

ISSN 2087-0256 , E-ISSN 2580-6939

# smatika Jurnal

STIKI Informatika Jurnal

Volume 08, Nomor 01, April Tahun 2018



**Prototype Pengereng Biji Jagung Berbasis  
Mikrokontroler**  
Syahminan

**Analisis Hama pada Tanaman Anggur dengan  
Pendekatan Metode CF (*Certainty Factor*) Berbasis  
*Mobile Android***  
Permata Ika Hidayati

**Restorasi Citra *Optical Character Recognition* dengan  
Algoritma Recurrent Hopfield**  
Kukuh Yudhistiro

**Rancang Bangun Sistem Cerdas Suara untuk  
Pengendalian Keamanan Kendaraan Bermotor Roda  
Dua**  
Mochamad Subianto, Oesman Hendra Kelana, Hendra Setia Ligawan

**Rancang Bangun Prototipe Sistem Kontrol Penggunaan  
Air Prabayar Berbasis Arduino Uno**  
Mochamad Subianto, Hendry Setiawan, Kielvien Lourensius Eka S. P.

**Penerapan Metode Simpleks untuk Optimasi Jumlah  
Kalori Makanan yang Dibutuhkan dalam Menu Diet**  
Setya Ardhi, Hari Sutiksno

# **PENGANTAR REDAKSI**

STIKI Informatika Jurnal (SMATIKA Jurnal) merupakan jurnal yang diterbitkan oleh Lembaga Penelitian & Pengabdian kepada Masyarakat (LPPM), Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) Malang.

Pada edisi ini, SMATIKA Jurnal menyajikan 6 (*enam*) naskah dalam bidang sistem informasi, jaringan, pemrograman web, perangkat bergerak dan sebagainya. Redaksi mengucapkan terima kasih dan selamat kepada Pemakalah yang diterima dan diterbitkan dalam edisi ini, karena telah memberikan kontribusi penting pada pengembangan ilmu dan teknologi.

Pada kesempatan ini, redaksi kembali mengundang dan memberi kesempatan kepada para Peneliti di bidang Teknologi Informasi untuk mempublikasikan hasil-hasil penelitiannya melalui jurnal ini. Bagi para pembaca yang berminat, Redaksi memberi kesempatan untuk berlangganan.

Akhirnya Redaksi berharap semoga artikel-artikel dalam jurnal ini bermanfaat bagi para pembaca khususnya dan bagi perkembangan ilmu dan teknologi di bidang Teknologi Informasi pada umumnya.

**REDAKSI**

ISSN 2087-0256, E-ISSN 2580-6939

# smatika Jurnal

STIKI Informatika Jurnal

Volume 08 Nomor 01, April Tahun 2018

---

**Pelindung**

Yayasan Perguruan Tinggi Teknik Nusantara

**Penasehat**

Ketua STIKI

**Pembina**

Pembantu Ketua Bidang Akademik STIKI

**Mitra Bestari**

Prof. Dr. Ir. Kuswara Setiawan, MT (UPH Surabaya)  
Dr. Ing. Setyawan P. Sakti, M.Eng (Universitas Brawijaya)

**Ketua Redaksi**

Subari, S.Kom, M.Kom

**Section Editor**

Jozua F. Palandi, S.Kom, M.Kom  
Nira Radita, S.Pd., M.Pd  
Siti Aminah S.Si., M.Pd

**Layout Editor**

Saiful Yahya, S.Sn, MT.

**Tata Usaha/Administrasi**

Muh. Bima Indra Kusuma

**SEKRETARIAT**

**Lembaga Penelitian & Pengabdian kepada Masyarakat  
Sekolah Tinggi Informatika & Komputer Indonesia (STIKI)  
Malang**

SMATIKA Jurnal

Jl. Raya Tidar 100 Malang 65146

Tel. +62-341 560823

Fax. +62-341 562525

Website: [jurnal.stiki.ac.id](http://jurnal.stiki.ac.id)

E-mail: [jurnal@stiki.ac.id](mailto:jurnal@stiki.ac.id), [lppm@stiki.ac.id](mailto:lppm@stiki.ac.id)

## DAFTAR ISI

---

<b>Prototype Pengering Biji Jagung Berbasis Mikrokontroler .....</b>	<b>01 - 08</b>
Syahminan	
<b>Analisis Hama pada Tanaman Anggur dengan Pendekatan Metode CF (<i>Certainty Factor</i>) Berbasis <i>Mobile Android</i> .....</b>	<b>09 - 17</b>
Permata Ika Hidayati	
<b>Restorasi Citra <i>Optical Character Recognition</i> dengan Algoritma Recurrent Hopfield .....</b>	<b>18 - 22</b>
Kukuh Yudhistiro	
<b>Rancang Bangun Sistem Cerdas Suara untuk Pengendalian Keamanan Kendaraan Bermotor Roda Dua .....</b>	<b>23 - 30</b>
Mochamad Subianto, Oesman Hendra Kelana, Hendra Setia Ligawan	
<b>Rancang Bangun Prototipe Sistem Kontrol Penggunaan Air Prabayar Berbasis Arduino Uno .....</b>	<b>31 - 39</b>
Mochamad Subianto, Hendry Setiawan, Kielvien Lourensus Eka S. P.	
<b>Penerapan Metode Simpleks untuk Optimasi Jumlah Kalori Makanan yang Dibutuhkan dalam Menu Diet .....</b>	<b>40 - 49</b>
Setya Ardhi, Hari Sutiksno	

**Undangan Makalah**

**SMATIKA Jurnal Volume 08 Nomor 02, Oktober Tahun 2018**

# Restorasi Citra *Optical Character Recognition* Dengan Algoritma Recurrent Hopfield

Kukuh Yudhistiro  
Universitas Merdeka Malang  
kukuh.yudhistiro@unmer.ac.id

## ABSTRAK

Pada proses *optical character recognition* seringkali terjadi image yang corrupt sehingga hasil pengenalan karakter tidak sesuai dengan yang diinginkan atau bahkan tidak menghasilkan keluaran. Paper ini akan membahas proses restorasi citra menggunakan algoritma Hopfield menggunakan pemrograman Visual Basic 2010. Unit inputan yang akan diuji adalah pola karakter "+", "x", dan "o". Semua pola masukan tersebut disimpan dalam bentuk vektor berdimensi 3x3 dengan nilai 1 atau -1. Tujuan akhir dari proses restorasi adalah memperbaiki citra yang diberikan. Maka perlu dataset vektor untuk ujicoba dimana vektor pola untuk ujicoba tersebut merupakan pola karakter yang mirip dengan pola karakter "+", "x", dan "o". Sedangkan hasil restorasi yang diharapkan adalah program mampu memperbaiki vektor pola ujicoba dan menampilkan pola mana yang mirip dengannya.

**Kata kunci:** neural network, hopfield, image restoration, restorasi citra, jaringan syaraf tiruan

## ABSTRACT

On OCR process, a corrupt image is often founded. So that the results of character recognition does not match the desired or even not produce output. This paper will discuss the process of image restoration using Hopfield algorithm using Visual Basic 2010 programming. The input unit to be tested is character pattern "+", "x", and "o". All input patterns are stored in 3x3-dimensional vectors with values of 1 or -1. The ultimate goal of the restoration process is to improve the given image. Then a vector dataset is needed for the test where the pattern vector for the test is a character pattern similar to the "+", "x", and "o" character patterns. While the expected restoration result is a program capable of fixing vector test pattern and showing which pattern is similar to it.

**Keywords:** neural network, hopfield, image restoration, image restoration, artificial neural network

## 1. PENDAHULUAN

Deteksi pola dengan cara melakukan pengelompokan data dapat dilakukan menggunakan dengan menggunakan data numerik maupun simbol dan citra. Tujuan pengelompokan data tersebut adalah untuk mengenali suatu objek yang terkandung pada sebuah citra. Kemampuan visual manusia dalam hal mengenali objek yang dilihatnya inilah yang dicoba ditiru oleh mesin yaitu menerima masukan berupa citra objek yang akan diidentifikasi, melakukan praproses pada citra tersebut, dan memberikan informasi tentang objek dalam citra.

Salah satu identifikasi obyek dapat dilakukan melalui proses *optical character recognition (OCR)*. Bila data tersebut mengandung teks maka perangkat lunak tersebut akan melakukan konversi dari *image* ke dalam bentuk teks sehingga data tersebut dapat diolah kembali melalui perangkat lunak lain *word processing*.

Obyek *noise* sering ditemukan pada hasil OCR. *Noise* tersebut disebabkan oleh beberapa faktor seperti kualitas dokumen yang jelek, kaca alat OCR kotor, terdapat bekas cairan berwarna yang mengotori atau dokumen berada dalam posisi miring. *Noise* pada dokumen dapat mengurangi

keakuratan dalam proses pengenalan karakter. Oleh karena itu diperlukan suatu praproses untuk menghilangkan *noise* yang ada.

## 2. METODOLOGI PENELITIAN

John Hopfield memperkenalkan arsitektur jaringan pada paper yang dipublikasikan tahun 1984, yang kemudian dikenal sebagai jaringan Hopfield. John Hopfield menunjukkan suatu memori asosiasi yang dapat menyelesaikan masalah optimasi dengan menggunakan jaringannya. Unit-unit pengolah dalam *network* Hopfield terhubung penuh. Hubungan antar jaringan tersebut bersifat langsung dan setiap pasang unit pengolah memiliki hubungan dalam dua arah. Setiap hubungan memiliki bobot. Bobot ini adalah suatu nilai skalar berdasarkan pada kekuatan hubungan (*connection strength*).

Misal  $T_{ij}$  adalah bobot dari unit  $i$  ke unit  $j$ . Dalam jaringan Hopfield, bobot  $T_{ij}$  dan  $T_{ji}$  mempunyai nilai yang sama sehingga  $T_{ij} = T_{ji}$ . Dalam jaringan Hopfield prosedur pembelajarannya adalah pembentukan bobot koneksi  $T_{ij}$ . Tiap kondisi dari jaringan Hopfield mempunyai sekumpulan energi. Nilai ini didefinisikan dengan:

$$E = -\frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L T_{ij} u_i u_j - \sum_{i=1}^L I_i u_i \tag{1}$$

dimana

$T_{ij}$  = bobot koneksi antara neuron I dan neuron j

$I_i$  = bias

Jaringan Hopfield biner mempunyai suatu lapisan pengolah. Setiap unit pengolah mempunyai sebuah nilai aktifitas atau kondisi (state) yang bersifat biner. Jaringan Hopfield bisa bekerja dengan kondisi 0 dan 1. Jaringan juga dapat bekerja jika digunakan nilai +1 dan -1, hanya saja diperlukan sedikit perubahan dalam persamaannya. Unit-unit pengolah dalam jaringan Hopfield adalah terhubung penuh (*fully connected*) yaitu setiap unit terhubung ke tiap-tiap unit yang lain. Hubungan-hubungan tersebut adalah hubungan langsung dan setiap pasang unit pengolah mempunyai hubungan dalam dua arah. Topologi hubungan ini mempunyai jaringan bersifat *recursive* karena keluaran dari setiap unit memberi masukan ke unit yang lain.

Jaringan ini memiliki bobot-bobot yang simetris. Pada jaringan Hopfield setiap unit tidak memiliki hubungan dengan dirinya sendiri, secara matematik hal ini memenuhi  $W_{ji} = W_{ij}$  ( di mana i = baris dan j = kolom) untuk  $i \neq j$  dan  $W_{ij} = 0$  untuk  $i = j$ .

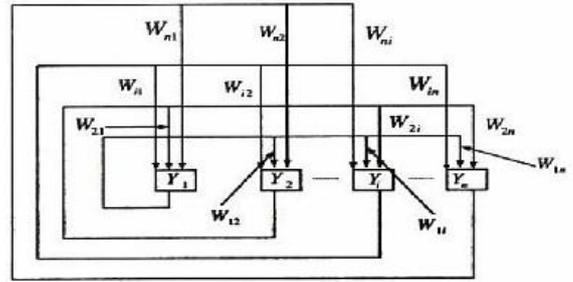
$$W_{ji} = \begin{bmatrix} 0 & w_{12} & w_{13} & w_{14} & w_{15} & w_{16} \\ w_{21} & 0 & w_{23} & w_{24} & w_{25} & w_{26} \\ w_{31} & w_{32} & 0 & w_{34} & w_{35} & w_{36} \\ w_{41} & w_{42} & w_{43} & 0 & w_{44} & w_{45} \\ w_{51} & w_{52} & w_{53} & w_{54} & 0 & w_{55} \\ w_{61} & w_{62} & w_{63} & w_{64} & w_{65} & 0 \end{bmatrix}$$

Gambar 1. Matrik Bobot Jaringan Hopfield Sebagai Vektor W

Ditunjukkan bahwa bobot –bobot yang terletak pada diagonal utamanya adalah nol yang menunjukkan bahwa neuron pada jaringan tersebut tidak memiliki hubungan dengan dirinya sendiri ( $W_{ij} = 0$ ;  $i = j$ ). Sementara itu kesimetrisan vektor bobot berarti berlakunya

$$W_{ij} = W_{ji} \text{ dimana } i \neq j,$$

sehingga  $w_{12}=w_{21}$ ,  $w_{13}=w_{31}$ , ..., dan seterusnya.



Gambar 2. Arsitektur Jaringan Hopfield

Dari gambar 2 di atas, dapat dijelaskan bahwa luaran setiap simpul diumpanbalikan ke input dari simpul – simpul lainnya melalui bobot koneksi  $W_{ij}$  yang tetap. Pada awalnya nilai  $W_{ij}$  diinisialisasi menggunakan algoritma Hopfield untuk sebuah pola yang masuk. Pola yang tidak dikenal dimasukkan ke dalam jaringan pada waktu kondisi nol. Kemudian jaringan akan beriterasi sampai output yang dihasilkan konvergen. Kondisi luaran yang bersifat konvergen terjadi bila pola output yang dihasilkan tidak berubah sampai pengulangan selesai. Pola yang dinyatakan oleh simpul luaran setelah jaringan konvergen disebut luaran jaringan syaraf tiruan.

**Algoritma Pembelajaran JST Hopfield**

Terdapat sejumlah versi algoritma Hopfield. Untuk deskripsi dan algoritma Hopfield menggunakan input vektor yang berupa angkatan biner. Untuk menyimpan sekumpulan pola biner digunakan notasi:

$$U_i, i = (1,2,3,\dots,n) \tag{2}$$

dimana

$n$  = jumlah pola yang dilatihkan

$U_i$  = elemen i dari vektor  $U$

Sedangkan untuk menyimpan bobot matriks  $W = \{W_{ji}\}$ , dengan notasi:

$$W_{ji} = \sum_{i=1}^n [2U_j - 1][2U_i - 1] \tag{3}$$

jika  $i \neq j$  dan  $W_{ji} = 0$  jika  $i=j$

dengan:

$W_{ji}$  = matrik bobot koneksi j ke i

$n$  = jumlah pola yang dilatihkan

$U_i$  = elemen i dari vektor  $U$  yang hanya memiliki nilai 0 dan 1

Sedangkan versi dari deskripsi algoritma Hopfield lainnya menggunakan input vektor yang berupa angka bipolar. Untuk menyimpan sekumpulan angka bipolar digunakan notasi :

$$U_i, i = (1,2,3,\dots,n) \tag{4}$$

Sedangkan untuk menyimpan bobot matriks  $W = \{W_{ji}\}$ , dengan notasi :

$$W_{ji} = \sum_{s=1}^n U_{sj} U_{si} \tag{5}$$

jika  $j \neq i$  dan  $W_{ij} = 0$  jika  $j=i$   
dengan:

- $W_{ji}$  = matrik bobot koneksi  $j$  ke  $i$
- $n$  = jumlah pola yang dilatihkan
- $U_{si}$  = elemen  $i$  dari vektor  $U$  yang memiliki nilai +1 dan -1
- $U_{sj}$  = elemen  $j$  dari vektor  $U$  yang memiliki nilai +1 dan -1

**Fungsi Energi Hopfield**

Tiap kondisi dari jaringan Hopfield mempunyai sekumpulan energi. Nilai ini didefinisikan dengan

$$E = -\frac{1}{2} \sum_j \sum_{i \neq j} W_{ji} u_j u_i \tag{6}$$

Persamaan ini dihubungkan dengan “energi”, meskipun ini tidak menjelaskan definisi energi yang sama seperti pada fisika. Fungsi energi pada persamaan diatas adalah sebuah fungsi objektif yang diminimalkan oleh jaringan. Pembaharuan dari jaringan Hopfield merupakan prosedur konvergen di mana energi dari keseluruhan jaringan akan menjadi semakin kecil, pada akhirnya jaringan akan berada pada kondisi stabil di mana pada kondisi ini energi berada pada nilai minimum. Nilai minimum ini mungkin lokal ataupun global.

Pada tiap saat unit pengolah diperbaharui, energi dari jaringan akan berkurang atau tetap. Sebagai hasilnya, prosedur pembaharuan ini akan selalu membuat energi dari jaringan untuk konvergen pada nilai minimum. Berikut uraian bahwa prosedur pembaharuan akan mengurangi energi atau tetap sama. Misalkan unit  $j$  adalah pengolah berikutnya yang akan diperbaharui. Bagian energi yang dipengaruhi oleh unit pengolah  $j$  adalah :

$$E_j = -\frac{1}{2} \sum_{i \neq j} W_{ji} u_j u_i \tag{7}$$

dimana dapat diformulasikan menjadi

$$E_j = -\frac{1}{2} u_j \sum_{i \neq j} W_{ji} u_i \tag{8}$$

ic Hard Limit dirumuskan sebagai :

ketika unit  $j$  diperbaharui :

$$\Delta E_j = \Delta u_j \left( -\frac{1}{2} \sum_{i \neq j} W_{ji} u_i \right) \tag{9}$$

dengan:

$$\Delta E_j = E_j^{baru} - E_j^{lama}$$

$$\Delta u_j = (u_j^{baru} - u_j^{lama}) \tag{10}$$

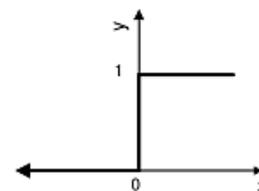
**Fungsi Aktivasi**

Fungsi aktivasi dapat didefinisikan sebagai penguatan non linier dalam sisem analog. Penguatan tersebut dikalkulasi dengan cara mencari rasio perubahan pada fungsi luaran neuron. Fungsi aktivasi tidak hanya digunakan pada saat perhitungan nilai keluaran neuron saja, tetapi turunan pertamanya juga dapat digunakan untuk menghitung perubahan bobot dan bias pada proses learning. Dalam penulisan ini digunakan fungsi aktivasi *threshold bipolar*. Beberapa fungsi aktivasi yang digunakan dalam jaringan syaraf tiruan adalah:

a. Fungsi *Step Function Hard Limit*

Jaringan dengan lapisan tunggal sering menggunakan fungsi ini untuk mengubah input dari suatu variable yang bernilai kontinu ke suatu output biner (0 atau 1). Fungsi ini dirumuskan sebagai berikut :

$$y = \begin{cases} 0, & \text{jika } x \leq 0 \\ 1, & \text{jika } x > 0 \end{cases} \tag{11}$$

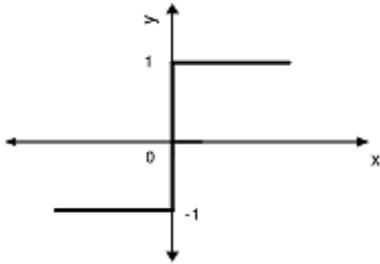


**Gambar 3.** Fungsi Aktivasi: Undak Biner (*Hard Limit*)

b. Fungsi Bipolar (*Symetric Hard Limit*)

Fungsi Bipolar sebenarnya hampir sama dengan fungsi *step*, hanya saja output yang dihasilkan berupa 1, 0 atau -1. Fungsi Symetr

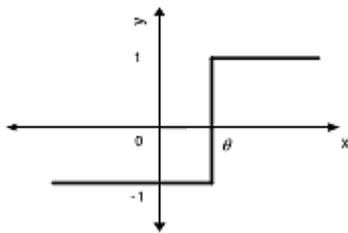
$$y = \begin{cases} 1, & \text{jika } x > 0 \\ 0, & \text{jika } x = 0 \\ -1, & \text{jika } x < 0 \end{cases} \quad (12)$$



Gambar 4. Fungsi Aktivasi: Bipolar (Symmetric Hard Limit)

- c. Fungsi Bipolar (Threshold Function)  
 Fungsi bipolar memiliki keluaran yang dihasilkan berupa 1, atau -1. Fungsi bipolar (dengan nilai ambang  $\theta$ ) dirumuskan sebagai:

$$y = \begin{cases} 1, & \text{jika } x \geq \theta \\ -1, & \text{jika } x < \theta \end{cases} \quad (13)$$



Gambar 5. Fungsi Aktivasi: Bipolar (Threshold Function)

**Metode Penelitian**

1. Membentuk dimensi array untuk ke 3 pola sehingga membentuk array 3x3. Function array-nya bernama pola() 9 kekanan dan 9 kebawah matrik bobot. Elemen array-nya didapat dari perhitungan bobot dari inputan.
2. Agar pemahaman pembacaan kerja program lebih mudah, maka perlu ditampilkan pola bobot dari inputan yaitu w11...w99
3. Data uji berupa pola karakter yang rusak tapi mirip dengan pola karakter "+", "x", "o".

**3. HASIL DAN PEMBAHASAN**

Cara kerjanya, diberikan data uji coba pola yang menyerupai pola karakter yang diharapkan (pada matrik x) kemudian dilakukan training, dilakukan perbaikan dan dari training yang ada, mana yang training errornya paling kecil atau perbedaannya paling sedikit. Karena pola berupa matrik 3x3, bila error trainingnya ada yang

berjumlah sama maka cenderung dianggap pola pertama. Disini diberikan 1 data testing untuk masing-masing pola.

```

C:\Windows\system32\cmd.exe
*****
Implementasi Algoritma Hopfield
Pada Perbaikan Citra
Kukuh Yudhistiro 2013
*****
Matrik bobot, W[9][9]=
w11 w12 w13 w14 w15 w16 w17 w18 w19
w21 w22 w23 w24 w25 w26 w27 w28 w29
w31 w32 w33 w34 w35 w36 w37 w38 w39
w41 w42 w43 w44 w45 w46 w47 w48 w49
w51 w52 w53 w54 w55 w56 w57 w58 w59
w61 w62 w63 w64 w65 w66 w67 w68 w69
w71 w72 w73 w74 w75 w76 w77 w78 w79
w81 w82 w83 w84 w85 w86 w87 w88 w89
w91 w92 w93 w94 w95 w96 w97 w98 w99

Hasil:
0-1 3-1-1-1 3-1 3
-1 0-1 3-1 3-1 3-1
3-1 0-1-1-1 3-1 3
-1 3-1 0-1 3-1 3-1
-1-1-1-1 0-1-1-1-1
-1 3-1 3-1 0-1 3-1
3-1 3-1-1-1 0-1 3
-1 3-1 3-1 3-1 0-1
3-1 3-1-1-1 3-1 0
    
```

Gambar 6. Matrik Bobot Berdimensi 9x9

Pola[0];	Pola[1];	Pola[2];
1 -1 1	-1 1 -1	1 1 1
-1 1 -1	1 1 1	1 -1 1
1 -1 1	-1 1 -1	1 1 1
* *	*	****
* *	****	* *
* *	*	****

Gambar 7. Matrik untuk Setiap Pola Pelatihan

Berikut vektor untuk masing-masing pola pelatihan:

```

{New Integer() {1, -1, -1,
                -1, 1, -1,
                1, -1, 1}},
New Integer() {-1, 1, -1,
                1, 1, -1,
                -1, 1, -1}},
New Integer() {1, 1, 1,
                1, 1, -1,
                1, 1, -1}}
    
```

Masukan pola yang rusak akan dikenali yang mendekati atau perbedaan matriknya lebih sedikit ke pola pelatihan masing-masing terlihat pada gambar berikut:

```
----- Bentuk Pelatihan ke[0] -----
*
*
*
Perbaikan ke: 0, dikenali sebagai pola:
*
*
*
Perbaikan ke: 1, dikenali sebagai pola:
*
*
*
*
```

Gambar 8. Perbaikan Pola yang Mendekati Pola “x”

```
----- Bentuk Pelatihan ke[1] -----
*
***
*
Perbaikan ke: 0, dikenali sebagai pola:
*
***
*
Perbaikan ke: 1, dikenali sebagai pola:
*
***
*
*
```

Gambar 9. Perbaikan Pola yang Mendekati Pola “+”

```
----- Bentuk Pelatihan ke[2] -----
***
***
***
Perbaikan ke: 0, dikenali sebagai pola:
***
***
***
Perbaikan ke: 1, dikenali sebagai pola:
***
*
***
Press any key to continue . . .
```

Gambar 10. Perbaikan Pola yang Mendekati Pola “o”

[4] SK, Roswaldi. *Pengontrolan Kualitas Keragaman Warna Produk Industri Menggunakan Metode Neural Network*. Padang: 2009

#### 4. KESIMPULAN DAN SARAN

Dari hasil percobaan perbaikan pola karakter menggunakan algoritma hopfield di atas diperoleh bahwa restorasi dapat berjalan dengan optimal pada vektor berdimensi 3x3, namun ada kemungkinan hasil restorasi dapat mengalami kesalahan pada vektor berdimensi lebih besar. Maka disarankan untuk penelitian mendatang bisa menggunakan vektor atau pixel yang lebih besar.

#### 5. REFERENSI

[1] Ahmad, U. *Pengolahan Citra Digital Dan Teknik Pemrogramannya*. Yogyakarta : Graha Ilmu: 2005

[2] Gonzalez, R.C & Richard, E.W., *Digital Image Processing*, Massachusetts, Addison-Wesley Publishing:2002

[3] Maryanto, H., Suwarno, Sri., Krisnawarti, L.D. *Restorasi Citra Blur Dengan Algoritma Jaringan Syaraf Tiruan Hopfield*. Yogyakarta:2010