

Character Recognition Dengan Menggunakan Jaringan Syaraf Tiruan

Hendri, S.Kom. M.Kom.
STMIK TIME, Jl. Merbabu No 32 AA-BB Medan
Email : h4ndr7@hotmail.com

Abstrak

Jaringan saraf tiruan telah banyak dimanfaatkan untuk pengenalan pola citra, karakter dan suara yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia, salah satu nya yaitu pada proses *character recognition*. Cara kerjanya yaitu gambar input akan diubah ke bentuk matriks berukuran 10 x 10 dengan menggunakan algoritma reduksi lalu disimpan ke dalam suatu *file knowledge base*. Pola yang disimpan tadi akan digunakan menjadi pola yang akan diacu pada saat terjadi pengujian pola terhadap komputer. Pengujian ini dilakukan dengan membandingkan pola yang diuji dengan pola yang sudah tersimpan di komputer dengan menggunakan jaringan saraf tiruan.

Kata Kunci : jaringan saraf tiruan, citra digital, *character recognition*

1. PENDAHULUAN

Makin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran kecerdasan buatan. Seiring dengan perkembangan teknologi, muncul beberapa teknologi yang bertujuan untuk membuat agar komputer menjadi cerdas sehingga dapat menirukan kerja manusia sehari-hari. Keputusan yang dibuat komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan buatan tidak hanya dominan di bidang ilmu komputer (informatika), namun juga sudah merambah di berbagai disiplin ilmu yang melahirkan berbagai ilmu seperti : pengolahan citra, teori kendali, robotika dan salah satunya adalah pengenalan pola dengan jaringan saraf tiruan.

Kecerdasan buatan tidak hanya dominan di bidang ilmu komputer (informatika), namun juga sudah merambah di berbagai disiplin ilmu yang melahirkan berbagai ilmu seperti: pengolahan citra, teori kendali, robotika dan salah satunya adalah pengenalan pola dengan jaringan saraf tiruan. Jaringan saraf tiruan telah banyak dimanfaatkan untuk pengenalan pola citra, karakter dan suara yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia. Jaringan ini sering pula digunakan untuk ekstraksi ciri (*feature*) pada proses awal pengenalan pola yang mampu mereduksi dimensi input pola ke jumlah yang lebih sedikit sehingga pemrosesan komputer menjadi lebih hemat. Jaringan saraf tiruan mempunyai kelebihan untuk mengenali pola karakter yang diperkecil sekaligus pergeseran bentuk. Jaringan akan dilatih terus menerus sampai diperoleh *error*

minimum. Penerapan jaringan saraf tiruan untuk mengenali karakter ini dapat digunakan untuk membantu pengenalan dan penulisan karakter bagi anak-anak yang belajar menulis.

2. LANDASAN TEORI Jaringan Saraf Tiruan

Jaringan saraf tiruan adalah merupakan salah satu representasi ‘buatan’ dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Sedangkan saraf itu sendiri merupakan sel-sel yang memproses informasi pada otak dan membentuk sebuah jaringan sebagai suatu kesatuan yang menghubungkan saraf tersebut. Istilah ‘buatan’ yang digunakan pada jaringan saraf tiruan diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. Jaringan saraf tiruan memodelkan suatu struktur yang mirip dengan otak manusia.

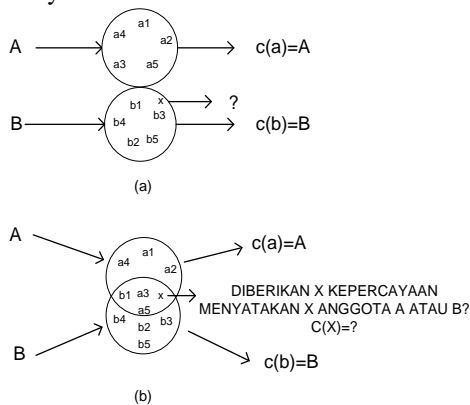
Pengenalan Dan Pencocokan Pola

Fokus tugas pengenalan pola berubah-ubah tergantung keadaan. Dalam beberapa kasus, kemiripan atau keserupaan (*similarity*) menjadi perhatian utama. Untuk kasus yang lain, fokusnya adalah memperkirakan kepercayaan (*beliefs*). Perbedaan yang dilakukan sangatlah penting sekalipun pada umumnya baik keserupaan maupun kepercayaan perlu diperkirakan.

Bila kita yakin bahwa kita belum pernah melihat suatu pola tertentu sebelumnya, tentu masuk akal bila pola tersebut diklasifikasikan berdasarkan keserupaan. Artinya, kita mengklasifikasikan pola baru itu sebagai anggota kelas obyek A jika pola itu sangat serupa dengan pola-pola yang merupakan anggota kelas A. Pemfokusan ini telah memadai jika tidak ada tumpangtuh (*overlap*) dalam ruang pola. Artinya, jika setiap penggambaran pola terpetakan tepat satu dan hanya satu indeks keanggotaan kelas.

Keadaan semacam ini ditunjukkan dalam Gambar 2.1(a). Obyek-obyek A dan B masing-masing ditunjukkan oleh banyak pola yang berbeda, namun pola-pola tersebut tidak tumpangtuh dalam ruang *feature*. Pola-pola itu dipetakan dengan cara dari banyak ke satu tujuan keanggotaan. Untuk keadaan semacam ini, prosedur deterministik berbasis komputer untuk memperkirakan keserupaan merupakan hal yang sangat penting. Prosedur semacam ini membuat kita dapat melihat perbedaan antar kelas dan melakukan generalisasi untuk setiap kelas. Aturan klasifikasi yang umum adalah : jika

pola x paling serupa dengan pola kelas A, maka pola x tersebut merupakan anggota kelas A. Sebaliknya, gambar 2.1(b) memperlihatkan suatu situasi dimana suatu pola dapat menjadi anggota dari kelas yang berbeda. Artinya, pola tersebut (pola x) dapat merepresentasikan baik obyek A maupun obyek B. Dalam hal ini, fokus perhatian kita lebih terarah pada memperkirakan kepercayaan daripada mengukur keserupaan, yakni berapakah penimbang relatif “tingkat kepercayaan” bahwa x merepresentasikan obyek A atau obyek B, dan seterusnya.

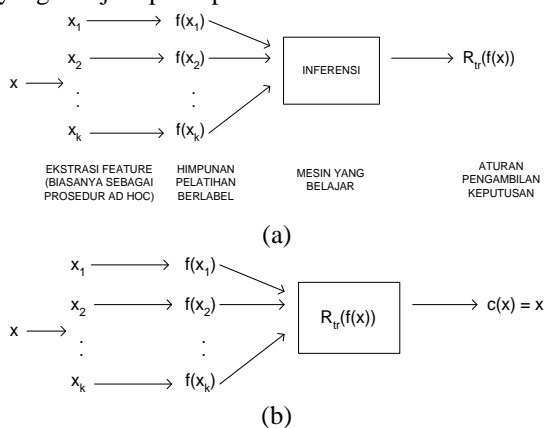


Gambar 2.1 Mengukur Keserupaan dan Kepercayaan

(a) *Generalisasi pengukuran keserupaan : Apakah x lebih serupa dengan a atau b?*

(b) *Pengukuran kepercayaan bahwa suatu pola merupakan anggota suatu kelas tertentu; di sini ditampilkan pola yang dapat merepresentasikan dua kelas.*

Dalam kenyataan, pengenalan pola yang sebenarnya berada di antara dua ekstrim di atas. Sekalipun pola-pola itu terpisah dalam suatu ruang *feature* tertentu, namun masih terdapat problema tentang apa yang terjadi pada batas-batas daerah antar kelas di mana suatu tumpang-suh dapat terjadi. Secara prinsip, dan dalam praktek, pengenalan pola tidak hanya menangani klasifikasi, namun secara lebih umum juga mencakup pengukuran atribut. Kenyataannya, keanggotaan kelas dapat dipandang hanya sebagai salah satu dari begitu banyak atribut yang menjadi pusat perhatian.



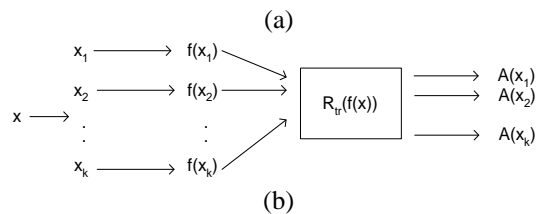
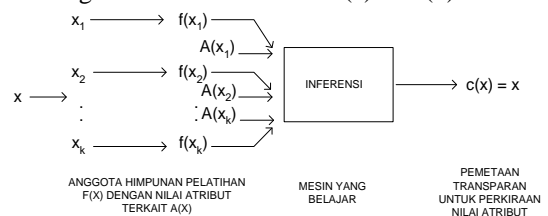
Gambar 2.2 Model Klasifikasi Pengenalan Pola Berbasis Komputer

(a) *Proses mempelajari aturan klasifikasi dengan menggunakan serangkaian pelatihan berlabel.*

(b) *Proses penggunaan pemetaan yang telah belajar untuk melakukan klasifikasi.*

Dalam model klasifikasi, kita akan menanganinya :

1. Melakukan ekstraksi *feature*; artinya memutuskan bagaimana manifestasi x dari obyek X harus digambarkan secara simbolik dalam bentuk f(x).
2. Mempelajari pemetaan transparan $R_{tr}()$; artinya menggunakan sekumpulan pola rangkaian pelatihan yang telah diberi label untuk membuat aturan-aturan keputusan.
3. Melatih pemetaan $R_{tr}(f(x))$ untuk melakukan pengerjaan klasifikasi yang sebenarnya.
4. Hal ini ditunjukkan secara amat terbatas dan singkat dalam Gambar 2.2 (a) dan (b).

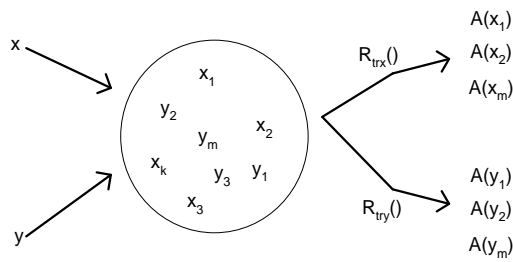


Gambar 2.3 Ilustrasi Skematik Model Pemeriksa Pengenalan Pola yang Lebih Umum

(a) *Menggunakan serangkaian pelatihan berlabel untuk mempelajari pemetaan transparan dalam memperkirakan nilai-nilai atribut.*

(b) *Menggunakan pemetaan transparan yang telah mengalami proses belajar untuk memperkirakan nilai-nilai atribut.*

Selanjutnya, diperlihatkan Gambar 2.3 (a) dan (b) yang memberikan penggambaran yang lebih baik. Dalam hal ini, penekanan diberikan pada atribut-atribut pola. Manifestasi x dari obyek X digambarkan oleh f(x). Pemetaan $R_{tr}(f(x))$ membawa setiap pola f(x) ke A(x). Pemetaan ini menggambarkan model pemeriksa pengenalan pola (tidak seperti model klasifikasi tradisional). Pemetaan ini akan mengalami proses belajar dengan cara dilatih dengan sejumlah sampel pola tertentu yang masing-masing diberi label (nama) yang bersesuaian dengan nilai atribut tertentu atau dengan beberapa nilai atribut bila terdapat lebih dari satu atribut. Pemetaan ini tidak hanya dapat menghasilkan kembali semua atribut pola rangkaian pelatihan, namun juga harus dapat menampung setiap pola baru sehingga nilai-nilai atribut yang benar dapat diperkirakan dari pemetaan $R_{tr}()$.



Gambar 2.4 Ilustrasi Skematik Pemerkiraan Nilai Atribut bila Terdapat Ketidakpastian

Gambar 2.4 memperlihatkan keadaan yang lebih umum. Gambar tersebut mengilustrasikan apa yang dipertaruhkan dalam memperkirakan atribut bila terdapat lebih dari satu kelas, dan bila ada kelas-kelas yang bertumpangtuh (*overlap*) dalam ruang *feature*. Dalam gambar ini, x_1, x_2, \dots, x_k , merupakan manifestasi yang berbeda dari obyek X , dan $f(x_1), f(x_2), \dots, f(x_k)$ merupakan representasi pola simbolik yang bersesuaian dengan manifestasi-manifestasi tersebut. Pemetaan transparan $R_{trx}()$ memetakan penggambaran ini ke ruang atribut, membawa $f(x_1)$ ke $A(x_1)$, $f(x_2)$ ke $A(x_2)$ dan seterusnya.

Sedangkan y_1, y_2, \dots, y_m merupakan manifestasi dari obyek Y , dan $f(y_1), f(y_2), \dots, f(y_m)$ adalah representasi pola dari manifestasi ini. Pemetaan transparan lain $R_{try}()$ memetakan $f(y_1)$ ke $A(y_1)$, $f(y_2)$ ke $A(y_2)$ dan seterusnya. Atribut yang menjadi perhatian adalah sama untuk setiap kasus; artinya nama atribut tersebut sama, dan nilai-nilai pemikirannya adalah $A(x_1), \dots, A(x_k)$, dan $A(R_{trx}() y_1), \dots, A(y_m)$. Untuk sebuah pola baru tertentu, tugas yang harus dilakukan adalah menentukan pemetaan mana, R_{trx} atau R_{try} , yang harus digunakan untuk melakukan pemerkiraan nilai atribut A . Dalam proses klasifikasi, biasanya keluaran (*ouput*) yang terjadi adalah sekumpulan indeks kelas diskrit, sedangkan dalam proses pemerkiraan keluarannya biasanya merupakan variabel yang nilainya kontinu

Simple Character Recognition

Tipe kelas berupa karakter huruf besar dan huruf kecil, angka 10 digit (0-9) dan simbol-simbol khusus seperti tanda titik, tanda seru, tanda kurung, lambang *dollar*, *pound* dan lain-lain. Algoritma pengenalan pola digunakan untuk mengekstraksi ciri dan menentukan karakter yang diamati ke kelas yang tepat. Jaringan saraf tiruan seperti *Self Organizing Maps* dimunculkan sebagai metode yang cepat untuk melakukan penggolongan dalam OCR. Algoritma ini menggunakan notasi sigma dan matriks dalam prosedur kerjanya.

3. ANALISA DAN HASIL

Perangkat lunak pengenalan karakter yang dibuat adalah dengan menggunakan metode *Simple Character Recognition*, yang pada dasarnya adalah dengan melakukan proses pencocokan pola setelah dilakukan reduksi terhadap karakter tulisan tangan tersebut.

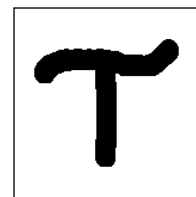
Untuk menuliskan sebuah karakter pada layar *picture box* adalah dengan menggunakan bantuan *mouse*. Di mana pada *mouse* digunakan 3 buah *event*, yakni :

1. *Mouse_down* : diterima dan dijalankan apabila tombol *mouse* ditekan.
2. *Mouse_move* : diterima dan dijalankan apabila *mouse* digerakkan.
3. *Mouse_up* : diterima dan dijalankan apabila tombol *mouse* dilepaskan.

Penggunaan ketiga *event* di atas dalam menuliskan sebuah karakter, bisa dijelaskan sebagai berikut:

1. Apabila *user* ingin menggambar sebuah karakter maka tombol *mouse* ditekan dan event *mouse_down* dijalankan dan mode gambar diaktifkan.
2. Apabila *mouse* digerakkan pemakai dan mode gambar aktif, maka pola karakter yang diinginkan digambarkan.
3. Apabila tombol *mouse* dilepas, maka pola karakter yang digambar diberi izin untuk diletakkan dalam memori gambar dan mode gambar di-nonaktifkan.

Misalnya dilakukan penulisan sebuah karakter ‘T’ kepada perangkat lunak dengan memanfaatkan *event* dari *mouse*, dimana karakter ‘T’ ini dapat kita lihat pada gambar di bawah ini,



Gambar 3.1 Karakter ‘T’

Dengan nilai panjang x lebar dari gambar sebuah karakter ‘T’ di atas adalah 2535 x 2535 pixel. Dengan berdasarkan kepada algoritma *Simple Character Recognition* langkah – langkah yang dilakukan adalah sebagai berikut,

1. Mencari sisi tepi dari karakter.
2. Melakukan reduksi menjadi matriks 10 x 10,
3. Melakukan pencocokan pola dengan *knowledge base*.

1. Proses Mencari Sisi Tepi

Pada tahap ini bertujuan untuk mengecilkan daerah kerja. Dengan melakukan proses pencarian sisi tepi, akan didapatkan hasil seperti yang terlihat pada gambar 3.2. Di mana tampak sebuah kotak mengelilingi karakter ‘T’ tersebut.

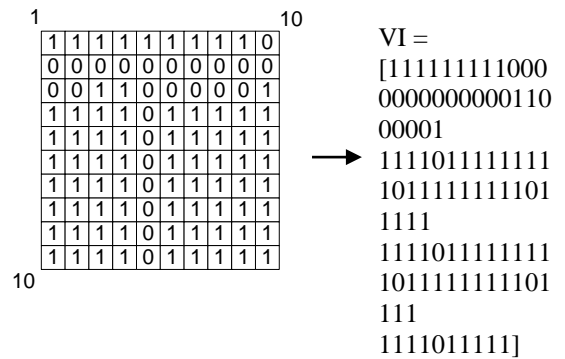
maka nilainya adalah 1 (satu). Hasil akhir dari proses reduksi ini adalah pola dari karakter yang digambarkan yang berupa matriks berdimensi 10 x 10.

Proses Pencocokan Pola

Pada tahap ini, setelah mendapatkan pola berupa matriks berdimensi 10 x 10, maka pola ini akan dicocokkan dengan pola yang ada pada *knowledge base*. Di mana persentase perhitungan serta pencocokan menggunakan algoritma *recognition*.

Tahapan – tahapan yang dilakukan dalam proses pencocokan pola adalah sebagai berikut,

- a. Ubah pola matriks berdimensi 10 x 10 menjadi vektor *input* berelemen 100 (VI).



Gambar 3.5 Ubah Vektor

- b. Setelah didapatkan vektor *input*, maka dilakukan langkah – langkah sebagai berikut,

- 1. Ambil sebuah nama karakter dan polanya dari *knowledge base*.

Misalkan, pada *knowledge base* terdapat 2 karakter hasil pembelajaran :

- I, dengan pola berupa vektor =
[11111111111000011000000000000001
1101111111101111]

11110111111111011000000000000110
00011001111111111]

- T, dengan pola berupa vektor =
[111111110000000000000000000001111
10111111111011111]

111101111111110111111110111111111
01111111110111111]

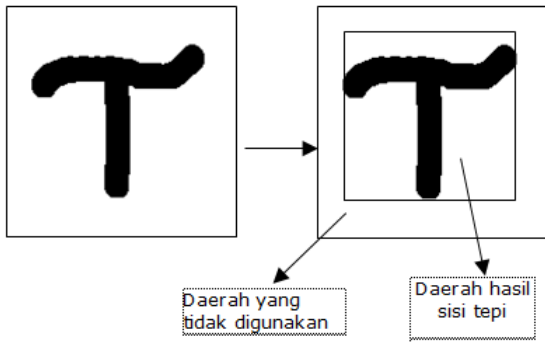
Ambil pola 'I' yang merupakan pola awal.
2. Bandingkan pola vektor *input* dengan pola vektor *knowledge base*. Jika sama maka pola kemiripan bertambah satu, demikian sebaliknya, jika berbeda maka pola kemiripan berkurang satu. Di mana pada keadaan awal, pola kemiripan adalah 0 (nol).

Misalnya:

V *input* (VI) = 0011 dan Vektor *knowledge base* (VK) = 0111

Pola kemiripan (PK) awal = 0, jumlah vektor = 4

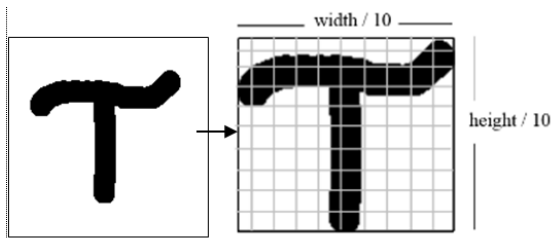
Vektor 1 : VI(1) = 0, VK(1) = 0 → PK = 0 + 1 = 1



Gambar 3.2 Karakter 'T' Setelah Didapatkan Sisi Tepinya

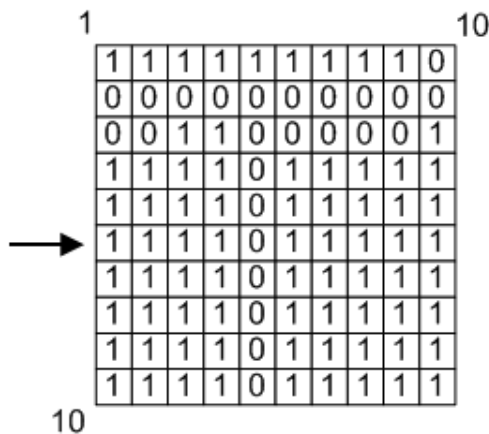
2. Proses Reduksi

Pada tahap ini bertujuan untuk memasukkan karakter tersebut menjadi sebuah matriks berdimensi 10 x 10. Dengan menggunakan algoritma *reduction*, tahapan – tahapan yang dilakukan adalah sebagai berikut,



Gambar 3.3 Pembagian Batas Sisi Tepi

Pertama daerah hasil pencarian sisi tepi dilakukan pembagian dengan 10. Masing – masing lebar dan tinggi dari daerah tersebut dibagi dengan 10, seperti yang terlihat pada gambar 3.3.



Gambar 3.4 Hasil Reduksi

Setelah dilakukan pembagian dengan 10, dilakukan pemeriksaan terhadap daerah – daerah titik temu berdasarkan sumbu x dan sumbu y antara garis – garis hasil pembagian tersebut seperti yang terlihat pada gambar 4.4. Jika daerah titik temu berwarna hitam maka pada nilai yang dimasukkan ke dalam matriks adalah bernilai 0 (nol). Demikian sebaliknya, jika daerah titik temu berwarna putih

Vektor 2 : $VI(2) = 0, VK(2) = 1 \rightarrow PK = 1 - 1 = 0$
 Vektor 3 : $VI(3) = 1, VK(3) = 1 \rightarrow PK = 0 + 1 = 1$
 Vektor 4 : $VI(4) = 1, VK(4) = 1 \rightarrow PK = 1 + 1 = 2$

Maka pola kemiripan adalah sebanyak 2. Untuk contoh di atas, setelah dilakukan perhitungan pola kemiripan dengan *knowledge base* hasilnya adalah sebagai berikut :

- Dengan karakter ‘I’, pola kemiripan adalah sebanyak 48.
- Dengan karakter ‘T’, pola kemiripan adalah sebanyak 94.

3. Ambil nama karakter dari *knowledge base* yang pola kemiripannya paling besar.

Untuk contoh di atas, karakter ‘T’ mempunyai pola kemiripan terbesar.

4. Hitung persentase:

$$\text{Persentase kemiripan} = \frac{\text{pola kemiripan}}{\text{jumlah pola yang}} \times 100\%$$

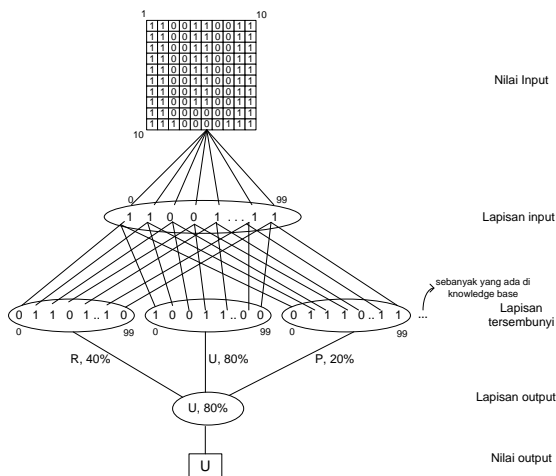
dibandingkan

$$\text{Persentase kemiripan} = \frac{94}{100} \times 100\% = 94\%$$

5. Jika persentase kemiripan kurang dari 68 % maka karakter tersebut perlu dilakukan pembelajaran.

Layer

Jumlah layer yang digunakan adalah 3 lapisan yakni lapisan *input*, lapisan tersembunyi dan lapisan *output*. Dimana lapisan pertama adalah lapisan untuk proses reduksi yang hasilnya berupa pola matriks 10 x 10, lapisan kedua adalah lapisan tersembunyi untuk proses pencocokan pola dengan yang ada di *knowledge base* dan yang ketiga adalah lapisan output yang berupa nama karakter hasil pembelajaran dan persentase kemiripan dengan pola yang ada di *knowledge base*.



Gambar 3.6 Layer pada Metode Simple Character Recognition

4. KESIMPULAN

Adapun kesimpulannya sebagai berikut:

1. Setiap gambar yang dipelajari oleh perangkat lunak akan diubah ke bentuk kerangka (*skeleton*) terlebih dahulu dengan menggunakan algoritma reduksi dan disimpan ke dalam *knowledge base*. Setelah itu, baru dicocokkan dengan menggunakan jaringan saraf tiruan dimana dilakukan perbandingan antara gambar input dengan kerangka (*skeleton*) yang tersimpan dalam *knowledge base*.
2. Persentase kemiripan dapat diperoleh dengan membandingkan pola yang telah disimpan setelah dipelajari (*learning*) dengan pola *testing* melalui perhitungan vektor *input* pola kemiripan dibandingkan dengan pola test yang diperoleh dibagi dengan jumlah pola yang dibandingkan.

5. DAFTAR PUSTAKA

Desiani, A., Arhami, M., **Konsep Kecerdasan Buatan**, Penerbit Graha Ilmu, 2009.

Pressman, R.S., **Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku Satu)**, Mc Graw-Hill Companies, Inc, Penerbit ANDI, 2012.

Setiawan Sandi, **Artificial Inteligencia**, Andi Offset Yogyakarta, 2008.

T. Sutoyo, Edy Mulanto, Vincent Suhartono, Oky Dwi Nurhayati, Wijanarto, **Teori Pengolahan Citra Digital**, Penerbit Andi, Yogyakarta, 2009.