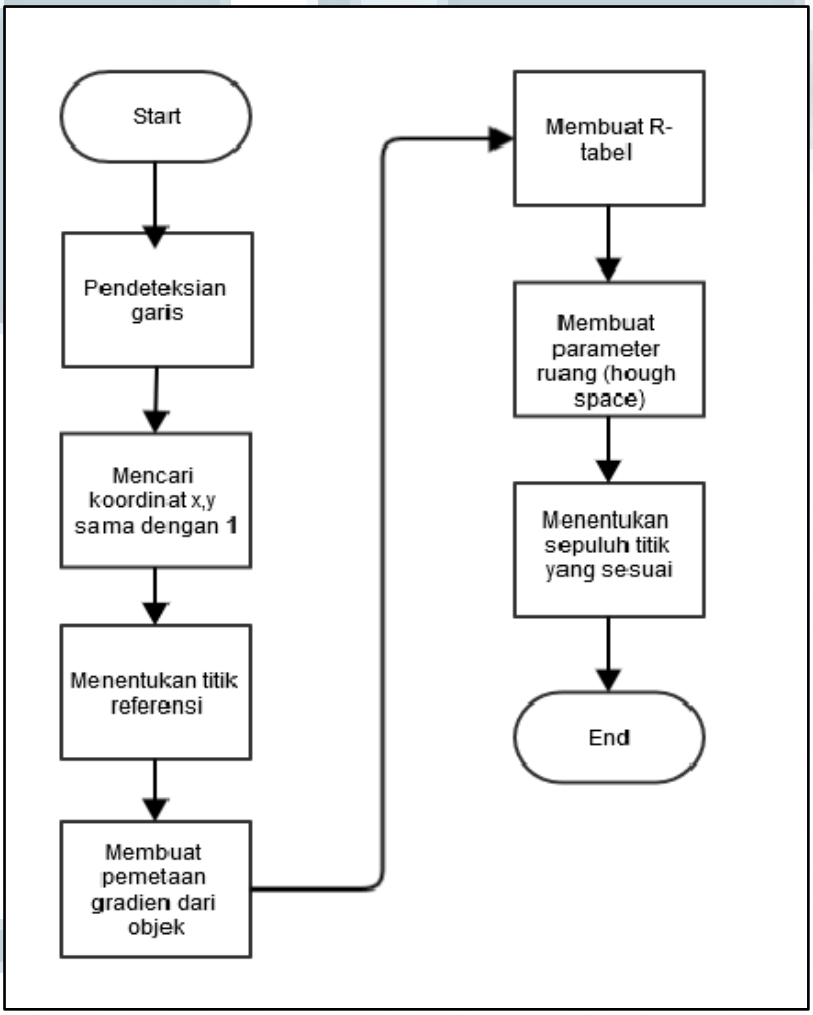
C. Flowchart Sub Proses Melakukan Proses Segmentasi

Flowchart sub proses melakukan proses segmentasi merupakan penjabaran secara lebih rinci tentang alur kerja dari proses segmentasi yang dilakukan sebelum citra dimasukkan pada tahap ekstraksi fitur. Proses segmentasi digunakan untuk memecah karakter-karakter atau huruf-huruf dari kata pada citra.

Gambar 3.4 *Flowchart* Sub Proses Melakukan Proses Segmentasi

D. Flowchart Sub Proses Melakukan Proses Ekstraksi Ciri

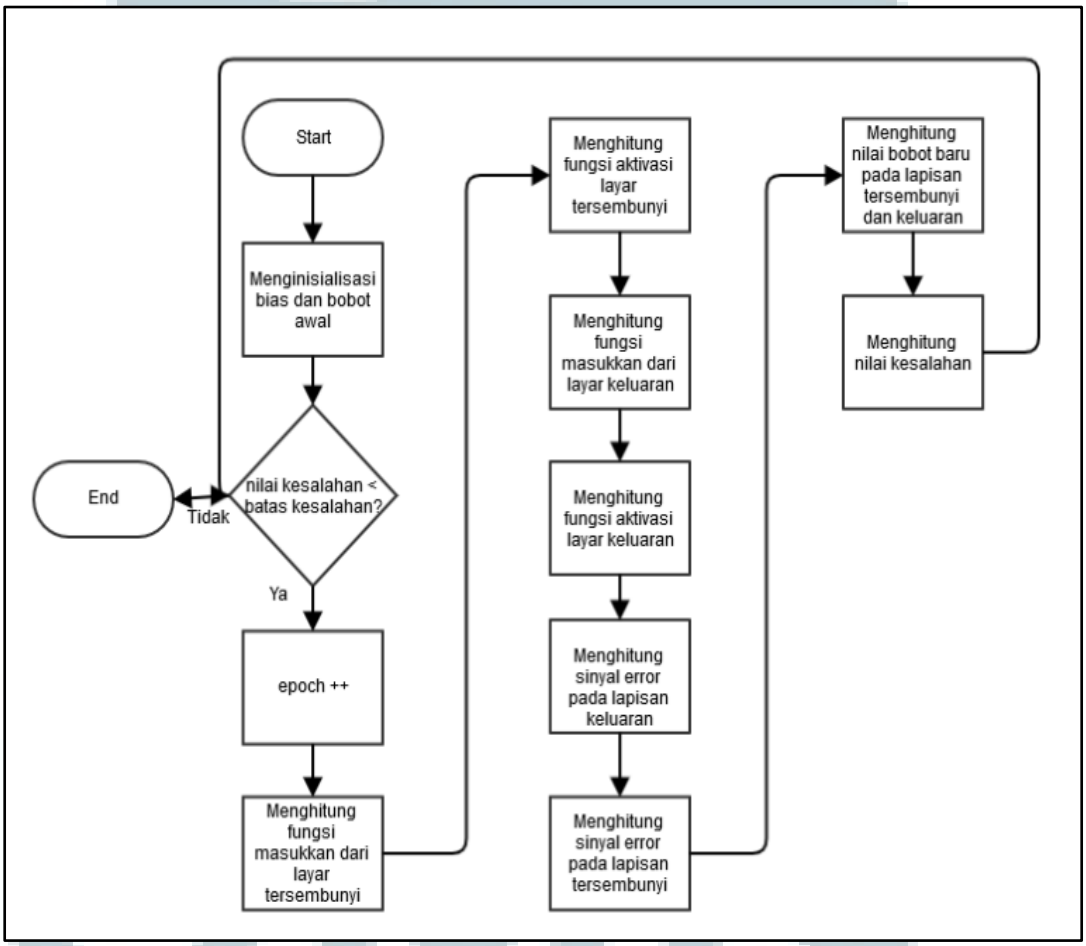
Flowchart sub proses melakukan proses ekstraksi ciri merupakan sebuah alur kerja dari proses ekstraksi ciri dimana pada aplikasi ini menggunakan algoritma *Generalized Hough Transform*. Algoritma ini kan melakukan pendeteksian terlebih dahuku menggunakan algoritma *canny edge detector* sebelum pemrosesan algoritma *Generalized Hough Transform* itu sendiri dilakukan.



Gambar 3.5 *Flowchart* Sub Proses Melakukan Proses Ekstraksi Ciri (GHT)

E. Flowchart Sub Proses Melakukan Proses Pelatihan JST

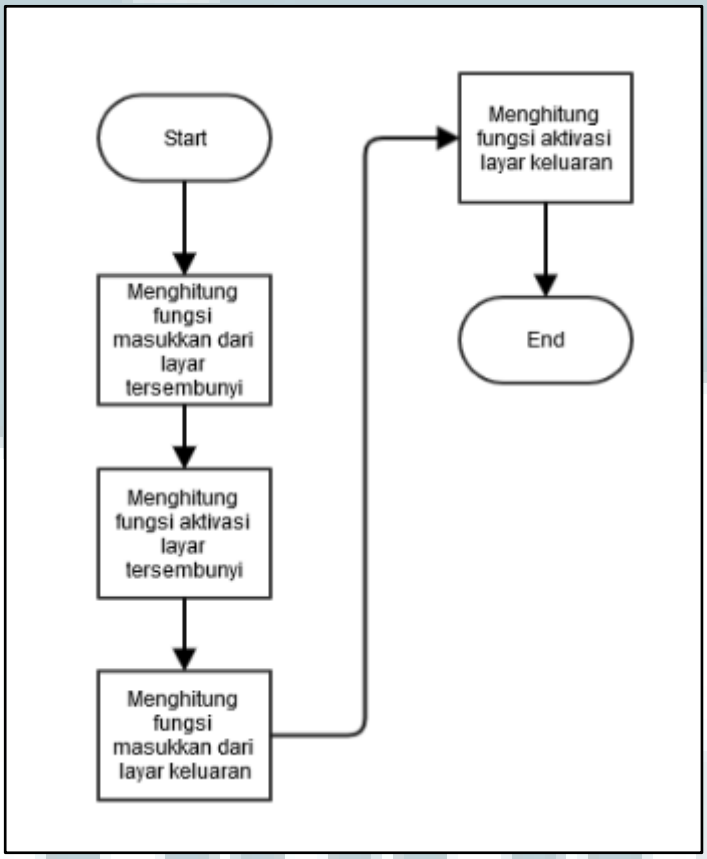
Flowchart sub proses melakukan proses pelatihan JST merupakan penggambaran alur kerja dari proses pelatihan JST yang menggunakan algoritma *Backpropagation.* Algoritma ini akan terus mengubah bobot sampai nilai kesalahan yang dihitung lebih kecil dari batas kesalahan yang sudah ditetapkan sebelumnya.



Gambar 3.6 *Flowchart* Sub Proses Melakukan Proses Pelatihan JST

F. Flowchart Sub Proses Melakukan Pengenalan JST

Flowchart ini menggambarkan tentang alur kerja dari pengenalan tulisan tangan yang dilakukan menggunakan jaringan syaraf tiruan. Proses yang dilakukan hanya proses *feed forward* saja, tidak ada pengubahan bobot pada proses ini. Penghitungan yang dilakukan hanya penghitungan fungsi masukkan dan aktivasi dari masing-masing neuron pada layar tersembunyi dan keluaran.



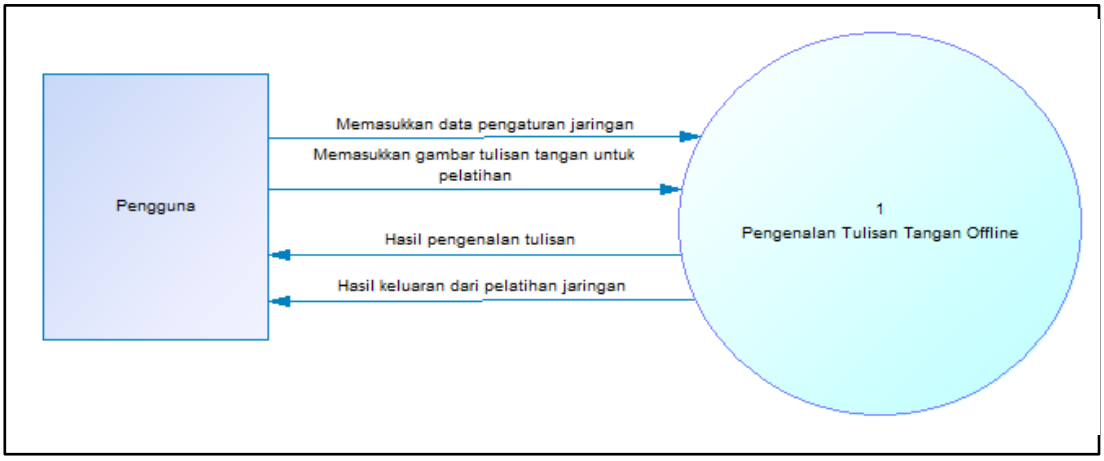
Gambar 3.7 *Flowchart* Sub Proses Melakukan Proses Pengenalan JST

3.3.2 Data Flow Diagram

Data flow diagram (DFD) merupakan representasi grafik dari aliran data pada sebuah sistem informasi. DFD digunakan untuk memvisualisasikan bagaimana sebuah sistem beroperasi. Di bawah ini merupakan gambar dari *data flow diagram* dari aplikasi yang telah dibuat.

A. Context Diagram

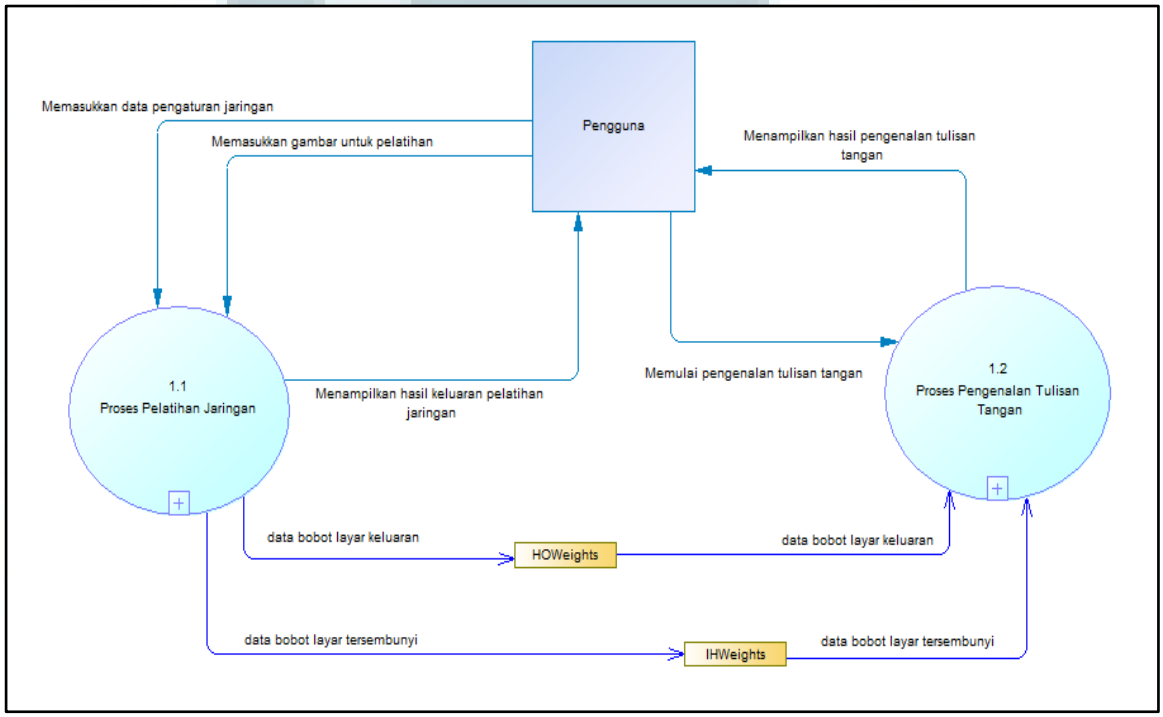
Pada *context digram* di bawah ini, terdapat interaksi antara pengguna/*user* dengan sistem pengenalan tulisan tangan *offline.* Interaksi yang dilakukan oleh pengguna adalah memasukkan data pengaturan jaringan syaraf yang diinginkan, misalnya besarnya jumlah neuron pada layar tersembunyi. Kemudian, pengguna dapat memasukkan gambar-gambar yang ingin dilatih oleh jaringan. Setelah proses pelatihan jaringan selesai dilakukan, maka sistem akan menampilkan hasil keluaran dari pelatihan jaringan masing-masing gambar kepada pengguna. Proses pengenalan tulisan tangan akan menampilkan hasil pengenalan tulisan tangan kepada pengguna yang menentukan apakah tulisan tersebut cocok atau tidak.



Gambar 3.8 *Context Diagram*

B. Diagram Level 1

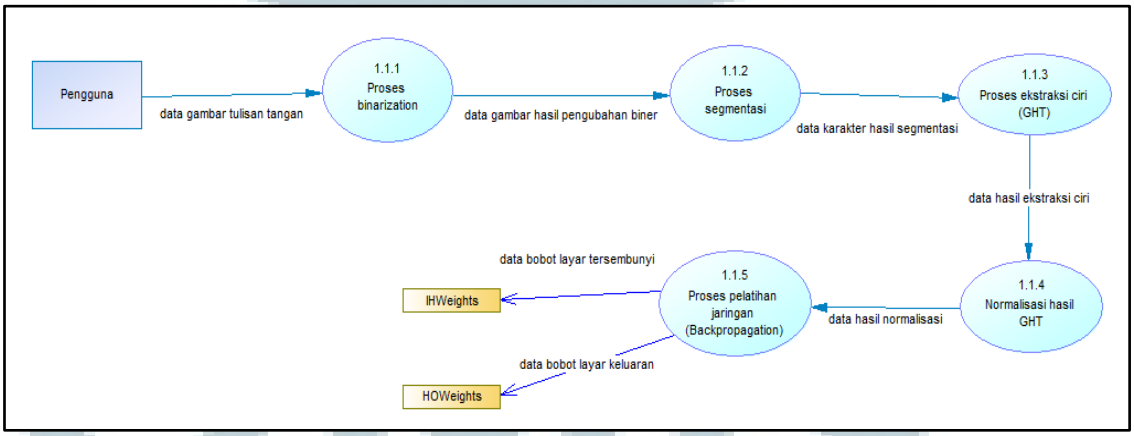
Diagram level 1 merupakan diagram turunan dari *context diagram* pada Gambar 3.8. Pada diagram level 1 ini terdapat dua buah proses, yaitu proses pelatihan jaringan dan proses pengenalan tulisan tangan. Proses pelatihan jaringan mendapatkan masukkan berupa data pengaturan jaringan dan gambar yang digunakan untuk pelatihan jaringan, kemudian proses ini akan menampilkan hasil keluaran dari pelatihan jaringan dan menghasilkan bobot layar tersembunyi dan layar keluaran yang akan disimpan ke dalam *database* pada tabel yang terpisah, yaitu tabel HOWeights untuk bobot layar keluaran dan IHWeights untuk bobot layar tersembunyi. Pada proses pengenalan tulisan tangan, pengguna menginisiasi proses pengenalan, kemudian data bobot dari *database* akan diambil untuk penghitungan pada proses pengenalan. Setelah selesai proses pengenalan, maka akan ditampilkan hasil pengenalan tulisan tangan.



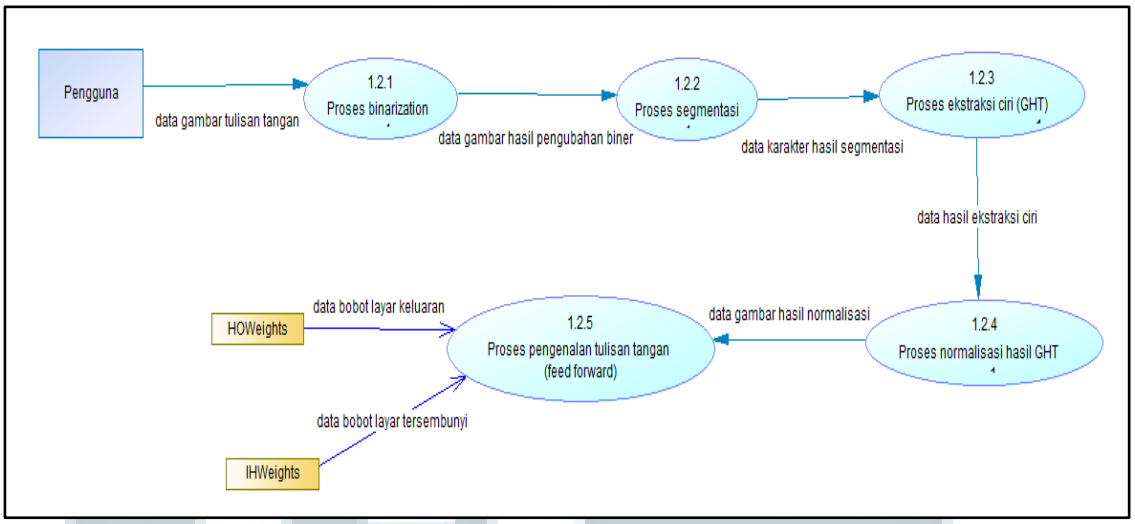
Gambar 3.9 Diagram Level 1

C. Diagram Level 2

Terdapat dua buah diagram pada level 2 yang berasal dari proses 1.1 dan proses 1.2 pada digram level 1. Kedua diagram ini menjelaskan secara lebih rinci proses dari pelatihan jaringan dan pengenalan tulisan tangan. Kedua proses tersebut melewati beberapa proses yang sama, yaitu proses *binarization*, segmentasi, ekstraksi ciri dengan algoritma *Generalized Hough Transform*, dan normalisasi hasil GHT. Perbedaan diantara kedua proses hanya terletak pada tahap akhir. Proses pelatihan jaringan (proses 1.1) akan menjalankan proses pelatihan dengan algoritma *Backpropagation* dan menyimpan bobot hasi pelatihan ke *database* (tabel HOWeights dan IHWeights). Sedangkan proses pengenalan tulisan tangan (proses 1.2) akan menjalankan proses pengenalan tulisan tangan oleh jaringan dengan fase *feed forward* dengan mengambil bobot yang sudah tersimpan pada *database* dari hasil pelatihan jaringan.



Gambar 3.10 Diagram Level 2 Proses 1.1



Gambar 3.11 Diagram Level 2 Proses 1.2

3.3.3 Struktur Tabel

Tidak adanya tabel yang terhubung antara satu sama lain, membuat aplikasi ini tidak memiliki *entity relationship diagram.* Oleh karena itu, hanya disampaikan struktur table pada aplikasi ini. Terdapat beberapa tabel yang digunakan untuk menyimpan data yang dihasilkan dari proses pelatihan jaringan dan terdapat pula tabel yang digunakan untuk menyimpan data user yang gambar tulisan tangannya sudah dilatih. Di bawah ini merupakan struktur tabel yang dibuat untuk mendukung proses aplikasi ini.

Tabel 3.1 Keterangan Tabel IHWeights

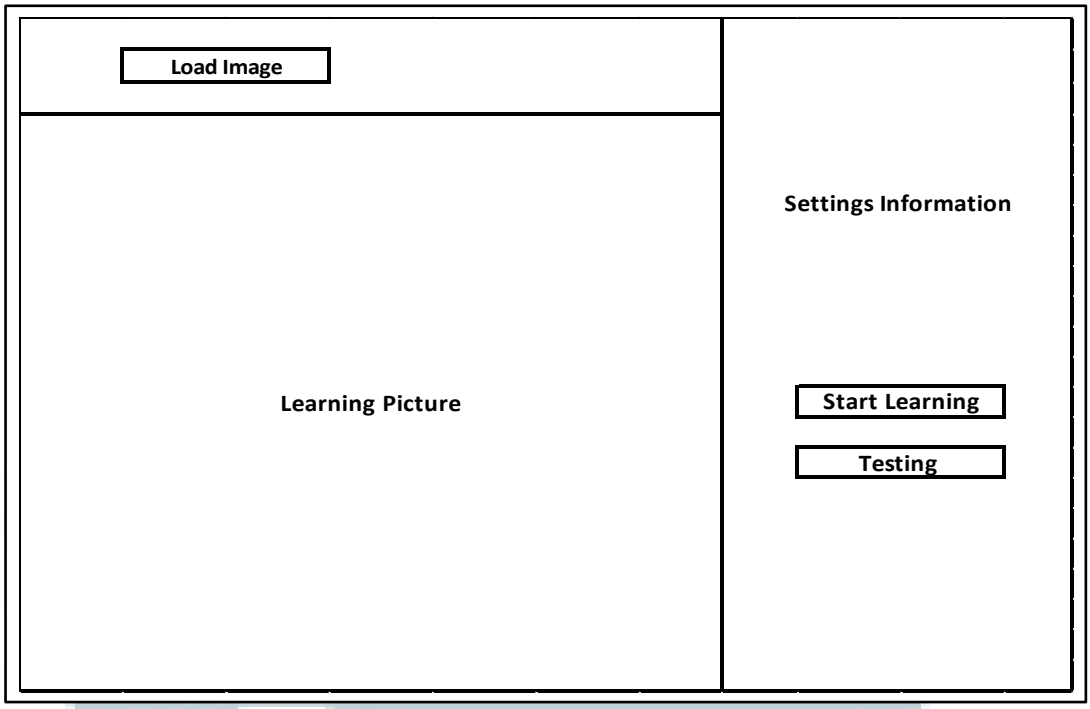
Kolom	Tipe Data	Keterangan
Row	integer	Baris pada matriks bobot
Col	integer	Kolom pada matriks bobot
Value	double(10,8)	Nilai bobot dari layar masukkan ke tersembunyi

Tabel 3.2 Keterangan Tabel HOWeights

Kolom	Tipe Data	Keterangan
Row	integer	Baris pada matriks bobot
Col	integer	Kolom pada matriks bobot
Value	double(10,8)	Nilai bobot dari layar tersembunyi ke keluaran

3.3.4 Perancangan Antarmuka Pengguna

Pada perancangan antarmuka pengguna ini, terdapat satu antarmuka yang mewakili seluruh penelitian ini, yaitu



Gambar 3.12 Antarmuka Pengguna Halaman Utama Aplikasi

Pada antarmuka ini, terdapat tombol *load image* untuk mengambil gambar dari direktori pada komputer. Kemudian akan ditampilkan pada area *learning picture.* Area *settings information* berisi tentang informasi data JST yang dapat diubah oleh pengguna untuk mendapatkan hasil yang optimal. Setelah semua sudah tersedia, pengguna dapat menekan tombol *start learning* untuk memulai proses pelatihan. Tombol *testing* digunakan untuk melakukan uji coba untuk gambar-gambar tulisan tangan yang sudah ditetapkan sebelumnya. Kemudian setelah uji coba selesai dilakukan, maka aplikasi akan menampilkan hasil dari pengenalan tulisan tangan apakah cocok atau tidak cocok.