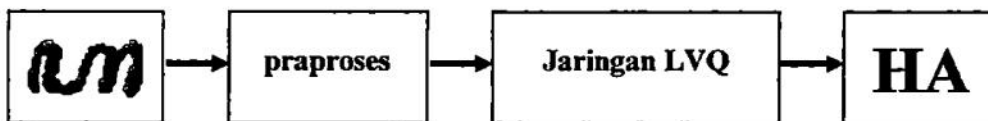


### BAB III

## ANALISA DAN PERANCANGAN SISTEM

### A. GAMBARAN UMUM SISTEM

Secara umum, program yang dibuat digunakan untuk mengenali gambar-gambar huruf jawa carakan (nglegeno). Pengenalan gambar huruf jawa tersebut menggunakan algoritman jaringan syaraf tiruan LVQ. Masukan yang diperlukan adalah menggambar menggunakan *mouse*. Sebelum diolah menggunakan jaringan LVQ, masukan mengalami praproses. Dari sistem tersebut akan menghasilkan keluaran berupa nama huruf yang teridentifikasi dari gambar masukan tersebut.



**Gambar 3.1** Gambaran Secara Umum Proses LVQ

Pengguna dimungkinkan untuk meng-*update* basis data pelatihan LVQ yang terdiri dari penambahan dan penghapusan data.

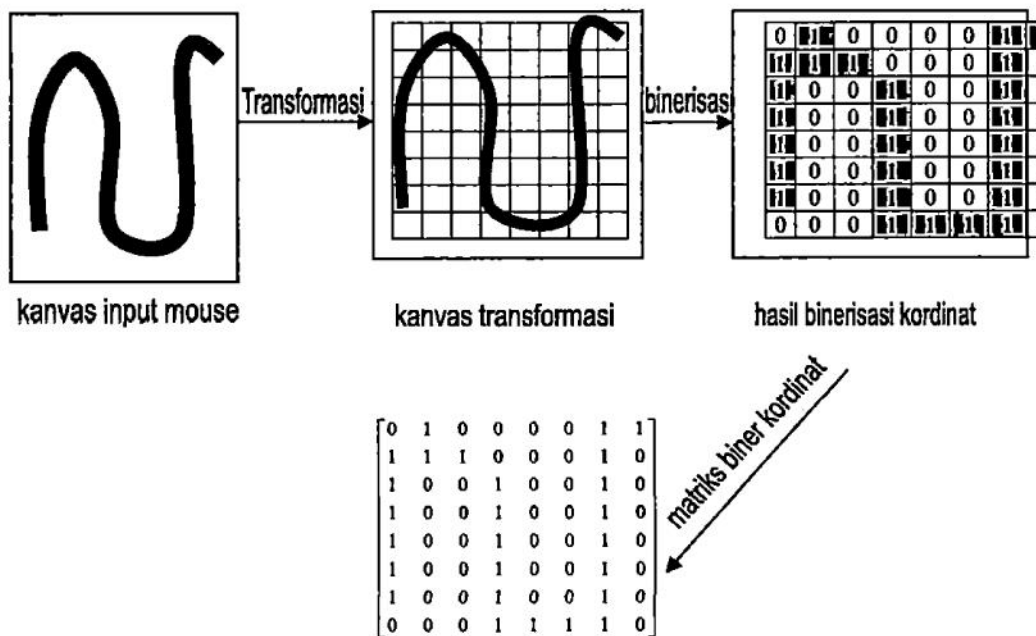
### B. ANALISA SISTEM

Sistem pengenalan pola huruf jawa ini merupakan suatu alat bantu yang digunakan untuk mengidentifikasi masukan huruf jawa yang diberikan. Masukan dalam sistem ini dibatasi hanya pada masukan lewat melukis huruf pada kanvas

dengan menggunakan *mouse*. Jejak – jejak arsiran *mouse* pada kanvas direkam dalam sebuah vektor yang mencatat kordinat titik – titik jejak *mouse* tersebut, kemudian kordinat – kordinat tersebut dihilangkan pinggir putihnya, pinggir putih ditentukan dengan terlebih dulu meng-*capture* atau menangkap gambar hasil arsiran *mouse* dalam sebuah persegi empat yang tepat memuat gambar tersebut, semua kordinat pada kanvas yang tidak terletak dalam persegi empat tersebut didefinisikan sebagai pinggir putih gambar, yang tidak diperlukan, dan dihilangkan pada tahap praproses.

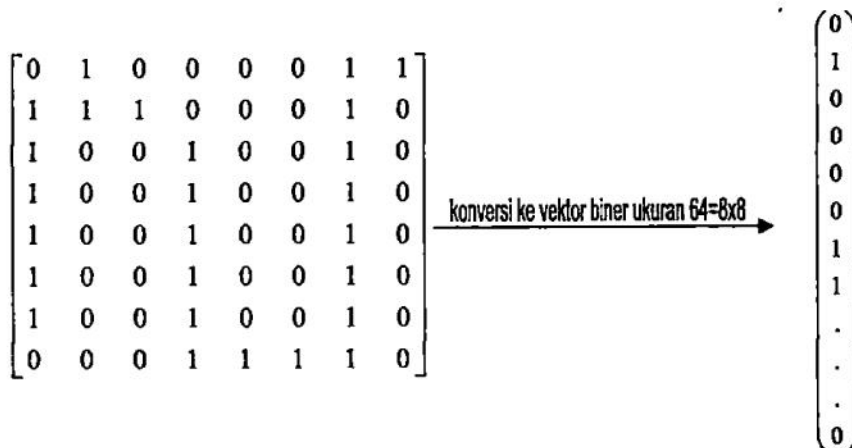
Selanjutnya semua kordinat citra yang terletak dalam persegi empat tersebut ditransformasikan ke dalam kanvas berukuran 10x11 secara tepat (sesuai rasio perbandingan panjang dan lebar persegi empat).

Pada kanvas 10x11, dilakukan binerisasi dengan cara menandai 1 (satu) pada setiap kordinat padanan dari citra pada kanvas 10x11 (kordinat hasil transformasi citra) dan semua kordinat yang bukan padanan citra ditandai dengan 0 (nol). Dengan demikian diperoleh secara lengkap sebuah pola biner yang berupa matriks biner berukuran 10x11.



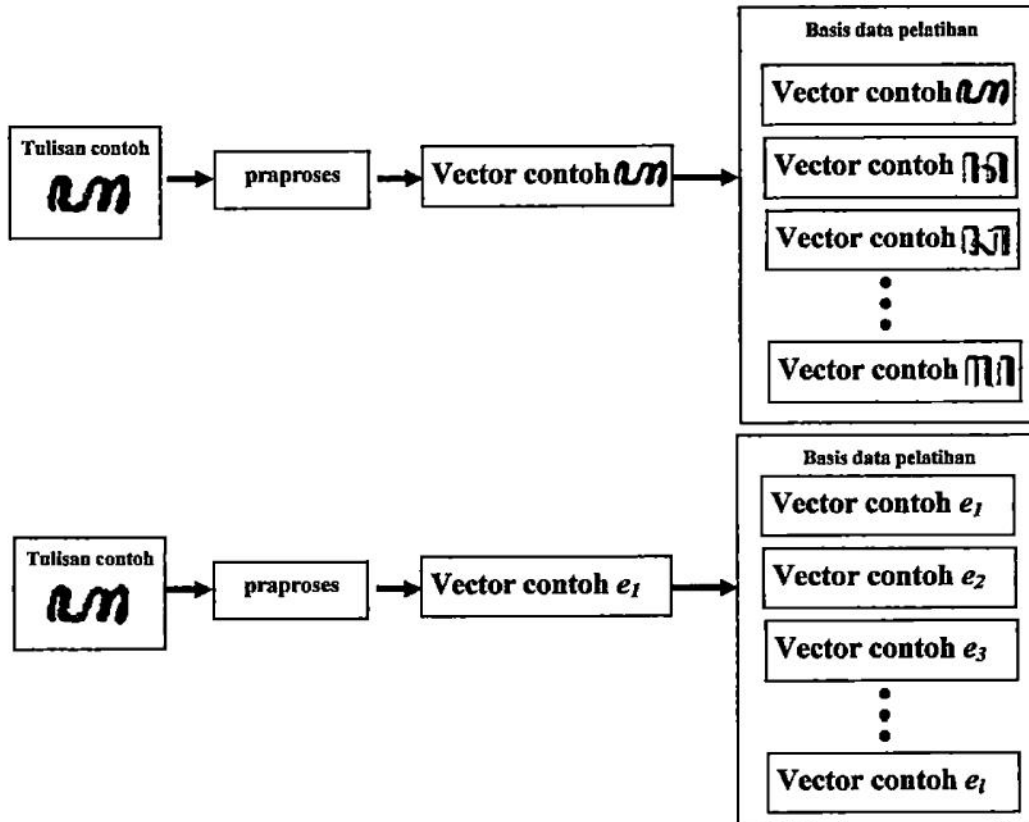
**Gambar 3.2. Ilustrasi matriks biner**

Selanjutnya proses ekstraksi ciri menjadi sederhana hanya dengan mengkonversi matriks biner ke dalam vektor biner yang berukuran  $110 = 10 \times 11$ .



**Gambar 3.3. Ilustrasi konversi matriks biner ke vektor biner**

Vektor biner ini menjadi sebuah vektor masukan bagi proses LVQ selanjutnya. LVQ dalam sistem ini adalah sebuah algoritma jaringan syaraf tiruan. Yang nantinya dipakai untuk mengenali huruf jawa. Untuk mengenali huruf jawa yang dimasukkan, sistem melakukan dua proses yaitu *training* dan *testing*. Pada proses *training*, basis data dibuat dengan menciptakan 20 buah vektor contoh yang merupakan masing – masing vektor contoh menyatakan sebuah huruf jawa (terdapat 20 buah huruf jawa), vektor – vektor contoh ini dibuat dengan memasukkan contoh – contoh tulisan tangan huruf jawa yang benar dan kemudian melewatkannya pada algoritma – algoritma praproses yang mana tulisan contoh itu menjadi bahan dasar untuk menentukan vektor referensi.



**Gambar 3.4. Ilustrasi Secara Umum Proses pelatihan LVQ**

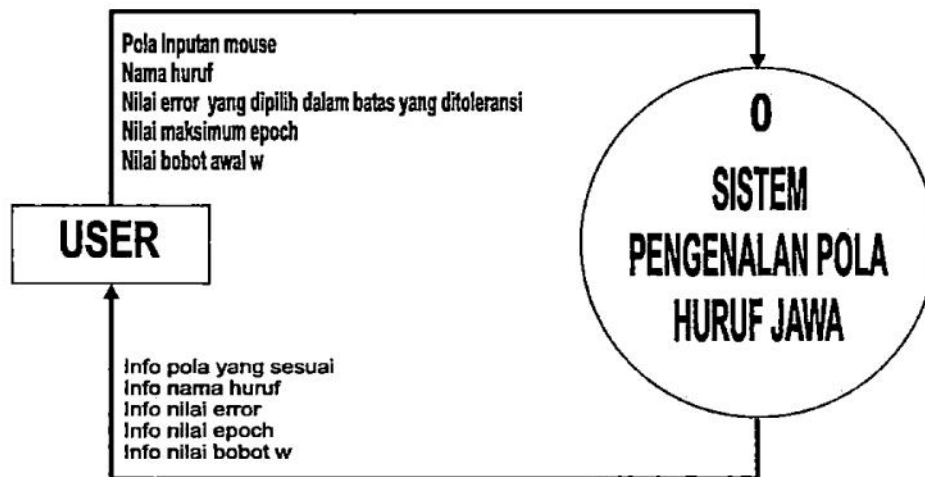
Dimana  $e_i$  terdiri dari  $R$  komponen. Yang merupakan nilai – nilai ekstraksi ciri dari contoh tulisan huruf jawa yang hendak digunakan sebagai basis *training* jaringan LVQ.

## C. PERANCANGAN SISTEM

### 1. Perancangan Proses

#### a) Diagram aliran Data (DAD)

## 1) DAD level 0 / Diagram Konteks

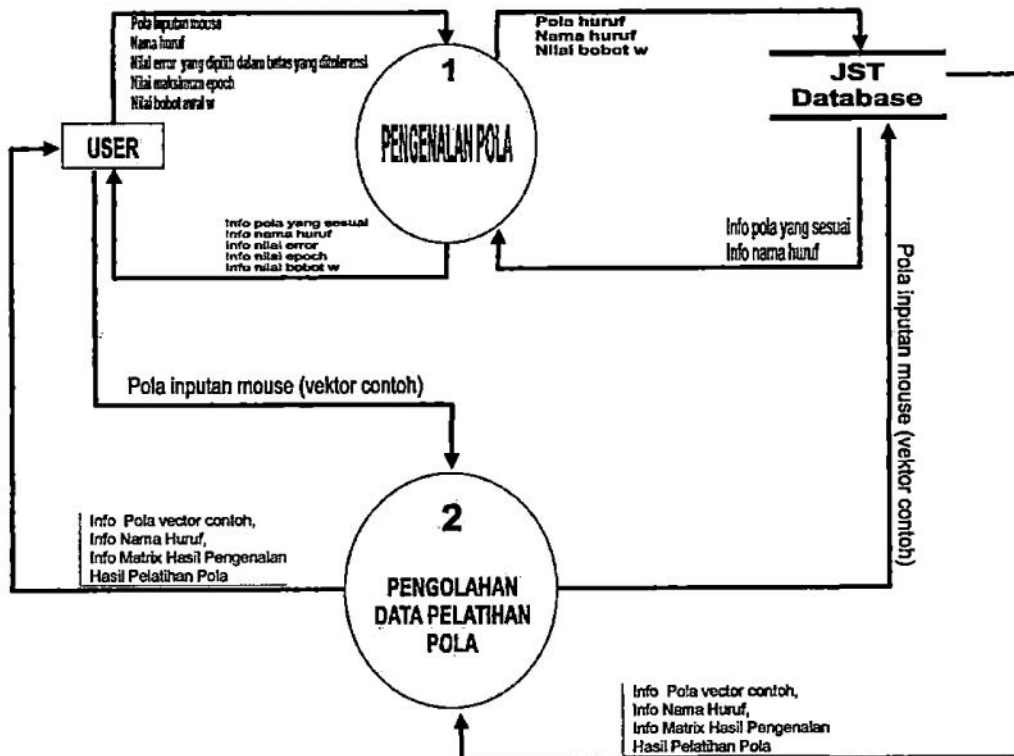


Gambar 3.5. DAD Level 0 Sistem Pengenalan Huruf Jawa

Sistem pengenalan pola huruf jawa membutuhkan masukan berupa pola masukan data pelatihan dari penggambaran *mouse*, nama huruf, nilai  $\varepsilon$  yang dipilih dalam batas yang ditoleransi, nilai maksimum *epoch*, nilai bobot awal  $w_{ij}$ . nilai  $w_{ij}$  merupakan suatu bobot jaringan LVQ yang diuji coba agar diperoleh kedekatan yang paling sesuai bagi masukan kepada data pelatihan yang ada. Data gambar data pelatihan (nilai vektor contoh), nama huruf dan nilai  $w_{ij}$  nantinya akan disimpan dalam *database* sebagai satu kesatuan yang dinamakan *template*. Gambar masukan akan diproses oleh sistem pengenalan pola huruf jawa sehingga menghasilkan nama huruf

dari gambar masukan tersebut dan *error* yang terbentuk saat pengujian.

## 2) DAD level 1



**Gambar 3.6. DAD Level 1 Sistem Pengenalan Huruf Jawa**

Sistem pengenalan pola memiliki dua proses utama yaitu proses pengenalan pola dan proses pengolahan data pelatihan. Proses pengenalan pola berfungsi mengenali atau mengidentifikasi pola arsiran *mouse* sebagai masukan, dan diubah menjadi masukan jaringan syaraf tiruan LVQ. Proses pengenalan pola membutuhkan masukan berupa citra arsiran *mouse* yaitu jejak arsiran pada kanvas oleh *mouse*.

Masukan lainnya adalah data *template* (berupa nilai bobot  $w$  ; nama huruf, nilai *error* dalam batas yang ditoleransi, dan maksimum nilai *epoch / iterasi*). Proses tersebut menghasilkan keluaran berupa nama huruf dan *error* (kesalahan jaringan LVQ). Proses kedua adalah proses pengolahan data pelatihan. Proses tersebut berfungsi untuk mengelola *database template* atau data pelatihan jaringan LVQ. Masukan yang ada berupa citra arsiran *mouse* sebagai data pelatihan atau contoh diekstraksi dengan tahap praproses sehingga diperoleh sebuah vektor contoh, kemudian vektor contoh ini bersama dengan nama huruf dan nilai bobot yang masuk bersamaan *capture* citra arsiran *mouse* data pelatihan yang diekstrak tadi dimasukkan ke dalam *database template*.

### 3) DAD level 2 Proses 1 / Pengenalan Pola

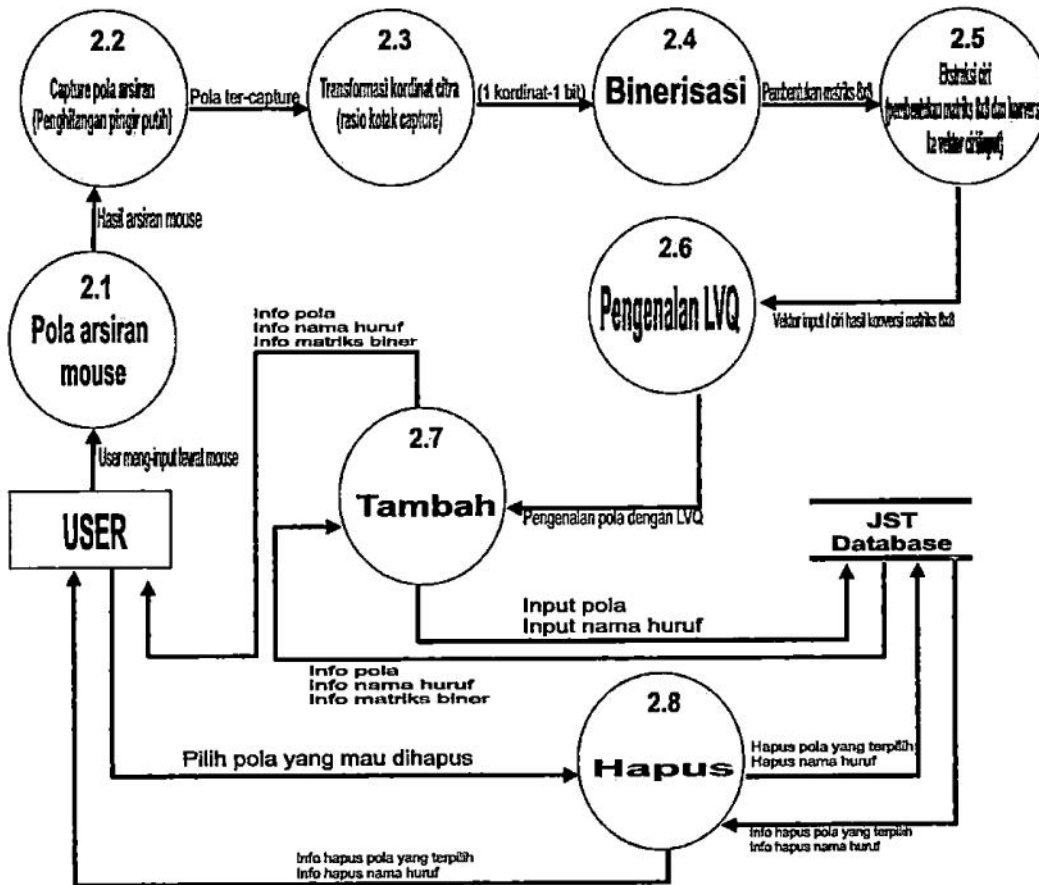
Proses pengenalan pola terbagi menjadi 5 proses yaitu proses masukan pola *mouse* (berupa proses pengarsiran oleh *mouse* terhadap permukaan kanvas), proses penghilangan pinggir putih pola yang tidak terasir *mouse* (*capture* terhadap citra yang dibentuk oleh arsiran *mouse* berupa kotak segi empat tepat menampung citra), proses transformasi kordinat – kordinat citra ke dalam ruang  $10 \times 11$ , proses binerisasi berupa penandaan nilai bit 1 (satu) terhadap kordinat – kordinat padanan citra di ruang  $10 \times 11$  dan nilai bit 0 (nol) terhadap kordinat – kordinat di ruang  $10 \times 11$  yang bukan padanan transformasi citra. Proses ekstraksi ciri berupa pembentukan matriks  $10 \times 11$  yang



#### 4) DAD level 2 Proses 2 / Pengolahan Data Pelatihan

Proses pengolahan data pelatihan terbagi menjadi proses masukan pola *mouse*; proses penghilangan pinggir putih pola, proses transformasi kordinat – kordinat citra, proses binerisasi, proses ekstraksi, proses tambah, proses hapus, dan proses simpan. Proses masukan pola berupa proses pengarsiran oleh *mouse* terhadap permukaan kanvas. Proses penghilangan pinggir putih pola berupa *capture* terhadap citra yang dibentuk oleh arsiran *mouse* berupa kota segi empat tepat menampung citra dan permukaan diluar *capture* dianggap pinggir putih pola dan dihilangkan. Proses transformasi kordinat-kordinat citra adalah transformasi kordinat – kordinat yang membangun citra dan ditransformasikan ke ruang  $10 \times 11$  berdasar rasio panjang lebar kotak *capture* yang tepat menampung citra. Proses binerisasi berupa penandaan nilai bit 1 (satu) terhadap kordinat-kordinat padanan citra di ruang  $10 \times 11$  dan nilai bit 0 (nol) terhadap kordinat – kordinat di ruang  $10 \times 11$  yang bukan padanan transformasi citra. Proses ekstraksi ciri berupa pembentukan matriks  $10 \times 11$  yang merepresentasikan bit – bit pada ruang  $10 \times 11$  dan konversi terhadap matriks tersebut ke vektor yang berukuran  $110=10 \times 11$ . Proses tambah membutuhkan masukan berupa nilai bobot  $w_j$ , nama huruf, dan vektor contoh (vektor yang merupakan hasil ekstraksi pola arsiran *mouse*). Proses tambah akan membuatkan *record* baru yang digunakan

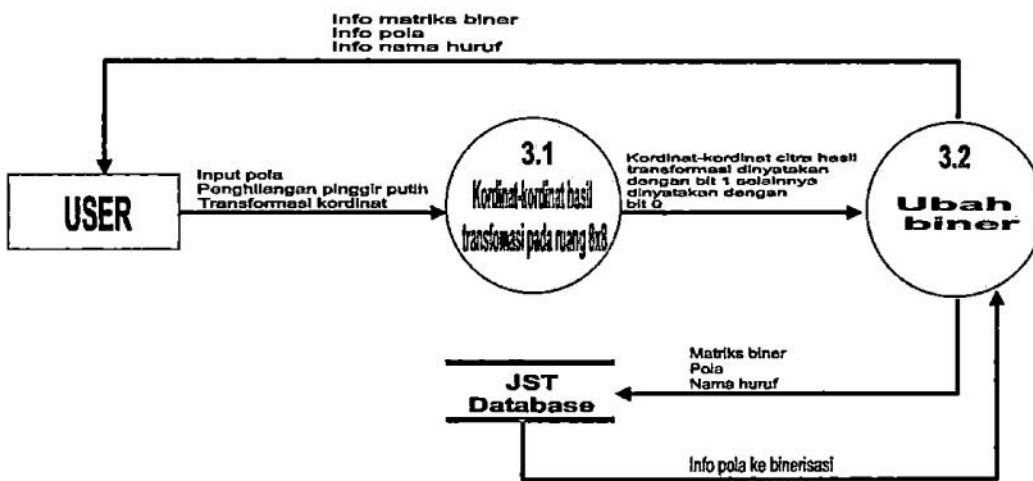
untuk menampung data – data masukan. Proses simpan berfungsi untuk menyimpan data ke *database template* dimana data dari proses tambah. Proses hapus akan mengambil data dari *database template* dimana proses hapus akan menghapus data yang terseleksi dan menampilkan pesan.



Gambar 3.8. DAD Level 2 Pengolahan pada Sistem Pengenalan Huruf Jawa

### 5) DAD level 3 proses 1 / Binerisasi

Dalam proses binerisasi pada sistem ini, dilakukan dengan prinsip yang sederhana, yaitu memetakan setiap kordinat citra hasil transformasi dalam ruang 10x11 bernilai bit 1 (satu) dan memetakan setiap kordinat pada ruang 10x11 yang bukan merupakan citra, bernilai bit 0 (nol).

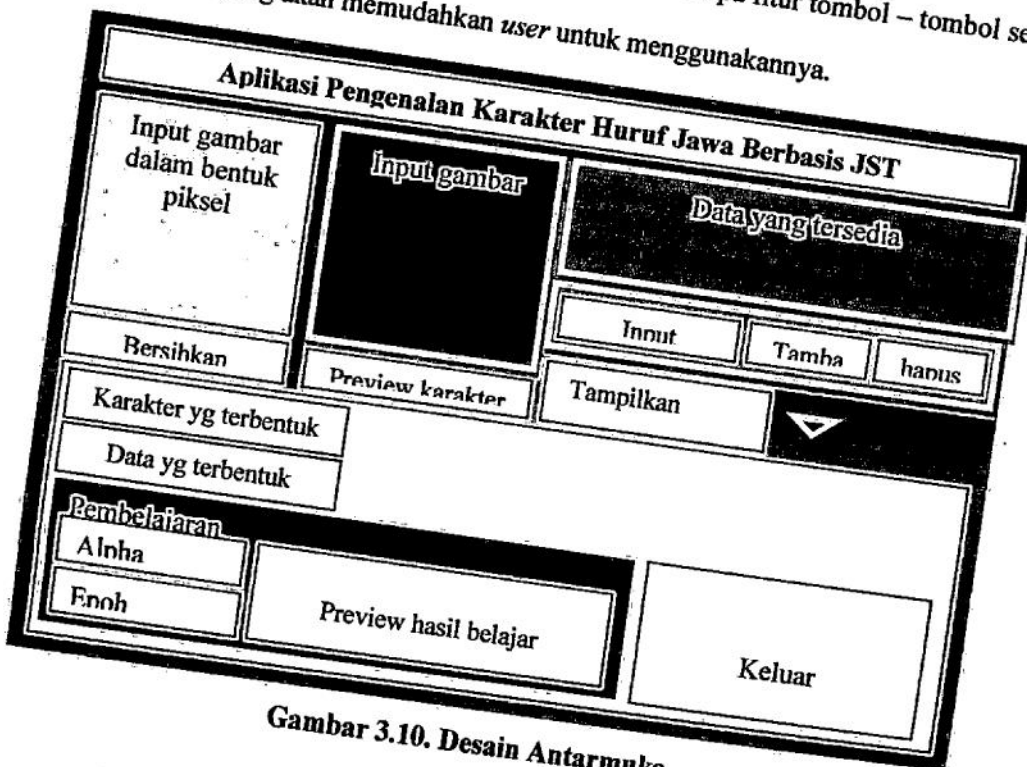


**Gambar 3.9. DAD Level 2 pada proses 1 binerisasi pada Sistem Pengenalan Huruf Jawa**

### D. PERANCANGAN ANTARMUKA (USER INTERFACE)

Perancangan antarmuka untuk aplikasi pengenalan karakter huruf Jawa ini pada dasarnya akan dibuat dengan memperhatikan beberapa hal diantaranya yaitu tampilan masukan yang dibuat harus mudah digambar oleh *user* serta tampilan keluaran yang dihasilkan harus mudah dipahami oleh penerima atau *user*.

Aplikasi pengenalan karakter huruf Jawa ini akan muncul ketika program dijalankan dan akan menampilkan tampilan utama. Tampilan utama merupakan tempat para *user* yang menyediakan beberapa menu untuk menjalankan aplikasi ini. Menu utama dari aplikasi ini terdiri dari beberapa fitur tombol – tombol serta fasilitas yang akan memudahkan *user* untuk menggunakannya.



Gambar 3.10. Desain Antarmuka

Pada tampilan utama, terdapat tiga area yaitu area masukan, area keluaran dan area pembelajaran. Area masukan terdapat dua layar gambar, tombol bersihkan yang berfungsi untuk menghapus masukan gambar, tombol lihat karakter serta tombol tambah. Untuk area keluaran terdiri dari layar data yang tersedia yang akan menampilkan semua data yang tersimpan di dalam *database*,

layar *input* karakter, layar karakter yang terbentuk yang menampilkan hasil dari *inputan* gambar, tombol hapus serta *combobox* tampilan yang berfungsi memasukkan nama karakter dimana tersedia karakter – karakter yang dapat dipilih oleh *user*. Pada area pembelajaran terdiri dari layar data yang terbentuk, layar pembelajaran yang terdiri dari dua jendela yaitu *Epo*h dan *Alpha*, tombol lihat hasil pembelajaran serta tombol keluar.

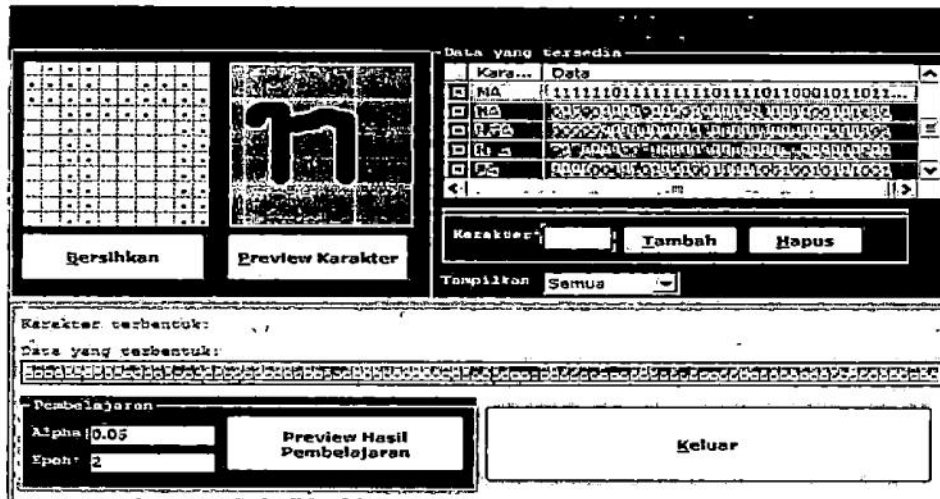
## **E. PENGUJIAN SISTEM**

### **1. Data Pengujian**

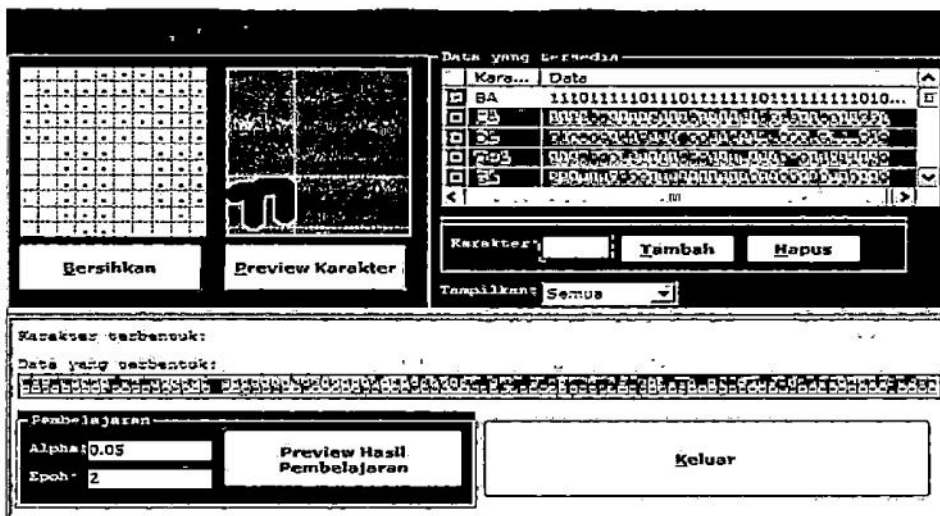
Data yang digunakan dalam pengujian sistem ini berupa masukan gambar dari arsiran *mouse*. Sebelumnya, pada basisdata sistem telah dimasukkan 20 buah vektor contoh yang tiap – tiap vektor contoh tersebut menyatakan sebuah huruf Jawa (terdapat 20 huruf Jawa). Vektor contoh ini dibuat dengan memasukkan contoh tulisan tangan huruf Jawa yang benar dan kemudian melewatkannya pada algoritma – algoritma praproses yang nantinya vektor contoh tersebut akan menjadi bahan dasar untuk menentukan vektor referensi.

### **2. Proses dan Hasil Pengujian**

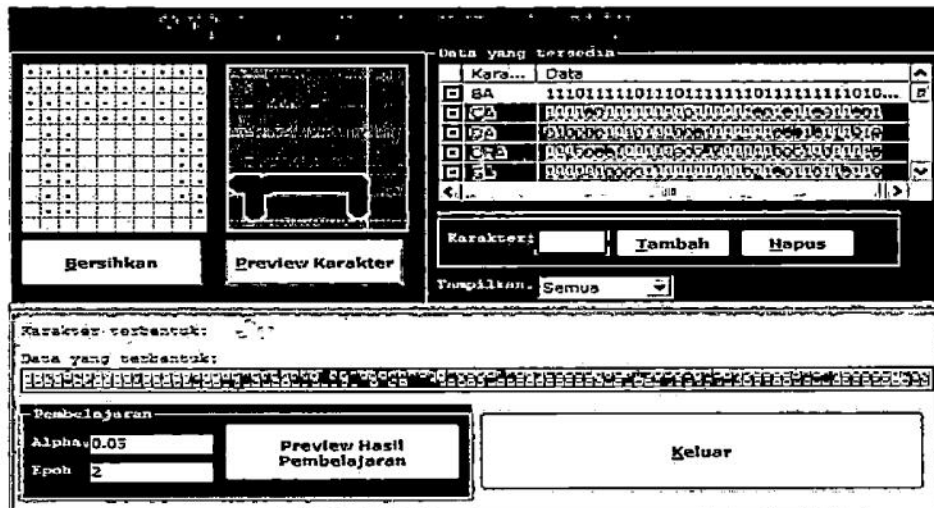
Proses pengujian dilakukan dengan menguji jaringan syaraf untuk mengenali sebuah masukan pola pada kanvas. Diambil satu buah contoh tulisan huruf Jawa Ra yang berbeda dengan pengujian sebanyak 5 kali.



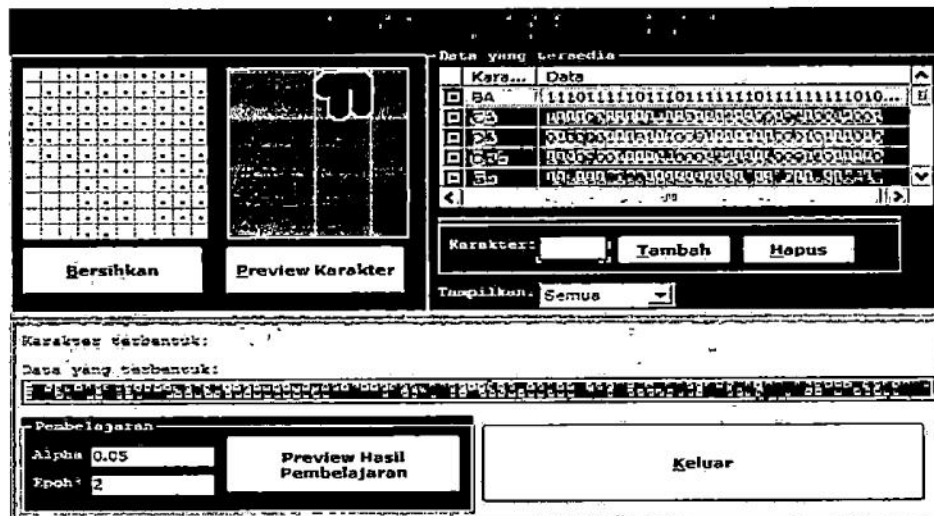
Gambar 3.11. Pengujian 1



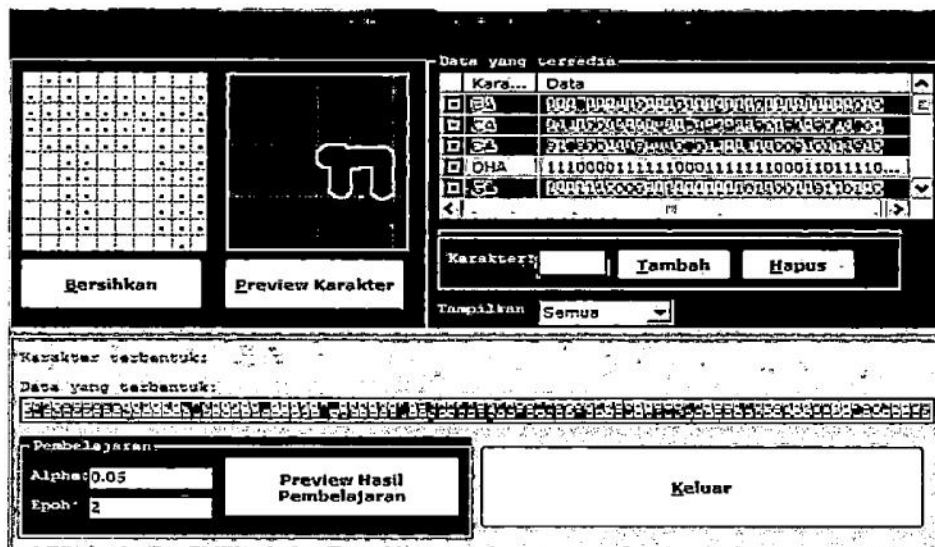
Gambar 3.12. Pengujian 2



Gambar 3.13. Pengujian 3



Gambar 3.14. Pengujian 4



**Gambar 3.15. Pengujian 5**

Gambar – gambar diatas menunjukkan hasil pelatihan terhadap 1 buah huruf Jawa dengan cara tulisan yang berbeda – beda sebanyak 5 kali pengujian. Setelah dilakukan pengujian terhadap huruf tersebut, sistem tidak sempurna dalam mengenali huruf RA. Dari 5 kali pengujian, terdapat 2 kali kesalahan dalam mengenali huruf tersebut. Hal itu terjadi karena perhitungan jarak antara vektor masukan dan vektor referensi cenderung besar atau tidak mendekati sama.

Berikut penjelasan perhitungan jarak dan bobot vektor referensi huruf Ra serta vektor masukan yang akan diuji :

### 2.1. Perhitungan bobot vektor referensi

Untuk melakukan perhitungan ini, digunakan perangkat lunak bantu yaitu Microsoft Excel agar mempermudah dan mempercepat proses perhitungan.



Mulanya, masukkan nilai – nilai vektor dari 20 buah vektor referensi ke dalam kotak yang tersedia. Gambar dibawah ini hanya menampilkan hasil dari input vektor referensi yang sesuai dengan contoh huruf yang ada, yaitu huruf Ra yang berada pada kolom “referensi 15”.

	FH	FI	FJ	FK	FL	FM	FN	FO	FP	FR	FS	FT	FU	FV	FW	FX	FY	FZ	GA	GB	GC	GD	GE	GF	GG	GH	GI	GJ	GK	GL	GM	GN	GO	GP	GQ	GR	GS	GT	GU	GV	GW	GX					
1																																															
2																																															
3																																															
4																																															
5																																															
6																																															
7																																															
8																																															
9																																															
10																																															
11																																															
12																																															
13																																															
14																																															
15																																															
16																																															
17																																															
18																																															
19																																															
20																																															
21																																															
22																																															
23																																															
24																																															
25																																															
26																																															
27																																															
28																																															
29																																															
30																																															
31																																															
32																																															
33																																															

Gambar 3.16. Input Vektor Referensi Ra

Kemudian hasil dari inputan vektor tersebut dimasukkan kedalam kotak vektor bobot, ditunjukkan pada gambar dibawah :

	R	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	
25																	
26																	
27																	
28																	
29																	
30																	
31																	
32																	
33																	
34																	

Gambar 3.17. Input Vektor Referensi Ra Menjadi Vektor Bobot

## 2.2. Perhitungan jarak dan bobot vektor masukan

Setelah ke-20 vektor referensi telah dimasukkan, akan diuji kedekatan jarak salah satu vektor referensi tersebut dengan beberapa vektor contoh yang nantinya akan dilatih. Seperti yang telah dijelaskan sebelumnya, vektor contoh diambil berdasarkan pengujian huruf Ra yang berbeda maka hanya pengujian yang pertama dan kelima

- *Pengujian 1*

Sebagai nilai awal ditentukan learning rate ( $\alpha = 0.05$ ), dengan pengurangan sebesar  $0.1 \cdot \alpha$ ; dan maksimum epoch = 2. Masukkan nilai vektor contoh pada kotak input pola, seperti gambar dibawah:



**Gambar 3.18. Input Vektor Contoh Pengujian 1**

Nilai – nilai vektor tersebut diisi sesuai dengan nilai – nilai yang terdapat pada kanvas berbentuk piksel. Selanjutnya, nilai vektor bobot referensi akan dikurangi dengan nilai vektor contoh sehingga diperoleh selisih nilai yang nanti akan digunakan untuk menghitung jarak terdekat (disebut sebagai  $C_j$ ).

Persamaan untuk menentukan selisih :

Selisih (J) = vektor bobot referensi – vektor contoh

$$= \| w_j - x \|$$

Sedangkan untuk menghitung jarak dan menentukan jarak terdekat, persamaannya :

$$\text{Jarak pada bobot vektor } W_n = \sqrt{\sum (w_j - x)^2}$$

The screenshot shows a spreadsheet with columns labeled AI, AJ, AK, AL, AM, AN, AO, AP, AQ, AR, AS, AT, AU, AV, AW, AX, AY, AZ. The rows are numbered 1 to 34. The content of the cells consists of vertical bars of varying lengths and patterns, representing vector differences. The status bar at the bottom indicates 'learning vector quantization' and 'masukkan vektor referensi'.

Gambar 3.19. Selisih Vektor Contoh Pengujian 1

Dengan menghitung jarak pada tiap – tiap bobot vektor maka akan diperoleh jarak yang paling dekat sesuai dengan hasil perhitungan.

The screenshot shows a spreadsheet with columns labeled AG, AH, AI, AJ, AK, AL, AM, AN, AO, AP, AQ, AR, AS, AT, AU, AV, AW, AX. The rows are numbered 94 to 127. The content includes numerical values and logical expressions. For example, row 120 has 'FALSE' in columns AI through AV, and 'TRUE' in column AX. Row 121 has 'FALSE' in columns AI through AV, and '15' in column AX. The status bar at the bottom indicates 'learning vector quantization' and 'masukkan vektor referensi'.

Gambar 3.20. Hasil Vektor Contoh Pengujian 1

Dari gambar diatas, vektor referensi ke-15 merupakan jarak yang terdekat antara vektor contoh dan vektor bobot. Jika telah diketahui jarak terdekatnya maka perbaiki bobotnya. Masukkan nilai vektor bobot kedalam kotak input bobot.

	BC	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS
1																
2																
3						0.25				1.00						
4						1										
5						1										
6						1										
7						0.25				1.00						
8						0.25				1.00						
9						0.25				1.00						
10						0.25				1.00						
11						0.25				1.00						
12						0				0						
13						1										
14						1										
15						1										
16						1										
17						0.25				1.00						
18						0.25				1.00						
19						1										
20						1										
21						1										
22						1										
23						0.25				-0.05						
24						0.25				-0.05						
25						1										
26						1										
27						1										
28						1										
29						0.25				-0.05						
30						0.25				-0.05						
31						1										
32						1										
33						0				0						
34						0				0						

**Gambar 3.21. Pengujian Nilai Bobot Vektor Contoh Pengujian 1**

Perhitungan nilai bobot ditentukan oleh 2 hal, yaitu jika vektor contoh dapat dikenali dengan baik maka :

$$\downarrow T = C_j \text{ maka } w_j(\text{baru}) = w_j(\text{lama}) + \alpha (x - w_j(\text{lama}))$$

jika vektor contoh tidak dapat dikenali dengan baik, maka :

$$\downarrow T \neq C_j \text{ maka } w_j(\text{baru}) = w_j(\text{lama}) - \alpha (x - w_j(\text{lama}))$$

- **Pengujian 2**

Langkah – langkah pengujian 2 ini hampir sama dengan yang pertama, perbedaannya terletak pada nilai vektor dan hasil yang akan diperoleh.

Dengan memberikan contoh tulisan huruf Jawa yang sedikit berbeda, sistem pengenalan huruf ini diharapkan dapat mengenali dengan sempurna.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1																								
2																								
3																								
4																								
5																								
6																								
7																								
8																								
9																								
10																								
11																								
12																								
13																								
14																								
15																								
16																								
17																								
18																								
19																								
20																								
21																								
22																								
23																								
24																								
25																								
26																								
27																								
28																								
29																								
30																								
31																								
32																								
33																								
34																								

Gambar 3.22. Input Vektor Contoh Pengujian 2

	AI	AI	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ
97																		
98																		
99																		
100																		
101																		
102																		
103																		
104																		
105																		
106																		
107																		
108																		
109																		
110																		
111																		
112																		
113																		
114																		
115																		
116																		
117																		
118																		
119	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
120																		
121																		
122																		
123																		
124																		
125																		
126																		
127																		
128																		
129																		
130																		

Gambar 3.23. Hasil Vektor Contoh Pengujian 2

Dari gambar diatas, menunjukkan adanya ketidaktepatan sistem pengenalan ini mengolah vektor masukan. Walaupun huruf yang digambarkan hanya memiliki sedikit perbedaan, namun jarak yang terhitung tidak diperoleh hasil yang akurat.

	BC	BD	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT
1																		
2																		
3							0.05				1.00							
4							1											
5							0				0							
6							0.05				1.00							
7							0.05				1.00							
8							0.05				1.00							
9							0.05				1.00							
10							0.05				1.00							
11							0.05				1.00							
12							0											
13							1											
14							1											
15							1											
16							0.05				-0.05							
17							1											
18							1											
19							1											
20							1											
21							1											
22							1											
23							1											
24							0.05				-0.05							
25							1											
26							1											
27							1											
28							1											
29							1											
30							1											
31							1											
32							1											
33							0.05				-0.05							
34							0.05				-0.05							

Gambar 3.24. Pengujian Nilai Bobot Vektor Contoh Pengujian 2

Pengujian bobot dilakukan dengan cara memasukkan nilai vektor bobot yang terpilih, yang dimaksud disini adalah vektor bobot 1. Lalu dengan persamaan yang telah terurai diatas, untuk  $T \neq C_j$  maka  $w_j(\text{baru}) = w_j(\text{lama}) - \alpha (x - w_j(\text{lama}))$ . Proses ini berlangsung selama 2 kali maksimal epoch dan kemungkinan hasilnya pun tidak dapat diprediksi apakah nantinya akan mendekati vektor referensi yang sesuai.

Adapun alasan – alasan yang menyebabkan ketidakakuratan sistem pengenalan ini, antara lain :

1. Kerapatan piksel pada kanvas gambar yang menyebabkan ketidaktepatan pembacaan nilai pada vektor referensi ataupun vektor contoh.
2. Penentuan maksimal epoch dengan jumlah terbatas sehingga proses pembelajaran tidak maksimal dan proses pengenalan tidak akurat.
3. Ruang menggambar yang kurang leluasa akibatnya kesalahan menggambar.
4. Penggambaran vektor – vektor referensi yang menjadi acuan atau target keluaran tidak diberikan kepada ahli sastra Jawa.