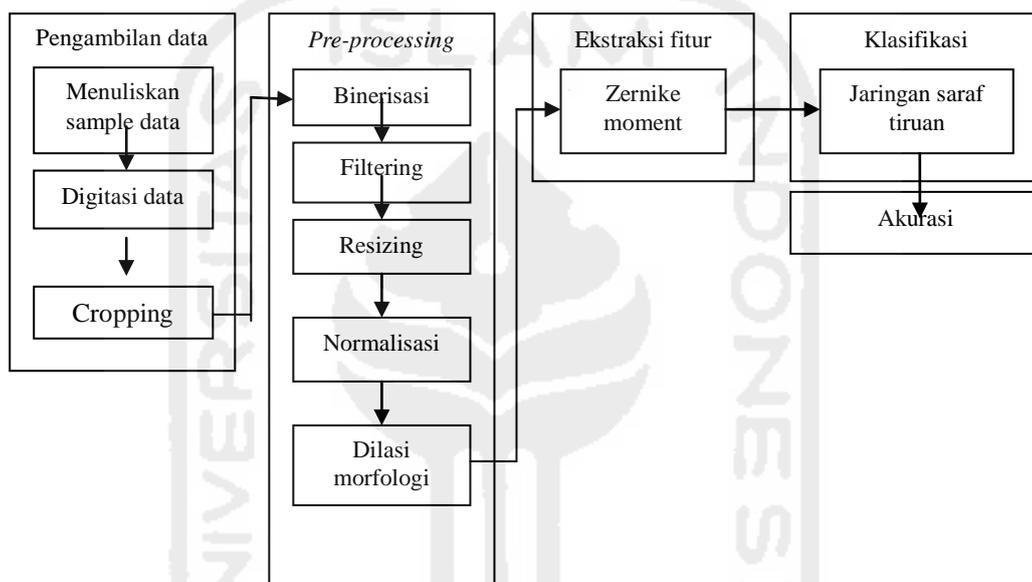


BAB III

PERANCANGAN SISTEM

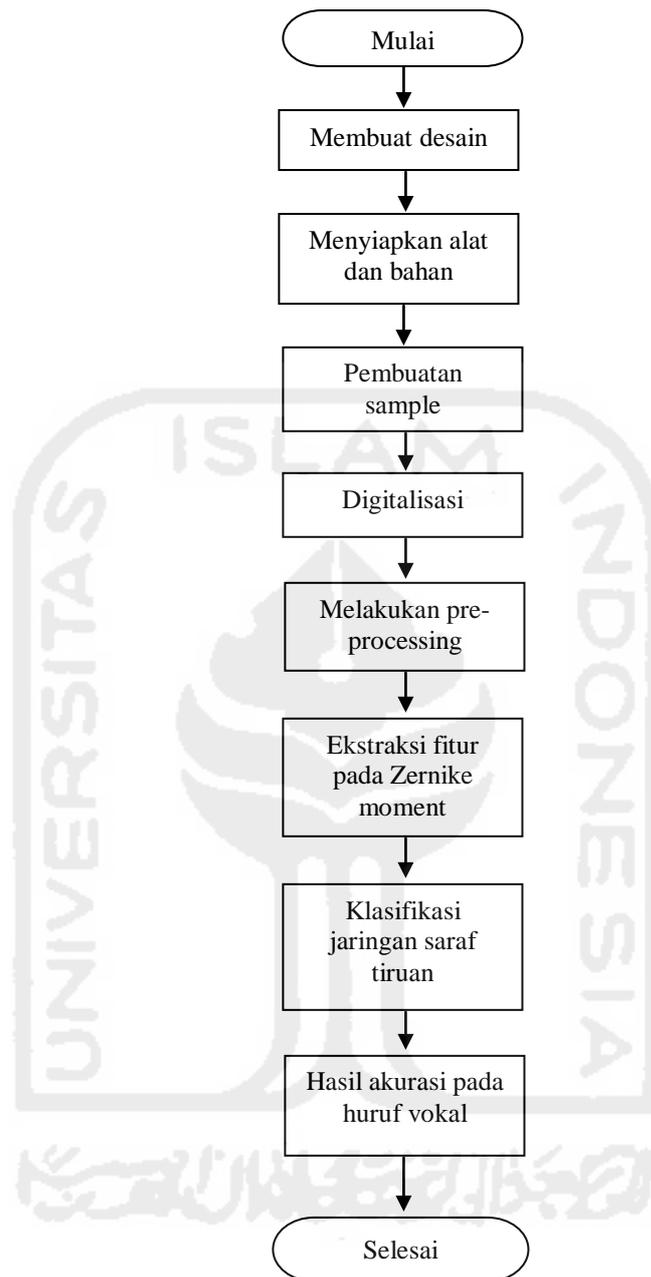
3.1 Alur Penelitian

Perancangan sistem simulasi identifikasi dan pengenalan pola tulisan tangan huruf vokal seperti terlihat pada gambar 3.1.



Gambar 3.1 Diagram alir perancangan sistem

Pembuatan sistem simulasi ini dibagi menjadi lima tahap, yaitu tahap pengambilan data, tahap *pre-processing*, tahap ekstraksi fitur, tahap klasifikasi dan akurasi. Perancangan simulasi ini menggunakan fitur *m-file* yang terdapat pada *software* Matlab 2010a. Diagram alir tahapan pembuatan sistem ditunjukkan pada gambar 3.2.



Gambar 3.2 Diagram alir tahapan pembuatan sistem

3.2 Peralatan yang Digunakan

Alat yang digunakan dalam penelitian ini meliputi perangkat keras dan perangkat lunak. Perangkat keras yaitu *notebook* SAMSUNG NP275E4V-K01ID. Perangkat lunak berupa program-program dalam *notebook* yaitu Windows 7,

Matlab, dan program-program pendukung lainnya.

3.3 Data

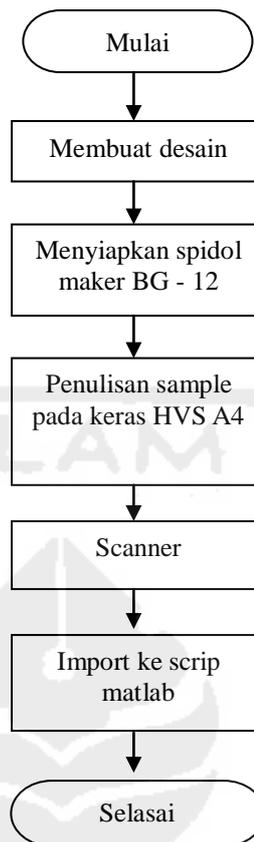
Bahan penelitian yaitu tulisan tangan vokal A , E , I , O , U . Masing-masing huruf diambil datanya sebanyak 10 tulisan tangan individu yang berbeda, sehingga diperoleh data sebanyak 50 sample dari keseluruhan huruf vokal. Rincian data diperlihatkan pada Tabel 3.1.

Tabel 3.1. Rincian data

Huruf yang akan dikenali polanya	Jumlah sample
A	10
I	10
U	10
E	10
O	10
Jumlah total	50

3.4 Tahap Digitalisasi Data

Tahap digitalisasi data adalah tahap awal dalam proses pengenalan pola huruf alphabet. Tahap awal ini memproses tulisan tangan menjadi sebuah data yang siap diolah lebih lanjut, yaitu merubah data kedalam bentuk citra digital. Diagram alir proses dari tahap pembuatan sampel bisa dilihat pada gambar 3.3.

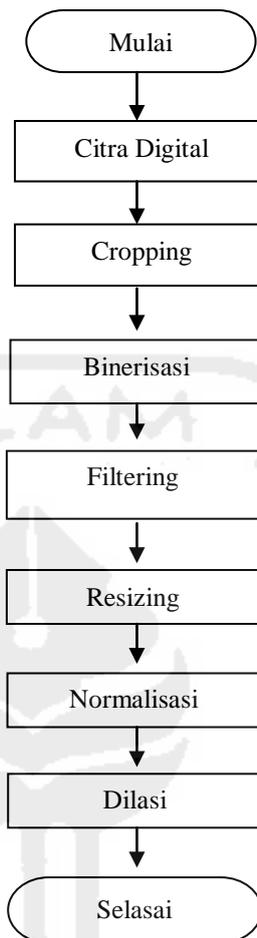


Gambar 3.3 Diagram alir pengambila

3.5 Tahap Pre-Proses

Tahap ini adalah untuk mempersiapkan sample data sebelum di proses lebih lanjut. Ditahap ini, data yang sudah berupa citra digital akan melalui serangkain tahapan pre-proses yang bertujuan untuk memperbaiki kualitas citra.

Gambar 3.4 memperlihatkan tahapan *pre-processing*.



Gambar 3.4 Diagram alir tahap *pre-processing*

3.6 Cropping

Data yang sudah dalam bentuk digital dilakukan proses pemotongan atau *cropping* pada koordinat tertentu. Tujuannya adalah untuk memisahkan masing-masing citra tulisan tangan huruf vokal, agar dapat dilakukan pemrosesan secara terpisah.

3.7 Binerisasi

Citra hasil pemindaian merupakan citra warna, pada tahapan pre-proses ini citra warna akan dikonversi menjadi citra biner atau citra hitam putih. Proses konversi menggunakan perintah `rgb2gray` di MATLAB. Berikut adalah contoh

listing program proses binerisasi gambar huruf vokal ‘A’ pada sample yang pertama:

```
A = imread ('A_1.jpg');           %membaca file gambar
BW = im2bw(A, 0.5);             %proses binerisasi citra
figure, imshow (BW)            %menampilkan gambar
binerisasi
title('Hasil binerisasi');      %hasil gambar binerisasi
```

3.8 Filtering

Citra hasil pemindaian biasanya mengandung derau yang berupa bintik-bintik hitam, derau tersebut akan memberikan pengaruh negatif kepada proses pengolahan citra. Untuk menghilangkan derau pada citra, maka pada tahapan pre-proses dilakukan filtering. Metode filtering yang digunakan pada penelitian ini adalah median filtering. Berikut adalah contoh *listing program* proses filter median gambar huruf vokal ‘A’ pada sample yang pertama:

```
I = medfilt2(BW);               %proses memfilteran
figure, imshow (I)
title('Hasil Filtering');
```

3.9 Resizing

Citra hasil *cropping* memiliki ukuran yang tidak sama, dikarena pemotongan dilakukan pada koordinat yang berbeda pada masing-masing huruf. Untuk mempermudah proses pengolahan dan pengenalan pola, ukuran masing-masing citra huruf distandardkan. Pada penelitian ini ukuran citra yang digunakan adalah 230x230 pixel. Berikut adalah contoh *listing program* proses resizing gambar huruf vokal ‘A’ pada sample yang pertama:

```
S = imresize(I, [230 230]);           %menentukan ukuran pixel
figure, imshow (S)
title('Hasil Resize');
```

3.10 Normalisasi

Proses normalisasi bertujuan untuk memperbaiki posisi citra yang miring karena posisi pemindaian yang tidak presisi. Untuk memperbaiki kemiringan tersebut pada penelitian ini menggunakan *imrotate* pada sudut Θ . Berikut adalah contoh *listing program* proses normalisasi gambar huruf vokal ‘A’ pada sample yang pertama:

```
J = imrotate(S,0,'bilinear','crop');   %memutar gambar
figure, imshow (J)
title('Hasil Normalisasi');
```

3.11 Morfologi Dilasi

Proses morfologi yang diterapkan pada penelitian ini adalah dilatasi. Proses dilatasi merupakan proses penambahan pixel pada tepi objek pada citra. Proses ini bertujuan untuk lebih mempertegas objek pada citra. Berikut adalah contoh *listing program* proses dilasi gambar huruf vokal ‘A’ pada sample yang pertama:

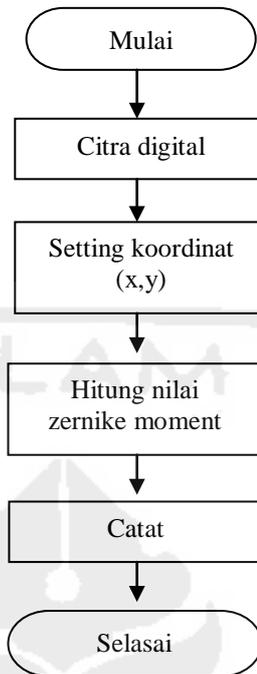
```
D=strel('square',3);                 %membentuk matriks persegi 3x3
D1=imdilate(J,D);                   %memperbesar ukuran obyek dalam
gambar
figure, imshow(D1)
title('Dilation with inbuilt function');
imwrite(D1,'baruU_10')              %membaca hasil gambar
```

3.12 Tahap Ekstraksi Fitur

Tahap selanjutnya ekstraksi fitur, tahap untuk mendapatkan fitur pada masing-masing gambar melalui titik koordinat *Zernike* moment. Dalam penelitian ini peneliti menggunakan delapan order pertama zernike moment, yang diperlihatkan pada tabel 3.2. Diagram alir proses dari tahap ekstraksi fitur bisa dilihat pada gambar 3.5.

Tabel 3.2. Delapan order pertama zernike moment

Order	Zernike Moment
0	$A_{0,0}$
1	$A_{1,1}$
2	$A_{2,0}$, $A_{2,2}$
3	$A_{3,1}$, $A_{3,3}$
4	$A_{4,0}$, $A_{4,2}$, $A_{4,4}$
5	$A_{5,1}$, $A_{5,3}$, $A_{5,5}$
6	$A_{6,0}$, $A_{6,2}$, $A_{6,4}$, $A_{6,6}$
7	$A_{7,1}$, $A_{7,3}$, $A_{7,5}$, $A_{7,7}$



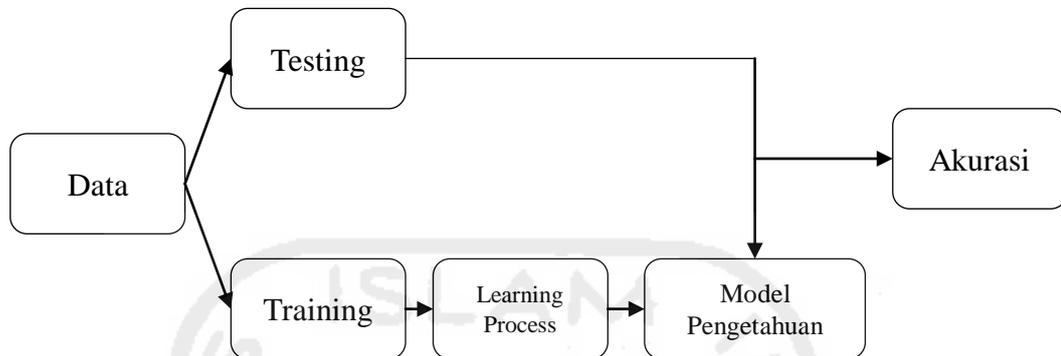
Gambar 3.5 Diagram alir proses tahap ekstraksi fitur

3.13 Klasifikasi JST

Pada proses klasifikasi, data hasil ekstraksi fitur dibagi kedalam dua bagian yaitu data pelatihan (*training*) dan data pengujian (*testing*). Pada data *training* huruf yang akan dilakukan pengenalan pola diberikan kelas 1 dan yang bukan diberikan kelas 0. Algoritme klasifikasi akan melakukan pembelajaran dan membentuk model pengetahuan berdasarkan data *training* tersebut. Model yang dihasilkan oleh algoritme klasifikasi akan diuji cobakan pada data *testing*.

Proses training dan testing menggunakan metode validasi silang sebanyak jumlah data yang diujicobakan yaitu 10 kali percobaan. Nantinya keseluruhan data pernah menjadi data *training* dan pernah menjadi data *testing*. Gambar 3.6

memperlihatkan bagan alir proses klasifikasi dengan menggunakan algoritme JST.



Gambar 3.6 Diagram alir klasifikasi JST

3.14 Akurasi

Evaluasi hasil pengenalan pola huruf hijaiyah dilakukan berdasarkan perhitungan akurasi, dapat dilihat pada persamaan (3.1).

$$Accuracy = \frac{Jumlah\ data\ diprediks\ benar}{Jumlah\ total\ data\ yang\ diprediks\ i} \times 100\% \quad (3.1)$$

Pada penelitian ini juga menerapkan *10-fold cross validation*, dimana pada tiap-tiap *fold* data *training* dan *testing* akan berubah-ubah, sehingga data *training* pernah menjadi *testing* dan sebaliknya data *testing* pernah menjadi *training*. Akurasi yang diambil merupakan rata-rata akurasi pada masing-masing *fold*