

Penerapan Algoritma Boyer-Moore Untuk Menentukan Pola Kesamaan Pada *Retinal Recognition*

Anwar Ramadha

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
anwar.ramadha@gmail.com

Abstract—Di zaman sekarang, keamanan yang tidak kalah penting adalah keamanan dibidang informatika, karena semua data dapat di bagikan di internet. Maka dari itu di ciptakanlah sebuah metode keamanan yang kuat yang hanya dapat diakses oleh orang-orang tertentu saja. Salah satunya adalah dengan menggunakan *Retinal Recognition*. Metode ini adalah salah satu metode dari teknologi canggih yang disebut dengan *biometric scanning*. Teknologi ini digunakan untuk mengambil data unik dari bagian tubuh manusia yang dapat digunakan untuk mengidentifikasi seseorang. Contoh lain dari *biometric scanning* adalah pemindaian sidik jari, dan pengenalan wajah wajah. Makalah ini akan fokus pada pembahasan tentang *retinal recognition*.

Hasil dari pengambilan data adalah sebuah data digital yang jumlah *byte*-nya kecil merepresentasikan citra, sehingga tidak membutuhkan usaha dan waktu yang banyak untuk melakukan identifikasi. Pada makalah ini akan digunakan algoritma pencocokan *string* untuk mencocokkan *byte-byte* yang disimpan dalam sistem dan masukan.

Kata Kunci — *boyer moore; retinal recognition; biometrics; string matching;*

I. PENDAHULUAN

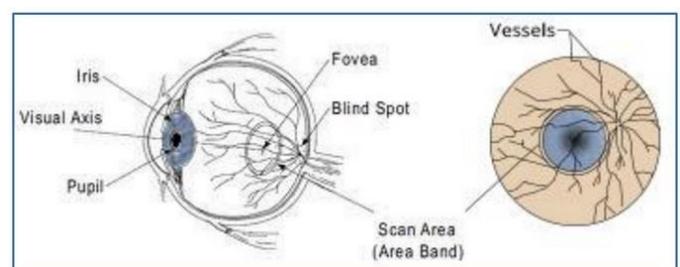
Di zaman ini, dimana saat teknologi informasi berkembang dengan pesat dan semua informasi dapat diberikan secara cuma-cuma maka kemanan pun semakin terancam, terutama keamanan dibidang informatika. Semua data dapat diakses oleh orang-orang yang tidak bertanggung jawab dan tidak memiliki hak akses. Maka dari itu diciptakanlah sebuah metode dan/atau teknologi canggih yang dianggap memiliki kemanan yang sangat kuat, yaitu sistem biometrik. Sistem ini (atau disebut juga *biometric scanning*) adalah metode yang menggunakan salah satu anggota tubuh manusia yang unik dan dapat membedakan atau mengidentifikasi seseorang. Contohnya adalah sidik jari, retina, dan pengenalan wajah. Tetapi saat ini yang paling aman dan dapat diterapkan semua orang adalah pemindai retina, karena bisa saja sidik jari seseorang terhapus karena terlalu banyak terpapar atau

mengonsumsi bahan kimia, dan juga pengenalan wajah tidak terlalu efisien karena ada beberapa orang yang mirip dengan orang lainnya.

Pengenal retina (*Retinal Recognition*) akan mengubah citra menjadi data digital untuk selanjutnya diproses menggunakan algoritma yang lebih simpel karena hanya menggunakan *byte-byte* pada sistem computer dan juga tidak membutuhkan banyak waktu dan usaha.

II. RETINAL RECOGNITION (PENGENAL RETINA)

Pada *eye biometric*, bagian mata yang sering digunakan adalah iris dan retina. Jika mata manusia dimisalkan dengan sebuah kamera, iris adalah bagian *aperture* sedangkan retina adalah bagian film dari kamera. Retina mata manusia mempunyai jutaan *fotoreseptor* (penangkap cahaya) dan jaringan sensor yang keduanya dapat mengubah cahaya menjadi impuls listrik, serta terdapat banyak pembuluh darah yang menjadi landasan untuk membangun *retinal recognition*. Karena retina terletak pada bagian mata terdalam dan tidak tersentuh bagian luar, maka retina menjadi bahan biometrik paling stabil.



Gambar 1 : Mata dan *scan area* yang digunakan pada *retinal recognition*. Sumber http://www.tutorialspoint.com/biometrics/images/retinal_scanning.jpg

Pada gambar 1 diperlihatkan bagian-bagian mata dan bagian yang digunakan untuk pengambilan data pada *retinal recognition*.

Pada *retinal recognition* peralatan yang digunakan biasanya menggunakan sinar inframerah berkekuatan rendah. Hal ini dilakukan untuk menghindari resiko kerusakan pada retina. Sinar inframerah tersebut berfungsi untuk menyinari pembuluh-pembuluh darah pada retina. Sebuah penelitian menunjukkan bahwa sinar inframerah lebih cepat diserap oleh pembuluh darah, sehingga pembuluh-pembuluh darah yang memiliki pola unik tersebut merefleksikan seberkas cahaya, dan cahaya inilah yang ditangkap oleh alat pemindai dan merekamnya sebagai pola.

A. Proses Pemindaian Retina

Secara garis besar, proses pemindaian retina dibagi menjadi beberapa proses, yaitu :

1. *Image/signal acquisition and processing* (memproses dan mengakuisisi citra/sinyal).

Pada bagian ini citra biometrik pengguna di pindai dengan menggunakan seberkas cahaya hijau yang melingkar sempurna. Pola dari pembuluh-pembuluh darah ditangkap pada proses ini dan menghasilkan tiga sampai lima pola citra dan kemudian dimasukkan ke dalam sistem biometric. Sistem ini di program untuk memanipulasi citra dengan menggunakan persamaan (*equations*) dan menyimpan hasil dari perhitungan setiap pixel.

Untuk meningkatkan fitur tertentu secara selektif, dan untuk menghilangkan *noise* pada gambar, maka data digital akan dikenai berbagai operasi pengolahan digital. Hal ini dilakukan untuk meminimalis kesalahan dalam pengambilan data.

Metode pengolahan citra dapat dikelompokkan menjadi tiga kategori fungsional, yaitu :

a. *Image Restoration*

Bagian ini secara garis besar dibagi menjadi dua bagian yaitu mengurangi *noise* pada gambar pada saat memperoleh sampel dan menghapus distorsi yang muncul selama pendaftaran biometrik.

Penghalusan citra yang dapat mengurangi *noise* dalam citra dapat dilakukan dengan cara mengganti setiap pixel dengan nilai rata-rata dengan pixel tetangga. Sistem biometrik menggunakan berbagai algoritma penyaringan dan teknik pengurangan *noise* seperti *median filtering*, *adaptive filtering*, histogram statistik, *wavelet transforms*, dll.

b. *Image Enhancement*

Teknik ini digunakan untuk meningkatkan visibilitas bagian atau fitur pada citra dan menekan informasi di bagian lain. Hal ini dilakukan hanya setelah restorasi (*image restoration*) selesai. Bagian ini terdiri dari pencerahan, penyesuaian kontras, dll, sehingga gambar dapat digunakan untuk proses lebih lanjut.

c. *Feature Extraction*

Ada dua tipe fitur yang di ekstrak dari citra, yaitu :

1. *General feature* – termasuk didalamnya adalah bentuk, tekstur, warna, dll, yang digunakan untuk mendeskripsikan isi dari citra.
2. *Domain-specific features*



Gambar 2 : Perbandingan antara citra asli dengan citra yang telah di *enhanced*. Sumber : http://www.tutorialspoint.com/biometrics/images/feature_extraction.jpg.

Ketika fitur diekstrak dari gambar, kita perlu memilih *classifier* yang cocok. Pemilih yang banyak digunakan adalah *Nearest Neighbor classifier*, yang membandingkan vektor fitur kandidat citra dengan vektor dari citra yang tersimpan dalam sistem.

B-Splines adalah perkiraan yang diterapkan untuk menggambarkan pola kurva dalam sistem biometrik sidik jari. Koefisien dari *B-Splines* digunakan sebagai fitur. Dalam kasus pengenalan iris, citra dari iris di dekomposisi menggunakan *Discrete Wavelet Transform (DWT)*, dan koefisien DWT kemudian dijadikan sebagai fitur.

Salah satu keuntungan dari *retinal recognition* terlihat jelas pada tahap ini, contohnya adalah faktor genetik yang tidak menentukan pola pembuluh darah. Retina memiliki keragaman bagian unik. Hal ini memungkinkan untuk menghasilkan *data point* untuk dari retina sampai sejumlah 400 titik. Jumlah ini dapat dikatakan jauh lebih besar dibanding dengan biometrik lain seperti sidik jari yang hanya memiliki 30-40 titik data unik.

Bagian ini menjadi bagian paling sulit dan paling lama dari sebuah *biometric scanning*. Kecepatan dan kemudahan untuk menyelesaikan proses yang bergantung dari penggunaannya. Posisi mata harus benar-benar dekat dari lensa yang akan memindai retina dan pengguna pun tidak boleh bergerak ketika pengambilan citra berlangsung agar diperoleh kualitas citra yang baik.

2. *Matching* (Pencocokan)

Proses ini menggunakan suatu sistem computer untuk memverifikasi dan mengidentifikasi pengguna.

3. Representation

Setiap bagian unik dari retina direpresentasikan menjadi sebuah *template* digital yang berukuran 35-100 *bytes*. *Template* ini merupakan *template* terkecil dalam sistem biometrik saat ini.

III. VERIFIKASI POLA RETINA DENGAN MENGGUNAKAN ALGORITMA STRING MATCHING

Diatas telah dipaparkan bahwa *retinal recognition* dapat memiliki 400 titik dan data tersebut diubah menjadi data digital yang ukurannya berkisar antara 35-100 *bytes*. Alat-alat biometrik yang digunakan sekarang menghasilkan *template* yang berukuran 96 *bytes*.

Ukuran *template* yang kecil inilah yang menjadi dasar penerapan algoritma pencocokan *string*. Dalam makalah ini akan dibahas tentang algoritma Boyer-Moore. Dengan data yang berukuran 96 *bytes* atau setara dengan 96 karakter, maka algoritma yang paling cocok adalah algoritma pencocokan *string* Boyer-moore, karena hanya akan memakan waktu yang sedikit. Dilipiluhnya algoritma ini karena didalam sistem aka nada banyak sekali *template* yang disimpan, sehingga memerlukan algoritma yang mangkus agar proses verifikasi berjalan dengan cepat.

Algoritma Boyer-Moore adalah algoritma pencocokan *string* yang dipublikasikan oleh Robert S. Boyer dan J. Strother Moore pada tahun 1977. Algoritma ini dianggap sebagai algoritma paling efisien pada aplikasi umum. Algoritma ini mencocokkan setiap karakter pada *string* dari sebelah kanan ke kiri, tidak seperti algoritma brute force dan algoritma Knuth-Morris-Pratt. Ide pencocokan dari kanan ke kiri ini didapat karena akan lebih banyak informasi yang akan didapat.

Jika dibandingkan dengan brute algoritma ini memiliki kompleksitas $O(m/n)$ pada kasus terbaik dimana brute force memiliki kompleksitas $O(n)$, sedangkan saat *worst case* algoritma ini memiliki kompleksitas $O(mn+A)$ dimana brute force memiliki kompleksitas $O(mn)$.

Contoh *worst case* pada algoritma ini adalah :

T : "aaaaaaaa...a"

P : "baaaaa"

Pada contoh diatas, ketidakcocokkan selalu terjadi pada awal pattern, sedangkan pencocokkan dimulai dari kanan ke kiri, lalu setelah ditemukan ketidakcocokkan, pattern akan digeser ke $i + 1$, dimana i adalah indeks dari teks.



Gambar 3 : *Worst case* pada algoritma Boyer-Moore.

Sumber : Slide kuliah strategi algoritma "Pencocokan String (2015)".

Algoritma Boyer-Moore mempunyai bagian preproses yang digunakan untuk mencari *Last occurrence* atau kemunculan terakhir dari suatu karakter di teks pada *pattern*. Contohnya jika kita memiliki teks "abcacd" dan sebuah *pattern* "abacab", maka,

X	A	B	C	d
L(X)	5	6	4	-1

Hasil dari fungsi *Last Occurrence* atau L(X) yang nantinya akan digunakan untuk seberapa banyak karakter pada teks harus digeser.

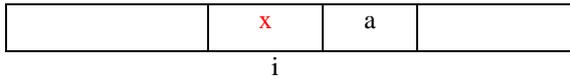
Secara umum, algoritma Boyer-Moore memiliki langkah-langkah sebagai berikut :

1. Algoritma Boyer-Moore mulai mencocokkan pattern pada awal teks.
2. Algoritma akan mulai mencocokkan karakter per karakter dari kanan ke kiri dengan teks yang diberikan, sampai salah satu kondisi berikut terpenuhi :
 - a. Karakter di pattern dan teks yang dibandingkan terjadi ketidakcocokkan (*mismatch*).
 - b. Semua karakter di pattern dan teks cocok dan algoritma akan mengembalikan *true* jika sesuai.
3. Jika ada ketidakcocokkan antara pattern dan teks, algoritma ini memiliki tiga kasus untuk jumlah pergeseran yang akan dilakukan, yaitu :

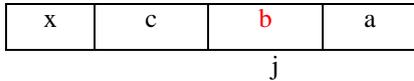
a. Kasus pertama

Jika P mengandung x dimanapun, geser P ke kanan sejajar dengan *Last occurrence* dari x.

T :

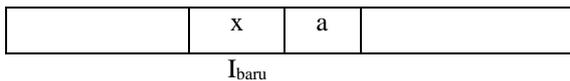


P :

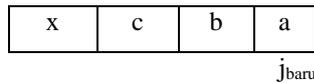


Maka x akan disejajarkan dengan x yang muncul di indeks paling besar pada *pattern*.

T :



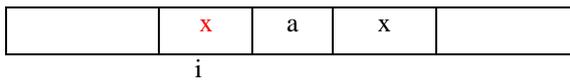
P :



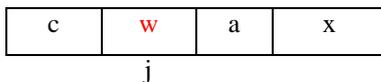
b. Kasus kedua

Jika P mengandung x dimanapun, tetapi menggeser P ke *last occurrence* tidak dimungkinkan, maka geser P ke kanan sebanyak 1 karakter ke $T[i + 1]$.

T :

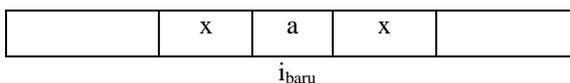


P :

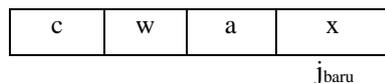


Karena kemunculan terakhir x ada di sebelah kanan j, maka tidak mungkin untuk menggeser *pattern* ke kiri, maka geser *pattern* kanan sebanyak 1 karakter ke $T[i+1]$.

T :



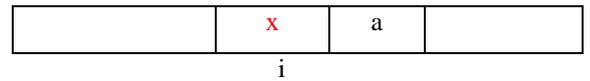
P :



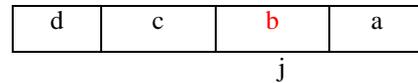
c. Kasus ketiga

Jika kasus pertama dan kedua tidak berlaku (karakter pada teks tidak ada di *pattern*), maka geser P ke kanan dan $P[1]$ sejajar dengan $T[i+1]$

T :

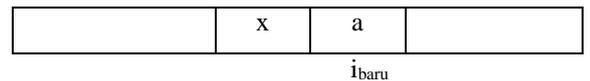


P :

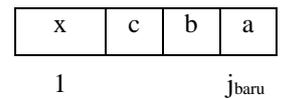


Geser P menjadi

T :



P :



Berikut ini adalah pseudo-code dari algoritma Boyer-Moore.

1. Pseudo-code dari *Last Occurrence function*.

```

procedure preBmBc (
    input P : array[0..n-1] of
char,
    input n : integer,
    input/output bmBc :
array[0..n-1] of integer
)
Deklarasi:
    i: integer

Algoritma:
    for (i := 0 to ASIZE-1)
        bmBc[i] := m;
    endfor
    for (i := 0 to m - 2)
        bmBc[P[i]] := m - i - 1;
    endfor
    
```

```

procedure preSuffixes(
  input P : array[0..n-1] of char,
  input n : integer,
  input/output suff : array[0..n-1] of integer
)
Deklarasi:
  f, g, i: integer
Algoritma:
  suff[n - 1] := n;
  g := n - 1;
  for (i := n - 2 downto 0) {
    if (i > g and (suff[i + n - 1 - f] < i - g))
      suff[i] := suff[i + n - 1 - f];
    else
      if (i < g)
        g := i;
      endif
      f := i;
      while (g >= 0 and P[g] = P[g + n - 1 -
f])
        --g;
      endwhile
      suff[i] = f - g;
    endif
  }
endfor

```

```

procedure preBmGs(
  input P : array[0..n-1] of char,
  input n : integer,
  input/output bmBc : array[0..n-1] of integer
)
Deklarasi:
  i, j: integer
  suff: array [0..RuangAlphabet] of integer
preSuffixes(x, n, suff);
for (i := 0 to m-1)
  bmGs[i] := n
endfor
j := 0
for (i := n - 1 downto 0)
  if (suff[i] = i + 1)
    for (j:=j to n - 2 - i)
      if (bmGs[j] = n)
        bmGs[j] := n - 1 - i
      endif
    endfor
  endif
endfor
for (i = 0 to n - 2)
  bmGs[n - 1 - suff[i]] := n - 1 - i;
endfor

```

2. Pseudo-code untuk algoritma utama boyer-moore

```

procedure BoyerMooreSearch(
  input m, n : integer,
  input P : array[0..n-1] of char,
  input T : array[0..m-1] of char,
  output ketemu : array[0..m-1] of boolean
)
Deklarasi:
  i, j, shift, bmBcShift,
  bmGsShift: integer
  BmBc : array[0..255] of interger
  BmGs : array[0..n-1] of interger
Algoritma:
  preBmBc(n, P, BmBc)
  preBmGs(n, P, BmGs)
  i:=0
  while (i<= m-n) do
    j:=n-1
    while (j >=0 n and T[i+j] =
P[j]) do
      j:=j-1
    endwhile
    if(j < 0) then
      ketemu[i]:=true;
    endif
    bmBcShift:=
BmBc[chartoint(T[i+j])]-n+j+1
    bmGsShift:= BmGs[j]
    shift:= max(bmBcShift,
bmGsShift)
    i:= i+shift
  endwhile

```

Langkah-langkah yang perlu dilakukan untuk memverifikasi pola adalah sebagai berikut :

1. Data *template* yang berada dalam pola direpresentasikan menjadi *string* yang akan digunakan sebagai teks.
2. Mengubah data yang didapat dari hasil pemindaian menjadi *string* yang mempunyai 96 karakter.
3. Lakukan pencocokan dengan algoritma Boyer-Moore.

REFERENCES

- [1] Munir, Rinaldi. "Strategi Algoritma". Informatika, Bandung : 2010.
- [2] Das, Ravi. 2007. *Retinal Recognition*. Keesing Journal of Documents & Identity, issue 22.
- [3] http://www.tutorialspoint.com/biometrics/biometrics_system_security.htm, diakses pada 7 Mei 2016 pukul 20.54.
- [4] Hill, Robert "Buzz". *Retina Identification*. Article is from the book : "Biometrics : Personal Identification in Networked Society", by Anil Jain, Ruud Bolle, and Sharath Pankati.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 7 Mei 2010

A handwritten signature in black ink, appearing to be 'Anwar Ramadha', written on a light blue background.

Anwar Ramadha
13514013