

# Penggunaan Algoritma Pattern Matching Boyer-Moore Pada *Fingerprint Scanner*

Praditya Raudi Avinanto / 13514087

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl Ganesha 10 Bandung 40132, Indonesia  
raudiradz@gmail.com

**Abstrak**—Pada era globalisasi ini, hampir seluruh kegiatan manusia telah menggunakan alat elektronik. Diantaranya ialah pemindai sidik jari. Pemindai sidik jari (*fingerprint scanner*) saat ini sudah banyak digunakan, mulai sebagai *attendance system* (sistem absensi/kehadiran), sebagai *access control* (sistem pengontrol akses ke dalam suatu ruangan, tempat atau sebuah sistem). Makalah ini akan membahas bagaimana proses pemindaian dan pencocokan pada *fingerprint* serta metodenya menggunakan algoritma *pattern matching Boyer-Moore*. *Fingerprint* merupakan alat identifikasi yang menggunakan sidik jari. Sidik jari yang dipindai akan menghasilkan sebuah citra dari pola sidik jari tersebut dengan detil berwarna hitam putih, yang kemudian pola tersebut disimpan lalu dibandingkan untuk setiap proses identifikasi. Ide dari pemrosesan ini ialah menggunakan *pattern matching* yang sudah dimodifikasi yaitu *pattern matching* dua dimensi. Makalah ini berfokus pada algoritma pemrosesan citra (pemindaian dan pencocokan), bukan pada teknis kerja alat *fingerprint*.

**Kata kunci** – *Boyer-Moore, fingerprint, pattern matching, string*.

## I. PENDAHULUAN

Pada era globalisasi ini, manusia mengalami kemajuan pesat dalam bidang teknologi baik teknologi industri, mekanik maupun informasi. Kemajuan teknologi ini memiliki manfaat dan juga kerugian yang sebenarnya tergantung dari cara kita menggunakannya. Dampaknya ialah hampir seluruh kegiatan manusia telah menggunakan alat elektronik.

Beberapa contoh kemajuan atau perkembangan teknologi di bidang informasi dan elektronik ialah terciptanya internet, sosial media, laptop atau computer dengan segala kemajuannya serta gadget – gadgetnya dan lain sebagainya. Dalam pembuatan komputer kita perlu juga yang dinamakan software untuk menyelesaikan suatu persoalan yang spesifik. Pembuatan komputer dan software itu sendiri tidak lepas dari

ilmu dasar sains dan matematik. Diantara semua perangkat teknologi diatas salah satu yang sedang digunakan oleh mayoritas masyarakat dunia ialah *fingerprint scanner*.

Pemindai sidik jari (*fingerprint scanner*) saat ini sudah banyak digunakan, mulai sebagai *attendance system* (sistem absensi/kehadiran), sebagai *access control* (sistem pengontrol akses ke dalam suatu ruangan, tempat atau sebuah sistem) hingga sebagai identitas pribadi seperti terdapat pada SIM (*Surat Izin Mengemudi*) atau *passport*.

*Fingerprint scanner* merupakan penemuan yang brilian karena sampai saat ini kemungkinan sangat kecil untuk melakukan pembobolan terhadap autentikasi pemindaian sidik jari. Karena tingkat keamanan yang sangat ketat ini diperlukan juga algoritma yang sangat kompleks dalam programnya. Selain kompleks algoritma program dalam *fingerprint* haruslah mangkus.

Ide kerja *fingerprint scanner* ialah memindai sidik jari dan merubahnya dalam bentuk citra / *image*, yang kemudian gambar itu disimpan (apabila sedang melakukan pembuatan id) atau dicocokkan dengan data id yang ada (apabila melakukan otentikasi). Tentunya yang disimpan bukanlah keseluruhan gambar melainkan pola yang didapat dari titik-titik tertentu yang akan dijelaskan pada bab dasar teori.

Dalam makalah ini penulis akan membahas bagaimana proses atau algoritma dari pemrosesan citra sidik jari yang secara khusus ialah pencarian titik-titik *minutiae*. Pencarian ini mengimplementasi algoritma *Boyer-Moore* dengan modifikasi pada *pattern* dua dimensi.

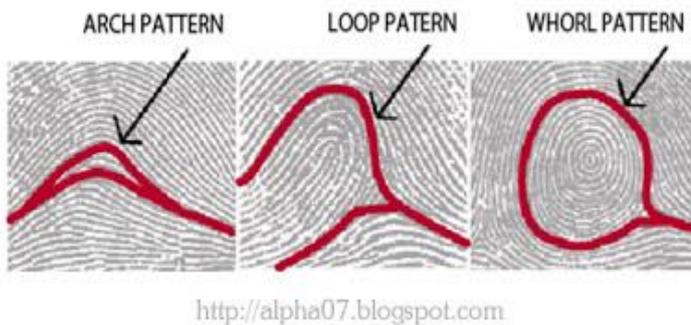
## II. DASAR TEORI

### A. Sidik Jari

Sidik jari terbentuk karena faktor genetik dan lingkungan. Kode genetik pada DNA memberi perintah untuk terbentuknya janin yang secara spesifik membentuk hasil secara random (acak). Sidik jari memiliki bentuk yang unik bagi setiap orang. Artinya, setiap orang memiliki bentuk sidik jari yang berbeda meskipun terlahir kembar. Jadi, walaupun sidik jari terlihat sama bila dilihat sekilas, bagi penyelidik terlatih atau dengan *software* khusus akan terlihat perbedaannya. Sebelum kita berbicara tentang alat pemindai sidik jari, kita akan berbicara tentang sidik jari tersebut.

#### Pola Sidik Jari

Secara umum, sidik jari dapat dibedakan menjadi beberapa tipe menurut **Henry Classification System**, yaitu *loop pattern*, *whorl pattern*, dan *arch pattern*.



Gambar 1. Pola Sidik Jari (Sumber: <http://alpha07.blogspot.co.id/2012/03/cara-kerja-fingerprint-scanner.html> 8 Mei 2016)

Sekedar informasi, hampir 2/3 manusia memiliki sidik jari dengan tipe *loop pattern*, 1/3 lainnya memiliki tipe *whorl pattern* dan hanya 5-10% yang memiliki tipe *arch pattern*. Lihatlah jari Anda, pola apakah yang terbentuk?

Pola-pola sidik jari seperti inilah yang digunakan untuk membedakan sidik jari secara umum. Namun untuk mesin pembaca sidik jari, pembedaan seperti ini belumlah cukup. Karena itulah mesin pemindai sidik jari dilengkapi dengan metode pengenalan lain yang disebut '*Minutiae*'.

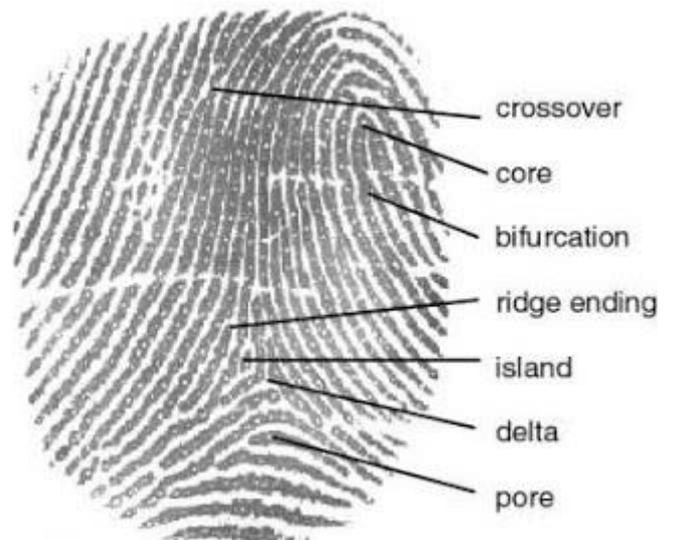
#### Minutiae

Minutiae berasal dari bahasa Inggris yang berarti 'barang yang tidak berarti' atau 'rincian yang tidak penting'. Seperti

artinya, minutiae sebenarnya merupakan rincian sidik jari yang tidak penting bagi kita, tetapi bagi sebuah mesin pemindai sidik jari, itu adalah detil yang sangat diperhatikan.

Untuk lebih jelasnya, *minutiae* pada sidik jari adalah titik-titik yang mengacu pada:

- *crossover* (persilangan dua garis)
- *core* (putar-balikan sebuah garis)
- *bifurcation* (percabangan sebuah garis)
- *ridge ending* (berhentinya sebuah garis)
- *island* (sebuah garis yang sangat pendek)
- *delta* (pertemuan dari tiga buah garis yang membentuk sudut, dan
- *pore* (percabangan sebuah garis yang langsung diikuti dengan menyatunya kembali percabangan tersebut sehingga membentuk sebuah lingkaran kecil).



Gambar 2. *Minutiae* (Sumber: <http://alpha07.blogspot.co.id/2012/03/cara-kerja-fingerprint-scanner.html> 8 Mei 2016)

### B. Fingerprint Scanner

Fingerprint Scanner adalah perangkat otentikasi yang menggunakan identitas biometric dari penggunaannya. Sesuai dengan namanya, fingerprint scanner adalah alat khusus untuk mendeteksi sidik jari penggunaannya. Fingerprint scanner mulai

banyak digunakan secara terintegrasi dengan *notebook* dan beberapa perangkat pendukung lainnya. Keberadaan fingerprint scanner mempermudah prosis login karena penggunaanya tidak perlu mengingat nama user dan password.

### Cara Kerja

Mesin pemindai sidik jari akan mencari titik-titik *minutiae* dan membuat pola dengan menghubungkan-hubungkan titik-titik tersebut. Pola yang didapat dari menghubungkan titik-titik inilah yang nantinya akan digunakan untuk melakukan pencocokan bila ada jari yang dipindai. Jadi, sebenarnya mesin sidik jari tidak mencocokkan gambar, tapi mencocokkan pola yang didapat dari *minutiae-minutiae* ini.

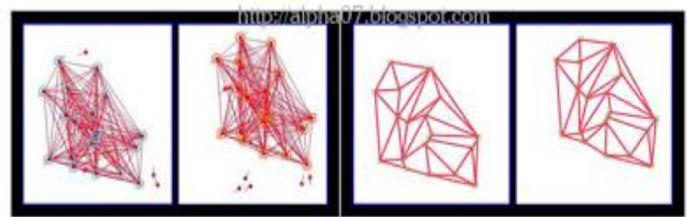
Mesin pemindai sidik jari bekerja dengan mengambil gambar dari sidik jari dan membedakan setiap pola atau alur dari sidik jari tersebut. Sebenarnya banyak cara dapat dilakukan untuk mengambil gambar dari sidik jari tersebut, namun metode umum yang dilakukan adalah dengan menggunakan 2 cara, yaitu dengan sensor optikal dan sensor kapasitansi.

Setelah mesin pemindai menyimpan image atau gambar yang diambil, mesin lalu melakukan '*searching minutiae*' atau mencari titik-titik *minutiae*. Lalu mesin pemindai akan mencari kecocokan pola pada *minutiae-minutiae* yang telah terkumpul tersebut.



Gambar 3. Titik-titik *minutiae* (Sumber: <http://alpha07.blogspot.co.id/2012/03/cara-kerja-fingerprint-scanner.html> 8 Mei 2016)

Jika mesin pemindai sidik jari mendapatkan pola sidik jari yang sama, maka proses identifikasi sudah berhasil. Tidak semua *minutiae* harus digunakan, dan pola yang ditemukan tidak harus sama. Maka dapat disimpulkan bahwa posisi jari kita pada saat identifikasi tidak harus sama persis dengan pada saat pertama kali menyimpan data sidik jari pada mesin tersebut.



Gambar 4. Pola yang terbentuk dari *minutiae* (Sumber: <http://alpha07.blogspot.co.id/2012/03/cara-kerja-fingerprint-scanner.html> 8 Mei 2016)

Pemindai sidik jari optikal dan kapasitif dianggap menghasilkan tingkat keamanan yang tinggi, karena tidak bisa dipalsukan dengan fotocopy sidik jari, sidik jari tiruan, atau bahkan dengan cetak lilin yang mendetil dengan guratan-guratan kontur sidik jari sekalipun.

### C. Pattern Matching Algorithm Boyer-Moore

Algoritma *Boyer Moore* adalah algoritma *pattern matching* yang dilakukan secara mundur pengecekannya yakni dari belakang ke depan. Algoritma ini adalah algoritma yang paling populer untuk pencarian *plain text* dalam teks standar karena kompleksitasnya yang ramping. Sementara, *worst case*-nya algoritma ini berjalan dalam kompleksitas  $O(nm)$ . Algoritma ini membutuhkan *pre-process* kemunculan terakhir dari setiap karakter yang muncul. Ide utama dari algoritma ini adalah, untuk karakter yang tidak muncul, maka pencocokan string tidak perlu dilakukan.

### Cara Kerja

Algoritma Boyer-Moore memiliki 4 kasus dengan penanganan berbeda :

- a. Teks[i] dan pattern[j] match, pengecekan mundur ke  $i - 1$  dan  $j - 1$ . Dan apabila  $j < 0$  maka pencocokan valid.
- b. Teks[i] dan pattern[j] tidak match dan terdapat Teks[i] misal x yang merupakan kemunculan terakhir dari pattern pada posisi  $< j$ , geser pattern sampai Teks[i] = x.
- c. Teks[i] dan pattern[j] tidak match dan terdapat Teks[i] misal x yang merupakan kemunculan terakhir dari pattern pada posisi  $> j$ , geser pattern sampai satu kali.

d. Teks[i] dan pattern[j] tidak match dan teks[i] tidak muncul pada pattern, i maju sejauh nilai j+1 saat ini.

Berikut adalah alur cara kerja algoritma *Boyer Moore*.

TEKS : AKU SUKA NANAS

PATTERN : NAN

NAN

NAN

NAN

*Boyer Moore* berjalan pada kasus terburuk ketika :

Teks = AAA..AAA

Pattern = BAA..A

Dalam kasus ini *Boyer-Moore* berjalan dalam kompleksitas  $O(nm)$ . *Boyer Moore* tidak terlalu baik dalam pencocokan string yang teks dan *pattern*-nya memiliki struktur yang mirip. Namun algoritma ini dapat berjalan hingga  $O(n/m)$  dalam *best case*. Algoritma *Boyer Moore* paling umum digunakan dalam *string matching* pada teks yang dapat dibaca manusia

### Pseudocode

```
public static int bmMatch(String text, String
pattern) {
    int last[] = buildLast(pattern);
    int n = text.length();
    int m = pattern.length();
    int i = m-1;

    if (i > n-1)
        return -1; // no match if pattern is
                // longer than text
    int j = m-1;
    do {
        if (pattern.charAt(j) == text.charAt(i))
            if (j == 0)
                return i; // match
            else { // looking-glass technique
                i--;
                j--;
            }
        else { // character jump technique
            int lo = last[text.charAt(i)];
```

```
//last occ
        i = i + m - Math.min(j, 1+lo);
        j = m - 1;
    }
} while (i <= n-1);

return -1; // no match
} // end of bmMatch()

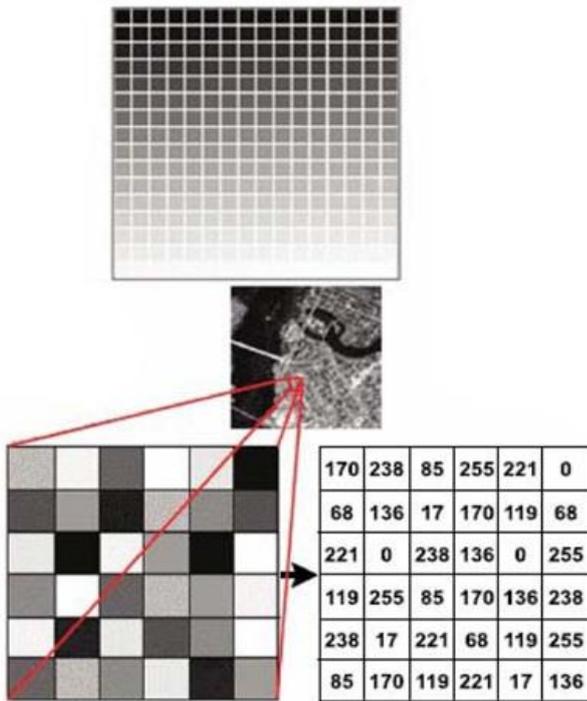
public static int[] buildLast(String pattern)
/* Return array storing index of last
occurrence of each ASCII char in pattern. */
{
    int last[] = new int[128]; // ASCII char set
    for(int i=0; i < 128; i++)
        last[i] = -1; // initialize array
    for(int i=0; i<pattern.length(); i++)
        last[pattern.charAt(i)] = i;
    return last;
} // end of buildLast()
```

### D. Citra dan Konversi dalam Pixel dan Bit

Data citra dikonversi dalam bentuk format digital mentah yang merupakan sekumpulan data numerik. Unit terkecil dari data digital adalah bit, yaitu angka biner 0 atau 1. Kumpulan dari data sejumlah 8 bit data adalah sebuah unit data yang disebut byte, dengan nilai 0 - 255. Dalam hal citra digital level energi dituliskan dalam satuan byte. Kumpulan byte ini dengan struktur tertentu bisa dibaca oleh software dan disebut citra digital 8-bit.

Pixel (picture element) adalah sebuah titik yang merupakan elemen paling kecil pada citra. Angka numerik (1 byte) dari pixel disebut digital number (DN). DN bisa ditampilkan dalam warna kelabu, berkisar antara putih dan hitam (gray scale), tergantung level energi yang terdeteksi. Pixel yang disusun dalam order yang benar akan membentuk sebuah citra. Kebanyakan citra yang belum diproses disimpan dalam bentuk gray scale, yang merupakan skala warna dari hitam ke putih dengan derajat keabuan yang bervariasi. Untuk Penginderaan Jauh, skala yang dipakai adalah 255 shade gray scale, dimana nilai 0 menggambarkan hitam, nilai 255 putih. Dua gambar di bawah ini menunjukkan derajat keabuan dan

hubungan antara DN dan derajat keabuan yang menyusun sebuah citra.



Gambar 5. Pola Sidik Jari (Sumber: <http://pengertian-definisi.blogspot.co.id/2011/09/analisis-data-citra-penginderaan-jauh.html> 8 Mei 2016)

Untuk citra multispectral, masing-masing pixel mempunyai beberapa DN, sesuai dengan jumlah band yang dimiliki. Sebagai contoh untuk Landsat 7, masing-masing pixel mempunyai 7 DN dari 7 band yang dimiliki. Citra bisa ditampilkan untuk masing-masing band dalam bentuk hitam dan putih maupun kombinasi 3 band sekaligus yang disebut color composite

### III. PEMBAHASAN

Pada bab ini penulis akan membahas bagaimana algoritma Boyer-Moore untuk patter dua dimensi dan juga aplikasinya pada pencocokan *minutiae*.

#### A. Algorithm Pattern Matching Dua Dimensi

Idenya adalah melakukan proses Boyer-Moore mulai dari indeks pattern (i,j) terbesar (pojok kanan bawah) ke indeks sebelumnya dan berakhir pada indeks (0,0) pada pattern. Proses mundur indeks dilakukan dari sumbu x (indeks j) terlebih dahulu.

Prekondisi sama seperti Boyer-Moore menghitung posisi paling terakhir dari suatu character. Namun kita melakukan prekondisi ini pada setiap baris, misal pattern berukuran 4x4

maka akan ada 4 hasil perhitungan prekondisi ini yang independen.

Algoritma ini memiliki 4 kasus dengan penanganan berbeda :

- Teks[i][j] dan pattern[k][l] match, pengecekan mundur ke l - 1 dan j - 1, apa bila l < 0 maka pengecekan mundur ke i - 1, k - 1, j = j + ukuran pattern, l = ukuran pattern - 1. Dan apabila k < 0 dan l < 0 maka pencocokan valid.
- Teks[i][j] dan pattern[k][l] tidak match dan terdapat Teks[i][j] misal x yang merupakan kemunculan terakhir dari pattern[k] pada posisi < l, geser pattern sampai Teks[i][j] = x, posisi tersebut.
- Teks[i][j] dan pattern[k][l] tidak match dan terdapat Teks[i][j] misal x yang merupakan kemunculan terakhir dari pattern[k] pada posisi < l, geser pattern satu kali searah sumbu x.
- Teks[i][j] dan pattern[k][l] tidak match dan teks[i][j] tidak muncul pada pattern[k], pattern maju sejauh nilai l+1 saat ini.
- Apabila j > ukuran kolom teks maka pencocokan dimulai dari j = panjang pattern-1 dengan i+1.

#### Pseudocode

Asumsi pattern persegi, text persegi, bentuk pattern dan text adalah array of string dan text selalu lebih besar dari pattern :

```
public static int[][] buildLast(String[]
pattern)
/* Return array storing index of last
occurrence of each char in pattern. */
{
    int last[][] = new int[pattern.length][256];
    // indeks baris menggambarkan baris ke x
    pattern, kolom menggambarkan posisi tiap char
    for(int i=0; i < pattern.length; i++)
        for(int j=0; j < 256; j++)
            last[i][j] = -1; // initialize array
    for(int j=0; j < pattern[0].length(); j++)
```

```

        last[i][pattern[i].charAt(j)] = j;
    return last;
} // end of buildLast()
public static int bmMatch(String[] text,
String[] pattern) {
    int last[][] = buildLast(pattern);
    int n = text.length();
    int m = pattern.length();
    int i = m-1; //indeks pattern
    int k,l = m-1; //indeks teks
    int counternaik = 0;
    if (i > n-1) || (j > n-1)
        return -1,-1; // no match if pattern is
        // longer than text
    int j = m-1; //indeks pattern
    do {
        if (pattern[i].charAt(j) ==
text[k].charAt(l))
            if (j == 0) && (i == 0)
                return i,j; // match
            else if (j==0) // case a
                j = m-1;
                l = l+ m-1;
                i--;
                counternaik++;
                k = k-1;
            else { // case a
                l--;
                j--;
            }
        else { // character jump technique b,c,d
            int lo = last[i][text[i].charAt(j)];
//last occ
            l = l + m - Math.min(j, 1+lo);
            j = m - 1;
            i = m - 1;
            if (l >= n) //case e
                l = m - 1;
                k = k+counternaik+1;
        }
    } while (i <= n-1);

    return -1,-1; // no match
} // end of bmMatch()

```

## B. Implementasi dalam Fingerprint Scanner

Seperti yang kita tahu dari dasar teori diatas, fingerprint scanner akan menghasilkan citra pola sidik jari pengguna. Citra tersebut dapat di “digitalisasi” menjad kumpulan bit-bit yang menggambarkan tingkat kehitaman yang kemudian dilakukan pengelompokan tiap 8 bit menjadi 1 byte dan menggambarkan tingkat keabuan. Kemudian 1 pixel citra mengandung beberapa byte tadi.

Pada bahasan ini penulis mengasumsikan pattern pasti dalam bentuk persegi karena pixel selalu persegi dan satu pixel hanya terdiri dari 4 byte (untuk memudahkan simulasi dan pemahaman).

Prekondisinya ialah fingerprint scanner sudah memiliki dulu data pola tiap *minutiae* dalam bentuk pixel. Kemudian pattern pola tiap *minutiae* dicocokkan hasil citra pola sidik jari pengguna agar dapat ditemukan titik-titik *minutiae* pada sidik jari tersebut. Semua byte dalam satu pixel dan citra diubah dulu ke character.

Berikut contoh skema proses scanning sidik jari dengan sebuah pola *minutiae* :

Misalkan pola minutiae crossover memiliki pattern pada satu pixel :

```

A b
C d

```

Dan pola sidik jari hasil pemindaian (setelah diubah ke byte) adalah :

```

D = ~ 1 B `
3 A b C d /
? C d w w q
Q w e r t y
Skema :

```

```

D = ~ 1 B `
3 A b C d /
? C d w w q
Q w e r t y
D = ~ 1 B `
3 A b C d /
? C d w w q
Q w e r t y

```

Keterangan :

Hijau = bagian yang sedang dicocokkan

Merah = posisi terjadi kesalahan

Biru = bagian yang sesuai dengan pattern

D = ~ 1 B `   
 3 A b C d /   
 ? C d w w q   
 Q w e r t y

D = ~ 1 B `   
 3 A b C d /   
 ? C d w w q   
 Q w e r t y

*Minutiae crossover* ditemukan di indeks 1,1

Lakukan pencocokkan tiap tipe minutiae hingga akhir citra sidik jari (sangat dimungkinkan ditemukan lebih dari satu titik). Lalu lakukan hal yang sama pada tipe minutiae yang lain maka semua titik minutiae akan didapatkan dan pola juga sudah dapat dibentuk dan disimpan.

Untuk proses otentifikasi pengguna metoda yang dilakukan sama seperti hal diatas namun pattern yang digunakan bukanlah data pola tiap tipe *minutiae* melainkan data tipe *minutiae* yang ada pada id pengguna, hal ini dilakukan untuk efektifitas program (tidak melakukan pencarian terhadap pattern minutiae yang tidak ada)

#### IV. KESIMPULAN

Kesimpulan dari makalah ini adalah:

1. Algoritma pattern matching bisa diterapkan pada citra digital, yang dalam hal ini berarti pada bidang dua dimensi.
2. Boyer-Moore merupakan algoritma yang cocok digunakan karena domain dari pattern ini sangat banyak yakni 256 karakter (1 byte) dan variansinya cukup tinggi mengingat yang diproses adalah citra.
3. Dalam makalah ini belum dibahas mengenai penentuan galat atau toleransi citra yang berbeda maka, masih perlu dikembangkan lagi penerapan pattern matching dengan algoritma lain agar kebutuhan galat ini dapat tercakupi.

#### V. KATA PENUTUP

Pertama, penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga penulis

dapat menyelesaikan tugas makalah IF2211 Strategi Algoritma ini. Penulis juga ingin berterima kasih kepada Dosen IF 2211 Tn. Rinaldi Munir dan Ny. Nur Ulfa Maulidevi yang sudah mengajarkan setiap materi dengan jelas sehingga penulis dapat menyelesaikan tugas tanpa kesulitan berarti. Terima kasih juga pada teman-teman yang telah memberikan masukan pada makalah ini.

Semoga ke depannya teori yang saya bahas di tugas ini dapat diimplementasikan atau bahkan menjadi pemicu munculnya algoritma baru yang lebih mangkus pada praktik yang membutuhkan pattern matching.

#### REFERENSI

- [1] "Analisis Data dan Citra Penginderaan Jauh". Irwanto Forester. Diakses tanggal 9 September 2011
- [2] Aldrin Svaifullah. (2011). Tabloid Pulsa Edisi 218 Th IX / 2011 / 5
- [3] Hueske. Edward. Firearms and Fingerprints. Facts on File/Infobase Publishing, New York. 2009. ISBN 978-0-8160-5512-8
- [4] *Henry. Edward R.. Sir (1900). "Classification and Uses of Finger Prints" (PDF). London: George Rutledge & Sons, Ltd.*
- [5] *Wang. Yongchang; O. Hao; A. Fatehmuria; D. I. Lau; L. G. Hassebrook (2009). "Data Acquisition and Quality Analysis of 3-Dimensional Fingerprints" (PDF). Florida: IEEE conference on Biometrics, Identity and Security. Retrieved March 2010.*
- [6] Munir, Rinaldi. 2016. Diktat Kuliah Strategi Algoritma. Bandung : Program Studi Teknik Informatika Institut Teknologi Bandung.

#### PERNYATAAN

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

Bandung, 9 Mei 2016



Pradiya Raudi Avinanto 13514087