

## Aplikasi Pengubah Citra Nominal ke Bentuk Terbilang

Tito Cahyo Prasetyo Pembimbing: Ir. Muhammad Aswin, MT. dan Adharul Muttaqin, ST., MT.

**Abstrak**-Saat ini media pembelajaran sangat beragam, apalagi yang berkenaan dengan masalah IT. Saat ini sejak tingkat pendidikan play group sudah diperkenalkan tentang keberadaan angka serta berhitung. Pada jaman dulu dan mungkin pada saat ini seorang guru masih mengajarkan kepada muridnya secara verbal tentang angka-angka itu sendiri. Dengan memanfaatkan teknologi komputer yang semakin berkembang tentunya dapat membantu manusia dalam mengenali suatu angka itu sendiri.

Salah satu ilmu komputer yang berkaitan dengan identifikasi objek, dalam hal ini lebih khususnya adalah Citra Karakter Angka adalah pengenalan pola. Pengenalan pola adalah merupakan suatu teknik untuk mengenali pola atau karakteristik dari suatu objek. Terdapat beberapa teknologi algoritma pada pengenalan pola, salah satunya yang akan digunakan adalah teknik Integral Proyeksi. Dengan mengimplementasikan teknologi ini akan mampu membantu sistem pembelajaran dalam pengenalan karakter angka.

Pada skripsi ini dibuat suatu aplikasi untuk mengidentifikasi jenis karakter angka dalam bentuk citra menggunakan bantuan alat Scanner. Citra ditangkap oleh scanner kemudian dilakukan proses ekstraksi ciri. Ciri yang diambil adalah Fitur dari tiap karakter angka. Fitur adalah suatu ciri yang diambil dari penjumlahan pixel secara vertikal dan horizontal dari citra karakter itu sendiri. Setelah didapat index-index dari fitur tadi, langkah selanjutnya adalah memperhitungkan dengan rumusan RMSE (*root mean square error*). Dari hasil akhir perhitungan tersebut didapatlah nilai terendah yang merupakan fitur terdekat dari Template yang sudah dideklarasikan terlebih dahulu.

**Kata kunci:** *Citra Karakter Angka, Integral Proyeksi, Scanner, Fitur, RMSE, Template*

### 1. PENDAHULUAN

Kemajuan teknologi informasi saat ini dapat dikatakan sangat pesat dan mampu menangani semua masalah yang berhubungan dengan pengolahan informasi. contoh real saat ini adalah dengan adanya system informasi dengan metode pengolahan data yang begitu complex

yang sebelumnya manusiapun tidak mampu menangani dengan adanya system ini hal yang sedemikian rupa telah terpecahkan. Informasi WEB yang, suatu teknologi yang kita tidak bias pungkiri lagi bahwa keberadaannya sangat membantu manusia dalam share informasi. pengolahan Citra pun sekarang sudah mulai digalakkan, Contoh nyata pemantauan garis marka di salah beberapa jalanan Jakarta sangat membantu polisi dalam menertibkan para pengendara jalan. dan masih banyak lagi teknologi informasi saat ini yang banyak membantu manusia dalam memecahkan masalah sehari-hari.

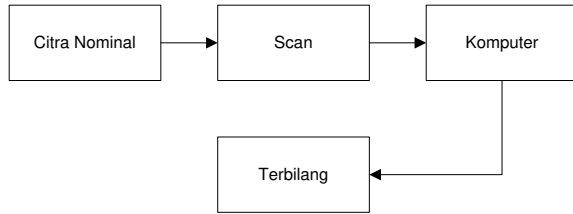
Namun walaupun demikian masih banyak hal yang harus kita kembangkan dalam disiplin ilmu teknologi informasi ini. misalnya saja hal-hal yang berkaitan dengan pengolahan citra. hal ini masih sangat jarang dikembangkan. kita lihat permasalahan yang ada sekarang dalam hal salah pembacaan akibat kurang mampunya pengelihatan manusia yang bias dikarenakan factor usia masih belum terpecahkan. dalam hal ini, teknologi informasi yang menangani masalah demikian pun masih belum banyak tersentuh atau bahkan belum dikaji.

Oleh karena itu dalam penelitian ini dikembangkan suatu perangkat lunak yang mampu mengenali sebuah tulisan tangan nominal dalam kehidupan sehari-hari. misalkan Cek, kwitansi dan hal-hal yang berhubungan dengan keuangan sangat riskan sekali apa bila terjadi salah pembacaan oleh human error. karena hal-hal tersebut sangat erat kaitannya dengan keuangan. perangkat lunak ini bias dikembangkan dalam banyak bidang, misalnya saja sebagai perangkat lunak pembantu dalam proses pembelajaran pengenalan nominal dan angka-angka. bahkan penulis berharap ada yang bersedia mengembangkan perangkat lunak ini kedalam output suara digital.

## 2. PEMBAHASAN

### 2.1. Blok Diagram Sistem

Garis besar design yang akan dibuat untuk aplikasi pengubah citra nominal ke bentuk terbilang terdiri dari beberapa langkah. Berikut ini blok diagram yang ditunjukkan dalam Gambar berikut:



**Gambar 2.1** Diagram blok sistem  
(Sumber: Perancangan)

## 2.2. Perancangan Perangkat Lunak

Perancangan perangkat lunak dibangun dengan menggunakan bahasa pemrograman C# dengan software IDE Microsoft Visual Studio 2010. System ini dirancang dengan spesifikasi sebagai berikut :

1. Melakukan *Grayscale* untuk mengabu-abukan gambar.
2. Melakukan Threshold mengubah image menjadi warna biner hitam dan putih saja.
3. Melakukan Segmentasi character mambagi image menjadi beberapa bagian.
4. Melakukan Resize Image untuk memperkecil Image.
5. Melakukan Integral Proyeksi supaya mendapat cirri dari tiap image karakter (extraksi ciri).
6. Melakukan Template Matching untuk mencocokkan dengan template yang sudah dideklarasikan sebelumnya.

Proses pembuatan program dalam hal ini pengolahan citra secara detail design aplikasi secara umum ditunjukkan pada Gambar 4.3 berikut:

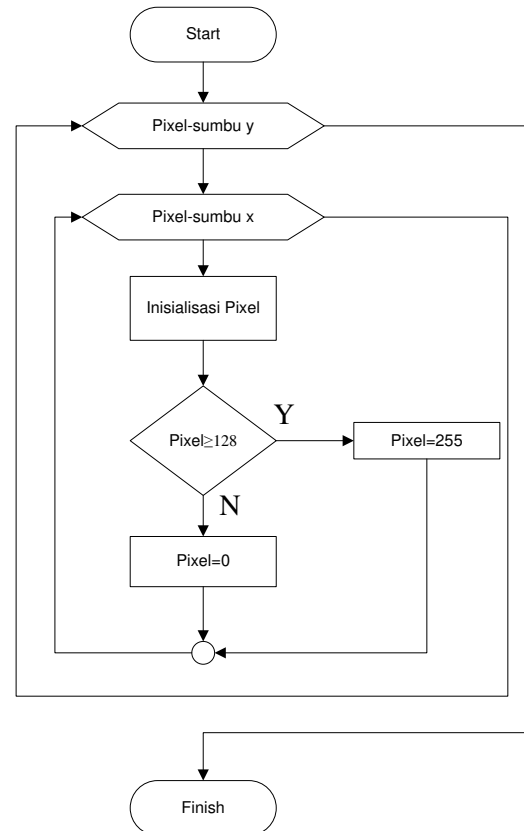
### 2.2.1. Perancangan Program Pengolahan Citra

Perancangan dalam hal ini terdapat dalam beberapa tahap, untuk lebih jelasnya perhatikan proses tahapan dibawah ini:

#### 2.2.1.1. Operasi *Threshold*

Sebuah citra memiliki nilai dengan range antara 0-255 (total 256). Hal ini merepresentasikan tingkat keabu-abuan dengan dapat secara jelas kita lihat bahwa bila tingkat itu bernilai 0 maka pixel tersebut berwarna putih. Kemudian bila tingkat keabua-abuan sebuah pixel tersebut bernilai 255,maka pixel itu berwarna hitam pekat. Dalam proses ini hanya diperkenankan warna monochrome. Untuk pixel dengan tingkat keabua-abuan 128 kebawah,maka pixel tersebut secara mutlak dijadikan bernilai 0.hal ini berarti dengan warna yang begitu terang/cerah akan dijadikan putih. Begitu juga sebaliknya apabila lebih atau sama dengan 128 maka

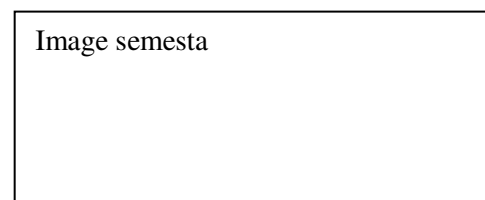
akan dijadikan berniali 255 atau warna hitam. Untuk lebih jelasnya simak diagram flowchar dibawah ini:



**Gambar 2.2** Flowchar Threshold  
(Sumber: Perancangan)

#### 2.2.1.2. Operasi *Segmentasi Karakter*

Segmentasi karakter ini digunakan untuk memecah karakter pada citra karakter. Pertama kali citra asli memiliki ukuran 1750x1275 pixel. Sedangkan citra yang diinginkan hanyalah berukuran 295x950. Pada proses segmentasi metodenya adalah pertama kali menentukan titik yang akan dijadikan referensi dimana/daerah mana yang akan di crop/potong. Bilamana kita ilustrasikan ada sebuah image dengan resolusi X dan Y. untuk lebih jelasnya bisa dilihat gambar ilustrasi di bawah ini:



**Gambar 2.4** Ilustrasi Proses Cropping  
(Sumber: Perancangan)

Pada gambar diatas sangat jelas bahwa untuk mendapatkan sebuah cropping image maka ada 3 variable yang dibutuhkan yakni titik referensi, X dan Y. Maka hasil yang didapat adalah sebuah image dengan resolusi X dan Y dengan komposisi pixel yang ada pada bagian Image semesta.

**2.2.1.3. Operasi Resize**

Proses yang digunakan adalah menggunakan metode bicubic interpolasi. Dimana pada setiap 4 pixel yang berkawan akan dibuat semacam rata-rata warna. Tetapi untuk hal ini rata yang dimaksud adalah warna mayoritas, hitam atau putih. Jadi secara lebih jelas, 4 piksel terdekat dijadikan patokan. Pada perumpamaan ini pixel yang paling pojok kita beri indek dan nama F(p,q) ada pixel terdekatnya yakni F'(p+1,q), F''(p,q+1) dan F'''(p+1,q+1). perhatikan ilustrasi dibawah ini:

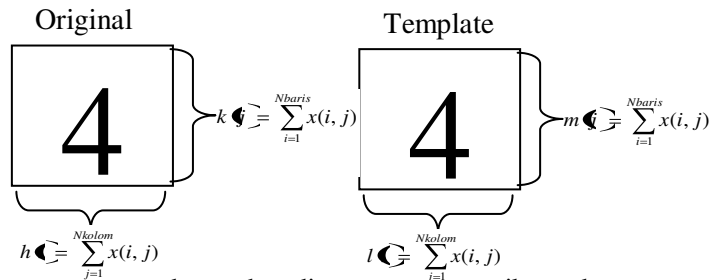
P(n,m)	P(n+1,m)	P(n+2,m)
P(n,m+1)	P(n+1,m+1)	P(n+2,m+1)
P(n,m+2)	P(n+1,m+2)	P(n+2,m+2)

Gambar diatas mendeskripsikan ada 3x3 pixel dengan variasi index yang berbeda. Pada kasus kali ini karena sebelumnya ada proses yang dinamakan threshold yaitu binierisasi, variasi yang didapat yaitu nilai 0 atau 1. Kemudian pada proses ini diambil sampel yaitu 4 pixel diantaranya P(n,m);P(n+1,m);P(n,m+1) dan P(n+1,m+1). Nilai yang akan dirubah nantinya yaitu P(n,m) penetapan nilai tersebut ditentukan berdasarkan rata-rata dari semua nilai pada 4 pixel itu. Dengan rumus rata seperti berikut:

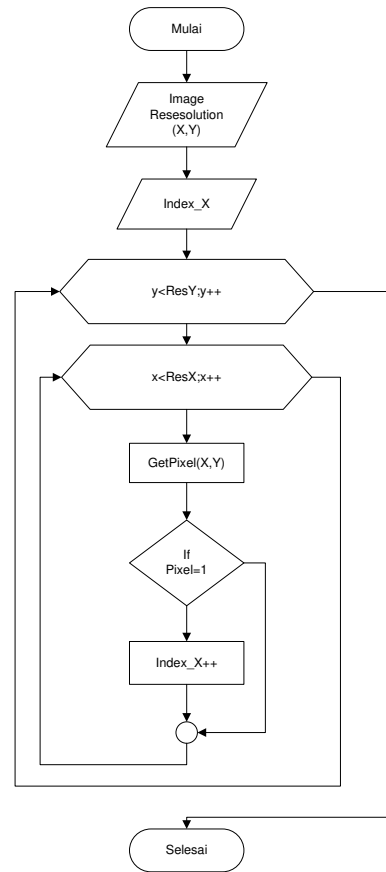
$$R_t = \frac{P(n,m) + P(n+1,m) + P(n,m+1) + P(n+1,m+1)}{4}$$

Dari rumus diatas pasti terdapat variasi nilai yaitu 1, 0.75, 0.5, 0.25, 0. Dari sini dapat ditetapkan bilamana nilai dari Rt adalah 1/0.75/0.5 maka nilai dari P(n,m) adalah 1 atau warna hitam dan bilamana Rt bernilai 0.25/0 nilai P(n,m) adalah 0 atau berwarna putih. Penetapan nilai diatas berdasarkan dari, agar supaya mendapatkan suatu titik/pixel hitam. Dalam artian pengambilan warna hitam lebih diprioritaskan daripada pengambilan warna putih. Dikarenakan pada template matching pencocokan berdasarkan warna hitam. Untuk secara algoritma flowchart bisa dilihat pada diagram dibawah ini:

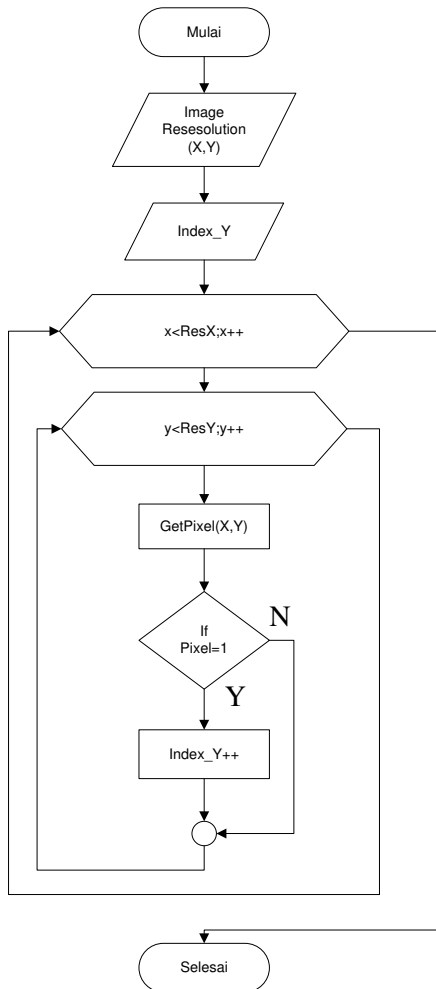
**2.2.1.4. Operasi Integral Proyeksi**



Pada gambar diatas mengiusasikan ada dua buah Image yang secara harfiah memiliki informasi karakter sama, yaitu angka 4. Dengan mengextrasi ciri dari masing-masing karakter tersebut maka dapatlah suatu ciri yang merupakan perpaduan dari penjumlahan baris dan kolom. Perpaduan baris dan kolom tersebut biasa disebut dengan *Fitur*. Untuk mendapatkan ciri baris dan kolom tersebut digunakan suatu metode yang disebut dengan Integral Proyeksi. Perhatikan diagram berikut untuk lebih jelas mengenai proses Integral Proyeksi dalam hal ekstrasi ciri:



**Gambar 2.5** Flowchar Proses Integral Proyeksi Fitur Index Kolom (Sumber: Perancangan)



**Gambar 2.6** Flowchar Proses Integral Proyeksi Fitur Index Baris  
(Sumber: Perancangan)

Setelah dilakukan proses ekstrasi ciri, didapalahr sebuah Informasi dengan format sebagai berikut. Dan informasi tersebut berlaku sama pada image template yang sudah dilaksanakan serupa.

- $\vec{I} = \{k(1), k(2), k(3), \dots k(15)\}$
- $\vec{I} = \{h(1), h(2), h(3), \dots h(15)\}$
- $\vec{T} = \{m(1), m(2), m(3), \dots m(15)\}$
- $\vec{T} = \{l(1), l(2), l(3), \dots l(15)\}$

fitur Template hasil dari perhitungan Integral proyeksi yang sama pula. Untuk yang berindex k

dan m, merupakan hasil perhitungan Integral proyeksi kolom. Hal ini dapat kita lihat jumlah komponen sebanyak 30 buah, yang menggambarkan jumlah pixel dalam kolom. Sebaliknya berbeda dengan yang berIndex h dan l. yang merupakan hasil perhitungan Integral proyeksi baris, hal ini ditandai dengan jumlah komponen yang sebanyak 15 buah dan representasi pula dari jumlah pixel lebar Image.

**2.2.1.6. Operasi Template Matching**

Template matching merupakan proses tahap akhir dimana untuk mendapatkan informasi apakah karakter yang telah didapat fiturnya di Integral proyeksi tersebut. Pada intinya template matching tidak hanya mencocokkan saja, tetapi juga ada perhitungan statistik yang disebut dengan RMSE (Root Mean-Square Error). Perhitungan tersebut melibatkan beberapa operasi matematika. Diantaranya pengurangan, pangkat, akar kuadrat, pembagian dan rata-rata. Untuk lebih jelas rumus RMSE adalah sebagai berikut:

$$RMSE_b(I_k, T_m) = \sqrt{\frac{\sum_{i=1}^{15} (k_i - m_i)^2}{15}}$$

$$RMSE_k(I_h, T_l) = \sqrt{\frac{\sum_{i=1}^{15} (h_i - l_i)^2}{15}}$$

Pada rumusan diatas akan mendapatkan hasil yang menentukan atau memutuskan persamaan dengan karakter template. Berikut gambaran toleransi serta keputusan kecocokan dengan karakter templatanya:

Kondisi match → RMSE = nilai terkecil dari perhitungan template

**2.3. Implementasi**

Program Aplikasi Pengubah Citra Nominal ke Bentuk Terbilang mempunyai tampilan seperti pada gambar dibawah ini. Tampilan dibuat semenarik mungkin, supaya para user tidak lelas dan jenuh melihat sesuatu yang biasa saja. Pada dasarnya memang program ini diperuntukkan untuk media elektronik pembelajaran mengenai angka itu sendiri. Jadi sasaran yang tepat adalah para anak-anak yang saatnya untuk belajar mengenai angka dan bilangan. Perhatikan tamplian pada berikut ini:



**Gambar 2.8** Tapilan awal ketika pertama kali di Runing  
(Sumber: Perancangan)

### 3. PENGUJIAN

Untuk mengetahui sistem bekerja dengan baik dan sesuai dengan perancangan, maka diperlukan serangkaian pengujian. Pengujian yang dilakukan dalam bab ini adalah sebagai berikut:

1. Pengujian program berdasarkan Image dengan berbagai variasi.
2. Analisa faktor kegagalan.
3. Kesimpulan hasil pengujian.

#### 3.1. Pengujian program berdasarkan Image dengan kondisi terdapat noise

Adapun pada saatnya suatu image tidak selalu dalam keadaan bersih. Adapun faktor dari media kertas yang diprint. Untuk lebih jelasnya perhatikan gambar berikut:



**Gambar 2.8** pengujian dengan adanya noise pada background  
(Sumber: Pengujian)

Pada gambar hasil pengujian diatas terdapat beckground pada satu karakter angka yakni karakter angka 7. Hal tersebut masih bisa ditolelir. Karena masih lebih kontras antara background dan karakter tersebut.

#### 3.2. Pengujian program berdasarkan Image dengan kondisi terdapat noise

Adapun tujuan dari pengujian ini adalah mengetahui juga apakah program tersebut efektif pula bilamana Image memiliki karakter dengan Font yang berbeda dengan font yang membentuk informasi Template. Pada kasus ini template terbentuk dari jenis font *Calibri*. Berikut ini adalah tabel yang merepresentasikan beberapa dari jenis font berbeda beserta kesensitifannya:

Jenis Font	Respon Program
Calibri	Terdeteksi
Times New Roman	Tidak Terdeteksi
Arial Narrow	Terdeteksi
Book Antiqua	Terdeteksi
FELIX TITLING	Terdeteksi
<i>Mistral</i>	Tidak Terdeteksi
Franklin Gothic Medium Cond	Terdeteksi

**Table 5.1** tabel pengujian terhadap berbagai jenis font

(Sumber: Pengujian)

#### 3.3. Pengujian program dengan warna font berbeda

Pada pengujian ini bertujuan untuk mengetahui kesensitifan program pada font yang berwarna serta terdapat noise. Pada gambar 5.4 dapat dilihat hasil yang tidak sesuai. Karena satu dianggap tidak ada angkanya. Kemudian dapat dilihat pula pada gambar 5.5 hal tersebut tidak akan menjadi masalah karena pada waktu resize sudah tereduksi dan banyak yang menghilang. Hasilnya noise dapat terhindarkan karena hal tersebut.



**Gambar 2.9** pengujian dengan Font yang berwarna serta terdapat noise  
(Sumber: Pengujian)

### 4. PENUTUP

#### 4.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis sistem maka dapat diambil kesimpulan sebagai berikut:

1. Proses aplikasi pengubah citra nominal ke terbilang pada intinya adalah menggunakan teknik *Integral Proyeksi*. Tetapi tidak luput juga bahwa proses lain selain proses tersebut diantaranya yakni *Threshold*, *Segmentasi karakter*, *Resize*, dan *Template Matching*.
2. Untuk membedakan objek karakter satu dan objek karakter yang menggunakan *Template Matching*. Yang pada dasarnya sebelum kompilasi program template sudah dideklarasikan terlebih dahulu. Dimana karakter yang mempunyai nilai perbedaan *Fitur* paling sedikit berarti itulah yang paling mendekati dengan karakter Template.
3. Keberhasilan Program aplikasi dipengaruhi oleh Citra yang bagus dan bersih, sesuai dengan font yang ditentukan yakni jenis *Calibri*, serta terdapat kontras warna yang cukup signifikan pada Karakter dan Background.
4. Program aplikasi ini sudah dapat diaplikasikan sebagai media pembelajaran sebagaimana mestinya karena dari hasil pengujian sudah cukup sesuai.

#### 4.2. Saran

Dalam perancangan dan pembuatan aplikasi pengubah citra nominal ke terbilang ini masih terdapat kekurangan atau kelemahan, oleh karena itu masih diperlukan adanya penyempurnaan dalam rangka pengembangan kemas depan. Adapun hal yang dapat disempurnakan atau perbaiki antara lain:

1. Memungkinkan untuk dapat menyediakan template yang sangat banyak supaya karakter apa pun dapat dikenali.
2. Menggunakan *Scanner* yang baik supaya Citra dapat terminimalisir dari gangguan ataupun noise yang dapat mempengaruhi pemrosesan program aplikasi

Academy of Sciences of the USA, vol. 79 no. 8 pp. 2554-2558, April 1982.

- Hertz, J., Krogh, A., & Palmer, R.G. (1991). Introduction to the theory of neural computation. Redwood City, CA: Addison-Wesley.
- McCullough, W.S., & Pitts, W.H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115-133
- Polyn, S.M., & Kahana, M.J. (2008). Memory search and the neural representation of context. *Trends in Cognitive Sciences*, 12, 24-30.
- Rizzuto, D.S., & Kahana, M.J. (2001). An autoassociative neural network model of paired-associate learning. *Neural Computation*, 13, 2075-2092.
- Joe Tebelskis (1995) Speech Recognition using Neural Networks School of Computer Science Carnegie Mellon University
- Henry Cohn, Chris Umans. A Group-theoretic Approach to Fast Matrix Multiplication. arXiv:math.GR/0307321. Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 11–14 October 2003, Cambridge, MA, IEEE Computer Society, pp. 438–449.
- Horn, Roger A.; Johnson, Charles R. (1985), *Matrix Analysis*, Cambridge University Press, ISBN 978-0-521-38632-6
- Knuth, D.E., *The Art of Computer Programming Volume 2: Seminumerical Algorithms*. Addison-Wesley Professional; 3 edition (November 14, 1997). ISBN 978-0201896848. pp. 501.

#### DAFTAR PUSTAKA

- Jonas Gomes, Luiz Velho. 2005. *Image Processing For Computer Graphics*. Springer.
- J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National*