
PENGEMBANGAN APLIKASI PENGENAL KARAKTER MENGGUNAKAN METODE *HETEROASSOCIATIVE MEMORY*

¹Alvin Lius, ²Emma R. Simarmata, ³Herman

^{1,3}STMIK Time Medan, ²Universitas Methodist Indonesia

DOI: <https://doi.org/10.46880/jmika.Vol1No1.pp10-21>

ABSTRAK

Salah satu contoh pengenalan pola yang cukup kompleks dalam bidang studi kecerdasan buatan adalah bagaimana membuat mesin mampu mengenali sebuah gambar karakter atau coretan tulisan tangan. Untuk mengenali pola karakter pada skripsi ini menggunakan metode *Heteroassociative Memory*. Proses pelatihan menghasilkan nilai bobot yang akan disimpan ke *database*. Nilai bobot ini akan digunakan pada fase pengenalan. Ukuran ekstraksi ciri yang digunakan adalah 10 x 10 atau 100 *bit* bipolar. *Bit-bit* ini akan mewakili suatu coretan karakter di dalam aplikasi. Hasil pengenalan terhadap pola karakter yang lebih baik dengan metode *Heteroassociative Memory* harus melakukan proses pelatihan yang sesering mungkin.

Kata Kunci: *Pengenalan Karakter, JST, Heteroassociative Memory*

PENDAHULUAN

Latar Belakang

Kecerdasan buatan berasal dari bahasa Inggris "*Artificial Intelligence*" atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Dalam hal ini, kecerdasan buatan merujuk pada mesin yang mampu berpikir, menimbang tindakan yang diambil dan mampu mengambil keputusan seperti yang dilakukan oleh manusia. Salah satu contoh aplikasi pengenalan pola yang cukup kompleks dalam bidang studi AI adalah pengenalan karakter. Masalah yang dihadapi adalah bagaimana membuat mesin mampu mengenali sebuah gambar karakter atau coretan tulisan tangan dan menerjemahkannya ke bentuk pola karakter. Fungsi pengenalan karakter ini dapat dikembangkan dan diimplementasikan dalam *softwarescanner* dan *smartphone*, untuk mendukung kecepatan dalam melakukan *input* karakter atau mengetik. Permasalahan pengenalan

karakter ini diselesaikan dengan mengimplementasikan jaringan syaraf tiruan.

Jaringan Syaraf Tiruan (JST) dapat digunakan untuk mengenali pola karakter yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang yang sudah beberapa waktu lalu tidak dijumpainya dan wajah atau bentuk tubuh sudah sedikit berubah. Karakter dapat berupa huruf kecil, huruf besar dan angka. Salah satu metode JST yang dapat digunakan untuk melakukan pengenalan terhadap pola karakter adalah metode *Heteroassociative Memory*. Jaringan syaraf *associative memory* adalah jaringan yang bobot-bobotnya ditentukan sedemikian rupa sehingga jaringan tersebut dapat menyimpan pengelompokan pola. Masing-masing kelompok merupakan pasangan vektor ($s(p)$, $t(p)$). Tiap-tiap vektor $s(p)$ memiliki n komponen dan setiap vektor $t(p)$ memiliki m komponen. Bobot-bobot tersebut dapat ditentukan dengan menggunakan *Hebb rule*. Untuk mengenali karakter tulisan, semua pola

karakter akan melalui fase *training* (pelatihan). Pada fase ini, semua pola karakter akan dipelajari oleh metode *Heteroassociative Memory*. Setelah itu, pola karakter yang telah dipelajari akan dapat dikenali melalui fase *recognition* (pengenalan).

Tujuan dari penelitian ini adalah:

1. Untuk memahami cara kerja metode *Heteroassociative Memory* dalam melakukan pengenalan terhadap karakter.
2. Untuk mengembangkan suatu perangkat lunak yang dapat mengenali karakter yang ditulis dengan menggunakan *mouse*.

Analisis Kebutuhan Sistem

Analisis kebutuhan sistem melakukan proses identifikasi dan evaluasi terhadap permasalahan yang ada, sehingga sistem yang dibangun sesuai dengan kriteria yang diharapkan. Sistem yang akan dibangun harus memenuhi kebutuhan fungsional sebagai berikut:

1. Sistem harus mampu melakukan proses *input* sebagai berikut:
 - a. Pengguna dapat menuliskan karakter tunggal (huruf atau angka) dengan menggunakan *mouse* pada area penulisan. Hasil coretan karakter berwarna hitam, sedangkan latar berwarna putih.
 - b. Pengguna dapat menghapus / mengosongkan kembali area penulisan.
2. Sistem harus mampu melakukan proses pelatihan dan pengenalan terhadap pola karakter, sebagai berikut:
 - a. Sistem dapat melakukan *cropping*, untuk membuang tepi kosong di sekitar coretan karakter dan melakukan proses ekstraksi ciri terhadap karakter dengan pola 10 x 10, sebelum dilakukan proses pelatihan dan proses pengenalan.
 - b. Sistem dapat melakukan proses pelatihan terhadap karakter dengan menggunakan metode *Heteroassociative Memory*, dan menampilkan tahapan perhitungannya.

- c. Sistem dapat melakukan pengenalan terhadap karakter dengan menggunakan metode *Heteroassociative Memory*, menampilkan proses perhitungan dan menampilkan nama karakter hasil pengenalan.

3. Sistem harus dapat menampilkan semua karakter yang sudah pernah dilatih dan disimpan sebelumnya ke *database*.

Proses Ekstraksi Ciri

Proses ekstraksi ciri akan mengubah gambar huruf menjadi *bit* digital, dalam penelitian ini yaitu 100 *bit* bipolar (1/-1). Algoritma dari proses ekstraksi ciri terhadap gambar karakter menjadi 100 *bit* bipolar dapat dituliskan sebagai berikut:

1. Pada gambar hitam-putih, periksa dari kiri ke kanan. Apabila suatu kolom mempunyai piksel hitam, maka set $X1 = \text{nomor} / \text{posisi kolom}$.
2. Periksa dari kanan ke kiri. Apabila suatu kolom mempunyai piksel hitam, maka set $X2 = \text{nomor} / \text{posisi kolom}$.
3. Periksa dari atas ke bawah. Apabila suatu baris mempunyai piksel hitam, maka set $Y1 = \text{nomor} / \text{posisi baris}$.
4. Periksa dari bawah ke atas. Apabila suatu baris mempunyai piksel hitam, maka set $Y2 = \text{nomor} / \text{posisi baris}$.
5. *Crop* gambar dari posisi ($X1, Y1$) sampai posisi ($X2, Y2$). Hasil *crop* adalah coretan huruf / karakter.
6. Hasil *crop* kemudian dibagi menjadi 10 x 10 kotak.
7. Periksa setiap area kotak, apabila terdapat lebih banyak piksel hitam daripada piksel putih, maka set warna kotak ke hitam. Bila tidak, maka set warna kotak ke putih.
8. Lakukan proses nomor-7 pada semua kotak yang berjumlah 100 buah (10 x 10), sehingga semua kotak mempunyai warna hitam / putih.

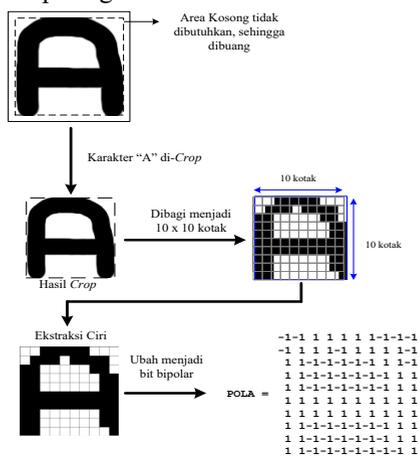
9. Lakukan digitalisasi *bit* bipolar untuk setiap kotak. Kotak yang berwarna hitam diwakili oleh nilai 1, sedangkan kotak yang berwarna putih diwakili oleh nilai -1.
10. Hasil ekstraksi ciri adalah barisan *bit* bipolar -1 atau 1 sebanyak 100 digit.

Sebagai contoh, misalkan huruf / karakter yang akan diekstraksi ciri seperti terlihat pada gambar 1 berikut.



Gambar 1. Huruf "A"

Proses ekstraksi ciri dilakukan dengan membagi citra huruf menjadi 10 x 10 kotak. Kemudian untuk setiap kotak, dihitung jumlah piksel hitam dan jumlah piksel putih yang berada di dalam area kotak. Bila jumlah piksel hitam lebih banyak daripada jumlah piksel putih di dalam suatu area kotak, maka kotak tersebut akan diberikan warna hitam. Sebaliknya, maka kotak diberikan warna putih. Proses ekstraksi ciri dapat dilihat pada gambar 2. berikut.



Gambar 2. Proses Ekstraksi Ciri Terhadap Huruf "A"

Hasil ekstraksi ciri dari citra karakter berupa 100 *bit* bipolar, merupakan representasi citra di dalam *database*. Selanjutnya, hasil ekstraksi ciri ini dilanjutkan ke fase pelatihan atau pengenalan dari metode *Heteroassociative Memory*.

Proses Pelatihan

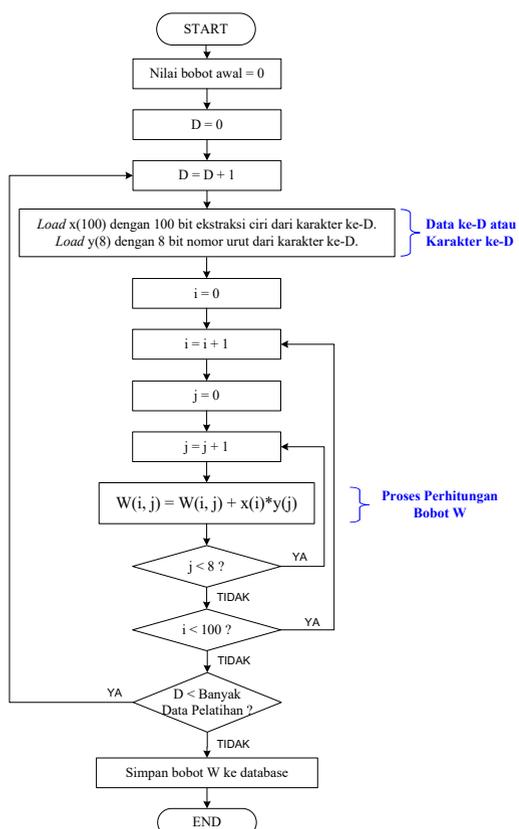
Pada proses pelatihan, pola karakter baru dimasukkan dan dipelajari oleh sistem. Hasil dari proses pelatihan adalah nilai bobot pelatihan yang akan digunakan pada proses pengenalan.

Algoritma *Heteroassociative Memory* yang digunakan untuk proses pelatihan adalah sebagai berikut:

1. Nilai awal semua bobot dari $w(1, 1)$ hingga $w(100, 8)$ adalah 0.
2. Untuk $D=1$ sampai jumlah data yang akan dilatih, lakukan proses berikut.
 - a. $Loadx(100)$ dengan 100 *bit* hasil ekstraksi ciri (*input*) dari karakter ke-D.
 - b. $Loady(8)$ dengan 8 *bit* nomor urut (*output*) dari karakter ke-D.
 - c. Lakukan proses pelatihan berikut:

Untuk $i = 1$ sampai 100,
 Untuk $j = 1$ sampai 8,
 Hitung : $W(i, j) = W(i, j) + x(i)*y(j)$
3. Simpan semua bobot W ke *database*.

Secara umum, proses pelatihan untuk mempelajari pola karakter baru dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar 3 berikut.



Gambar 3. Flowchart Proses Pelatihan

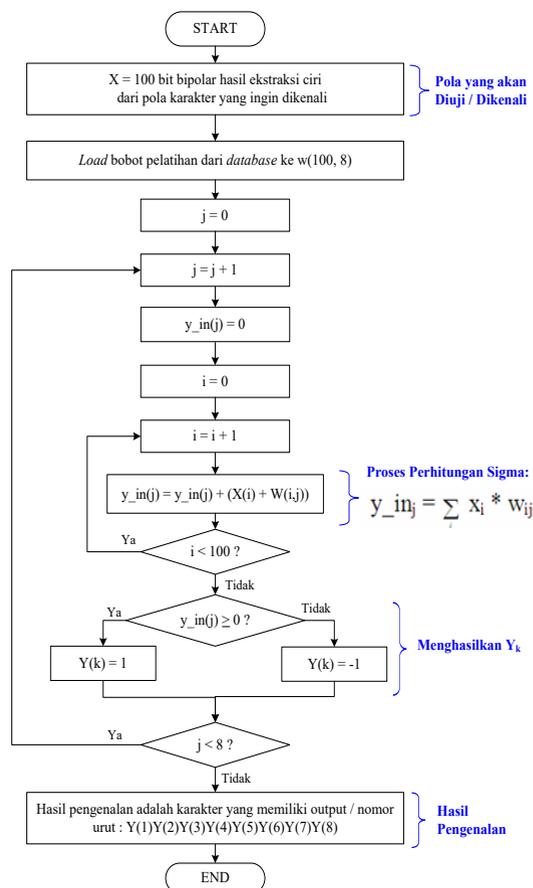
Proses Pengenalan

Algoritma dari proses pengenalan untuk menguji pengenalan pola karakter adalah sebagai berikut

1. Load x(100) dengan 100 bit bipolar hasil ekstraksi ciri dari pola karakter yang ingin dikenali.
2. Load bobot pelatihan dari database ke w(100, 8)
3. Untuk j=1 sampai 8,
 - a. $y_in(j) = 0$
 - b. Untuk i=1 sampai 100, hitung:
 $y_in(j) = y_in(j) + (X(i) + W(i,j))$
 - c. Jika $y_in(j) \geq 0$, maka $y(j) = 1$.
 - d. Jika $y_in(j) < 0$, maka $y(j) = -1$.
4. Urutkan dari y(1), y(2) sampai y(8)
5. Hasil pengenalan adalah karakter yang

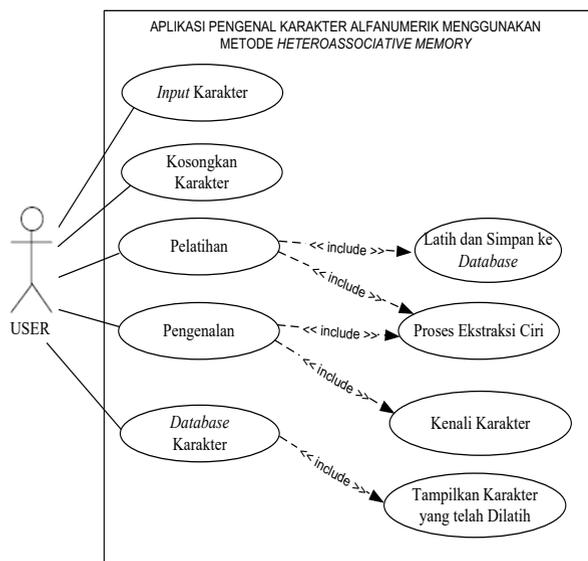
memiliki nomor urut sama dengan 8 bit bipolar berikut: y(1) y(2) y(3) y(4) y(5) y(6) y(7) y(8).

Flowchart dari proses pengenalan untuk mengenali pola karakter dapat dilihat pada gambar 4 berikut.



Gambar 4. Flowchart Proses Pengenalan

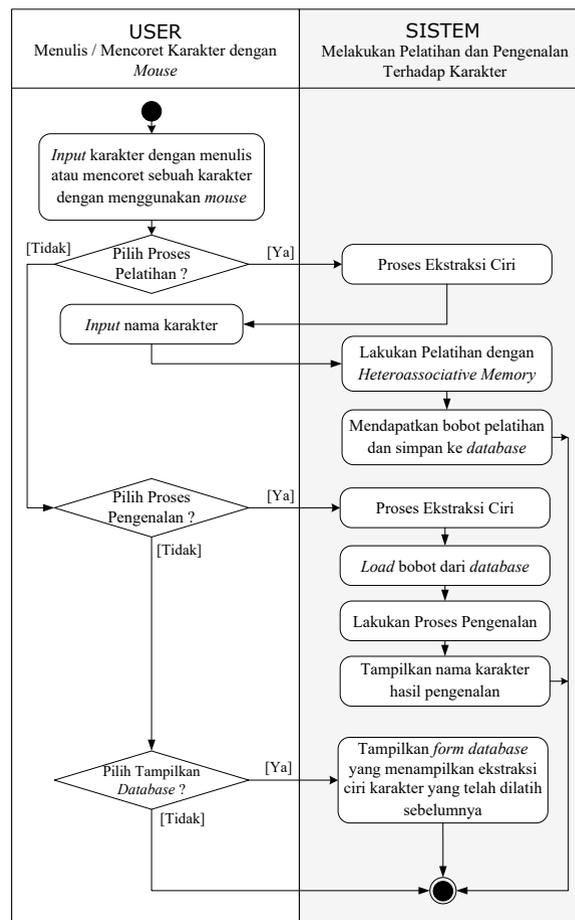
Sebagai contoh pada proses pengenalan, misalkan pola yang ingin dikenali, seperti terlihat pada gambar 5 berikut.



Gambar 7. Diagram Use Case dari Aplikasi

Pada gambar 6, use case diagram menggambarkan apa yang akan dilakukan oleh sistem. Dalam notasi Use Case, hubungan “include” antar use case, berarti use case X menggunakan use case Y sepenuhnya.

Proses yang terjadi di dalam aplikasi juga dapat digambarkan dengan activity diagram seperti terlihat pada gambar 8 berikut.



Gambar 8. Activity Diagram dari Sistem

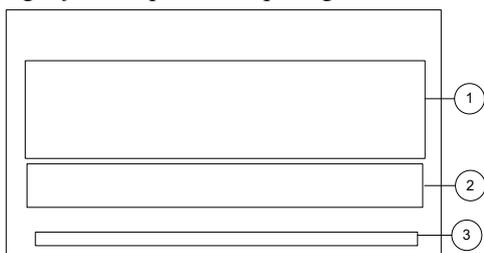
PERANCANGAN

Pengembangan aplikasi pengenalan karakter dengan menggunakan metode *Heteroassociative Memory* ini dilakukan dengan menggunakan bahasa pemrograman *Microsoft Visual Basic .Net 2008* dan *database Microsoft Access*. Aplikasi memiliki 5 (lima) buah form, antara lain:

1. *Form Splash Screen.*
2. *Form Utama.*
3. *Form Pelatihan.*
4. *Form Pengenalan.*
5. *Form Database.*

1. Form Splash Screen

Form ini berfungsi sebagai form pengenalan. Form ini menampilkan logo dan nama aplikasi serta identitas mahasiswa penyusun skripsi (nama dan NIM). Form ini akan tampil pertama sekali saat aplikasi dijalankan. Splash screen akan hilang bila user menekan sembarang tombol pada keyboard atau user menunggu selama 5 detik. Rancangan form dapat dilihat pada gambar 9.



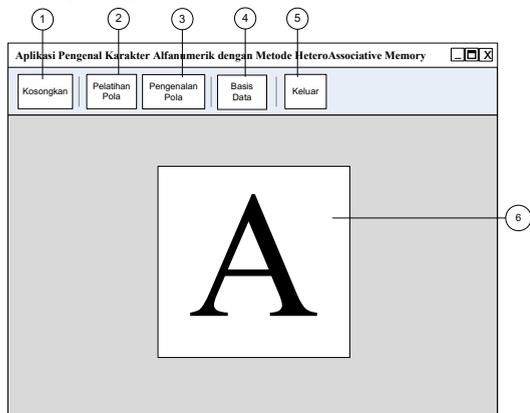
Gambar 9. Rancangan Form Splash Screen

Keterangan:

- 1 : nama aplikasi
- 2 : data penulis / penyusun skripsi
- 3 : jurusan dan nama kampus

2. Form Utama

Pada form ini terdapat area penulisan karakter dan toolbar yang berisi sejumlah tombol yaitu: kosongkan gambar, proses pelatihan, proses pengenalan dan database. Rancangan form dapat dilihat pada gambar 10.



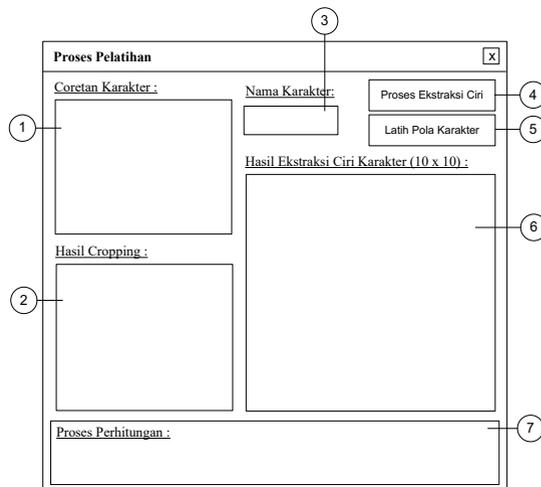
Gambar 10. Rancangan Form Utama

Keterangan:

- 1 : tombol 'Kosongkan', untuk menghapus / mengosongkan area penggambaran pada daerah no.6.
- 2 : tombol 'Pelatihan Pola', untuk melatih pola karakter dengan membuka Form Pelatihan.
- 3 : tombol 'Pengenalan Pola', untuk menampilkan hasil pengenalan pola karakter dengan membuka Form Proses Pengenalan.
- 4 : tombol 'Basis Data', untuk menampilkan pola karakter yang telah dilatih sebelumnya dan sudah tersimpan di dalam database, dengan membuka Form Database.
- 5 : tombol 'Keluar', untuk menutup form.
- 6 : area penggambaran / penulisan karakter / huruf.

3. Form Pelatihan

Form Pelatihan berfungsi untuk melakukan pelatihan pola terhadap input gambar karakter yang ditampilkan pada form Utama. Gambar karakter akan ekstraksi ciri terlebih dahulu, kemudian dilatih dengan menggunakan metode Heteroassociative Memory. Rancangan form dapat dilihat pada gambar 11 berikut.



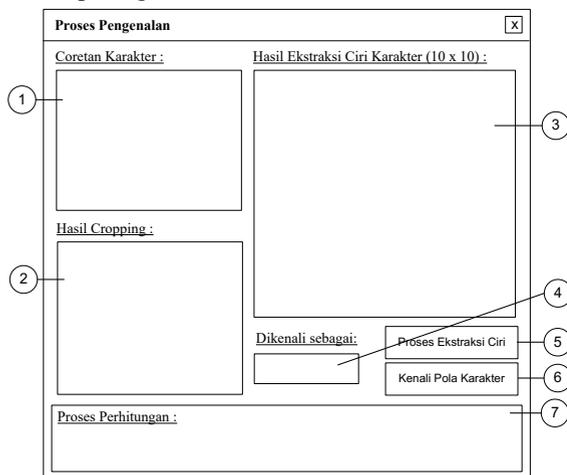
Gambar 11. Rancangan Form Pelatihan

Keterangan:

- 1 : *picturebox*, untuk menampilkan coretan karakter.
- 2 : *picturebox*, untuk menampilkan hasil *cropping*.
- 3 : *textbox*, untuk memasukkan nama dari karakter yang akan dilatih.
- 4 : tombol “Ekstraksi Ciri”, untuk melakukan proses ekstraksi ciri.
- 5 : tombol “Latih Pola”, untuk melakukan proses pelatihan terhadap pola karakter dan menyimpannya ke dalam *database*.
- 6 : *picturebox*, untuk menampilkan pola hasil ekstraksi ciri (10 x 10 kotak).
- 7 : *textbox*, untuk menampilkan proses perhitungan pada proses pelatihan.

4. Form Proses Pengenalan

Form Proses Pengenalan berfungsi untuk melakukan proses pengenalan terhadap sebuah karakter. Sama seperti pada proses pelatihan, gambar karakter akan diekstraksi ciri dan kemudian dikenali dengan menggunakan metode *Heteroassociative Memory*. Rancangan *form* dapat dilihat pada gambar 12 berikut.



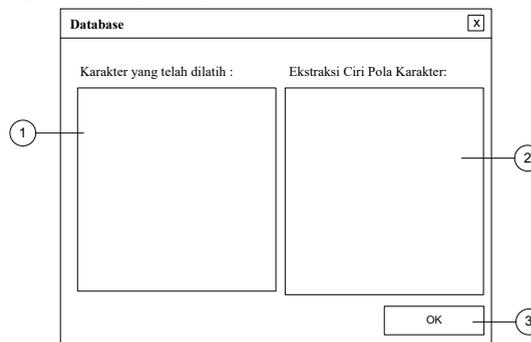
Gambar 12. Rancangan *Form* Proses Pengenalan

Keterangan:

- 1 : *picturebox*, untuk menampilkan coretan karakter.
- 2 : *picturebox*, untuk menampilkan hasil *cropping*.
- 3 : *picturebox*, menampilkan hasil ekstraksi ciri karakter (10 x 10 kotak).
- 4 : *textbox*, untuk menampilkan hasil pengenalan nama karakter.
- 5 : tombol “Ekstraksi Ciri”, untuk melakukan proses ekstraksi ciri.
- 6 : tombol “Kenali Karakter”, untuk melakukan proses pengenalan.
- 7 : *textbox*, untuk menampilkan hasil perhitungan pada proses pengenalan.

5. Form Database

FormDatabase berfungsi untuk menampilkan semua pola karakter yang sudah pernah dilatih dan disimpan di dalam *database*. *Form* ini dibutuhkan *user* untuk melihat pola karakter beserta nama karakter yang telah tersimpan di dalam *database*. Rancangan *form* dapat dilihat pada gambar 13 berikut.



Gambar 13. Rancangan *FormDatabase* Karakter

Keterangan:

- 1 : *list*, untuk memilih nama karakter yang sudah pernah dilatih.
- 2 : pola dari nama karakter yang sedang dipilih pada daerah no.1.
- 3 : tombol “OK”, untuk menutup *form*.

Perancangan Database

Database engine yang digunakan untuk menyimpan pola karakter dan bobot hasil pelatihan adalah *Microsoft Access 2007*. Di dalam file "database.mdb", terdapat 2 (dua) buah tabel, yaitu:

1. Tabel "Pola".

Tabel ini berfungsi untuk menyimpan semua nama karakter serta 100 bit pola karakter hasil ekstraksi ciri. Struktur tabelnya adalah sebagai berikut.

Tabel 1. Struktur Tabel "Pola"

| No. | Nama Field | Tipe | Keterangan |
|-----|------------|------|--|
| 1. | Nama | Text | Nama karakter |
| 2. | Input | Memo | Hasil ekstraksi ciri berupa 100 bit bipolar (<i>primary key</i>) |
| 3. | Output | Text | Output / nomor urut karakter dalam bentuk 8 bit bipolar |

2. Tabel "Bobot".

Tabel ini berfungsi untuk menyimpan bobot hasil pelatihan. Struktur tabelnya adalah sebagai berikut.

Tabel 2. Struktur Tabel "Bobot"

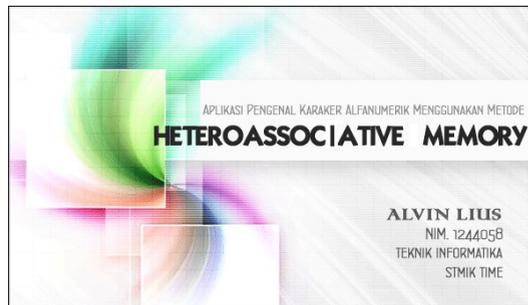
| No. | Nama Field | Tipe | Keterangan |
|-----|------------|-----------------|-----------------------------------|
| 1. | Indeks1 | Number (Long) | Indeks array pertama dari bobot w |
| 2. | Indeks2 | Number (Long) | Indeks array kedua dari bobot w |
| 3. | Nilai | Number (Double) | Nilai bobot |

Di antara kedua tabel, tidak terdapat *relationship* (hubungan), karena tabel "Pola" berfungsi untuk menyimpan semua hasil ekstraksi

ciri karakter yang pernah dilatih, dan tabel "Bobot" berfungsi untuk menyimpan semua bobot w hasil pelatihan dari pola karakter.

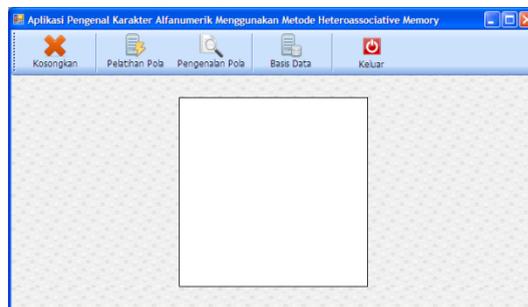
Hasil

Sub bab ini akan membahas cara menjalankan aplikasi pengenalan karakter dengan menggunakan metode *Heteroassociative Memory*. Jalankan aplikasi melalui file "Pengenalan.exe" dan *formsplash screen* akan muncul seperti terlihat pada gambar 14. *Form* ini berisi nama aplikasi dan identitas mahasiswa



Gambar 14. Tampilan *Form Splash Screen*

Tunggu beberapa saat atau tekan tombol *enter* pada *keyboard*, maka *form Splash Screen* akan tertutup dan selanjutnya muncul *form* Utama, seperti terlihat pada gambar 15.



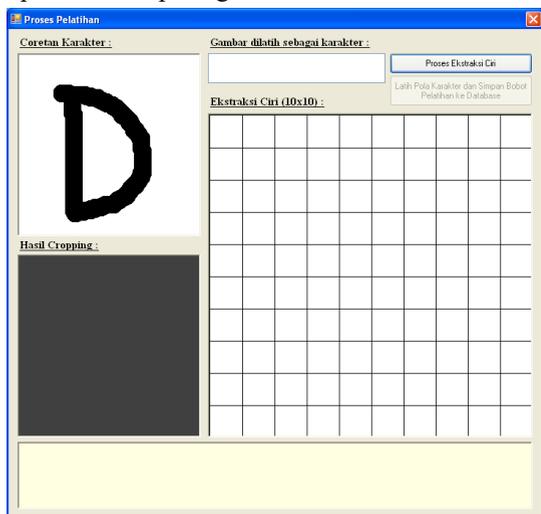
Gambar 15. Tampilan *Form* Utama

Untuk menghapus coretan tulisan, tekan tombol "Kosongkan" pada *toolbar*. Untuk memasukkan huruf, *click-and-drag mouse* pada area penulisan sehingga membentuk huruf, seperti terlihat pada gambar 16.



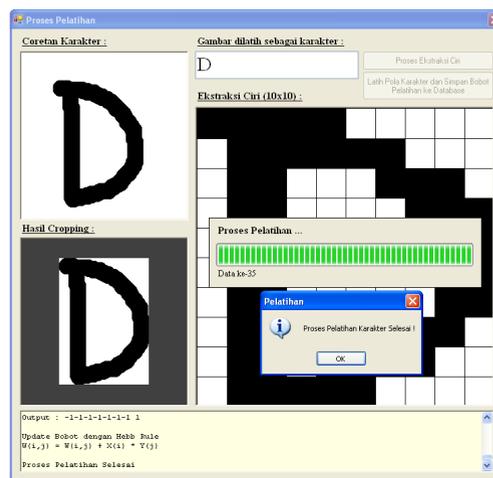
Gambar 16. Coretan Huruf “D” pada Form Utama

Untuk melakukan proses pelatihan pada pola karakter baru, tekan tombol “Pelatihan Pola” pada *toolbar*, dan *form* Pelatihan akan muncul seperti terlihat pada gambar 17.



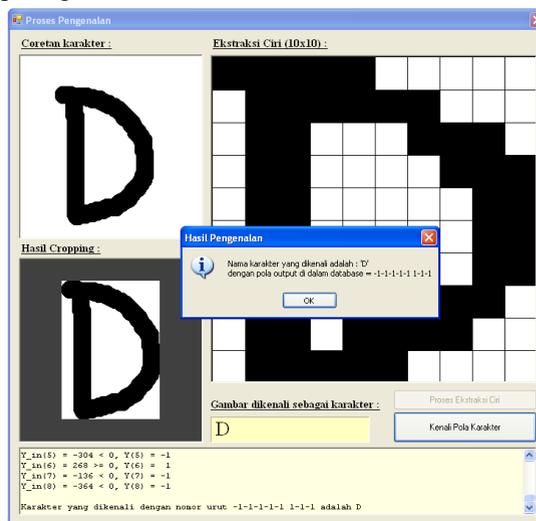
Gambar 17. Tampilan Form Pelatihan

Masukkan nama karakter dari coretan huruf pada *textbox*, tekan tombol “Proses Ekstraksi Ciri” untuk melakukan proses ekstraksi ciri terhadap karakter, dan tekan tombol “Latih Pola Karakter” untuk memulai proses pelatihan. Proses pelatihan dapat dilihat pada gambar 18.



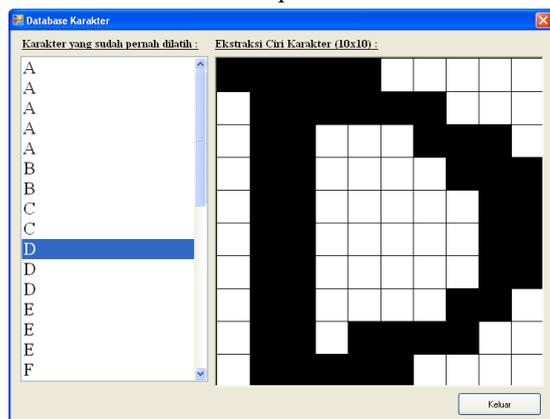
Gambar 18. Proses Pelatihan Selesai

Hasil pelatihan adalah bobot w yang akan disimpan ke dalam database. Untuk melakukan proses pengenalan, tekan tombol “Pengenalan Pola” pada *toolbar* dan *form* Pengenalan akan muncul seperti terlihat pada gambar 19. Tekan tombol “Proses Ekstraksi Ciri” untuk melakukan proses ekstraksi ciri terhadap karakter, dan tekan tombol “Kenali Pola Karakter” untuk memulai proses pengenalan. Hasil pengenalan dapat dilihat pada gambar 19.



Gambar 19. Tampilan Form Pengenalan

Untuk melihat semua pola huruf yang sudah pernah dilatih sebelumnya, tekan tombol “Basis Data” pada *toolbar* dan *form Database* akan muncul seperti terlihat pada gambar 20. Pilih salah satu huruf di *listbox*, dan hasil ekstraksi ciri dari huruf tersebut akan ditampilkan.



Gambar 20. Tampilan *Form Database*

PEMBAHASAN

Dari pengujian sistem secara manual, kelebihan dari aplikasi yang dibangun adalah:

1. Sistem menggunakan jaringan syaraf tiruan *Heteroassociative Memory*, sehingga *output* dari proses pengenalan langsung menghasilkan salah satu nomor urut karakter. Ini lebih efisien daripada harus membandingkan pola karakter dengan satu per satu pola di dalam *database*.
2. Sistem dapat mempelajari kembali pola yang salah dikenali melalui proses pelatihan.

Sedangkan beberapa kelemahan dari aplikasi yang dibangun adalah sebagai berikut:

1. Sistem hanya dapat menerima *input* coretan karakter dari *mouse*.
2. Sistem tidak dapat melakukan pengenalan terhadap beberapa karakter sekaligus.

KESIMPULAN

Setelah menyelesaikan pengembangan aplikasi pengenalan karakter alfanumerik menggunakan metode *Heteroassociative Memory*, beberapa hal yang dapat disimpulkan adalah sebagai berikut:

1. Aplikasi dapat digunakan untuk dapat mengenali karakter alfanumerik yang ditulis melalui *mouse* dengan menggunakan metode *Heteroassociative Memory*.
2. Hasil pengenalan terhadap pola karakter yang lebih baik dengan metode *Heteroassociative Memory* harus melakukan proses pelatihan yang sesering mungkin.
3. Untuk meningkatkan akurasi pengenalan, proses pelatihan dapat dilakukan beberapa kali untuk satu karakter. Bila hasil pengenalan terhadap suatu pola karakter salah, maka pola tersebut perlu dilatih dan dipelajari lagi, sehingga pola dapat dikenali dengan benar.

DAFTAR PUSTAKA

- Basuki, dkk. (2005). *Pengolahan Citra Digital Menggunakan Visual Basic*, Yogyakarta: Penerbit Graha Ilmu.
- Hartanto, S., Sugiharto, A., & Endah, S. N. (2014). Optical Character Recognition Menggunakan Algoritma Template Matching Correlation. *Jurnal Masyarakat Informatika*. 5 (9). 1-12.
- Kusumadewi, S. (2003). *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Penerbit Graha Ilmu.
- Putra, D. (2010). *Pengolahan Citra Digital*, Yogyakarta: Penerbit Andi.
- Puspitaningrum, D. (2006). *Pengantar Jaringan Syaraf Tiruan*, Yogyakarta: Penerbit Andi.

Siang, J. (2009). *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*, Yogyakarta: Penerbit Andi.

Simarmata, J. (2005). *Teknologi Komputer dan Informasi*. Yogyakarta: Penerbit Andi.

Soetam, R. W. (2015). *Pemrograman Visual Aplikasi Database dengan Visual Basic .Net*, Jakarta: Penerbit Thousand Stars.

Sutojo, T., dkk. (2011). *Kecerdasan Buatan*, Yogyakarta: Penerbit Andi.

Sutoyo, dkk. (2009). *Teori Pengolahan Citra Digital*, Yogyakarta: Penerbit Andi.