

Pencarian Potongan Gambar Menggunakan Algoritma Boyer Moore

Andrian Octavianus-13512602
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512602@std.stei.itb.ac.id

Abstract— Pada pencarian gambar di internet, pada saat ini masih banyak yang menggunakan judul gambar sebagai kunci pencarian gambar tersebut. Namun saat ini sudah ada pencarian yang dapat dilakukan dengan menggunakan gambar sebagai kuncinya. Dengan menggunakan pencarian gambar menggunakan gambar ini akan dapat mempermudah seseorang untuk mencari apakah gambar yang di miliki oleh pengguna internet tersebut sudah tersebar di internet atau tidak. Pencarian gambar ini dapat dilakukan dengan menggunakan algoritma pattern matching Boyer Moore dan matrik color image pada piksel yang digunakan untuk menentukan apakah potongan gambar tersebut terdapat pada gambar tersebut atau tidak.

Index Terms—Pencarian Gambar, Boyer Moore, Color image

I. PENDAHULUAN

Pencarian gambar pada computer saat ini masih banyak yang hanya menggunakan judul gambar sebagai kunci pencarian. Untuk melakukan pengecekan apakah ada gambar yang sama dalam komputer tapi judul gambar tersebut beda. Hal tersebut dapat dilakukan dengan cara manual, yaitu dengan membandingkan gambar dengan mata. Hal ini tentu saja akan menyusahkan dalam hal pencarian dan pencocokan gambar tersebut sendiri. padahal gambar yang tersebar di internet saat ini sudah sangat banyak. Bahkan untuk jumlah gambar yang ada di internet saat ini sudah dapat dikatakan sebagai big data. Hal ini disebabkan oleh pengguna internet dapat dengan bebas mengunggah gambar yang mereka miliki ke dalam internet.

Dengan banyaknya gambar yang ada di internet pengguna internet tentu juga dapat mengunduh gambar yang ada di internet. Namun saat ini sudah ada search engine yang sudah dapat mencari gambar dengan menggunakan gambar sebagai kunci pencarian. Search engine tersebut melakukan pencarian berdasarkan tingkat kemiripan pada gambar yang menjadi kunci pencarian atau gambar yang sedang dicari. Sehingga hasil yang diberikan pada search engine tersebut mengembalikan gambar yang memiliki tingkat kemiripan yang hamper sama. Hal ini tentu saja dapat memudahkan pengguna internet untuk mengetahui apakah gambar yang mereka

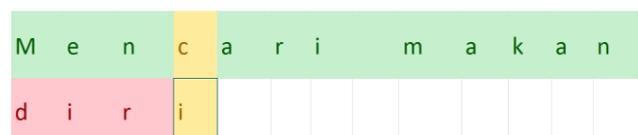
miliki banyak tersebar di internet atau tidak. Dengan banyaknya para plagiarism yang mengambil gambar atau foto kita tanpa izin akan dapat dengan mudah diketahui dengan menggunakan fasilitas search gambar ini.

Pada makalah ini akan dijelaskan pembuatan pencarian gambar yang sama dengan menggunakan algoritma Boyer Moore.

II. DASAR TEORI

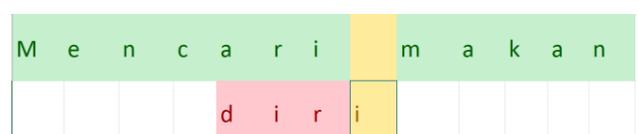
1. Algoritma Boyer Moore

Algoritma Boyer-Moore adalah salah satu algoritma untuk mencari suatu pattern di dalam teks, dibuat oleh R.M Boyer dan J.S Moore. Ide utama algoritma ini adalah mencari pattern dengan melakukan perbandingan karakter mulai dari karakter paling kanan dari pattern yang dicari. Dengan menggunakan algoritma ini, secara rata-rata proses pencarian akan menjadi lebih cepat jika dibandingkan dengan algoritma lainnya. alasan melakukan pencocokan dari kanan (posisi terakhir pattern yang dicari). Cara kerja dari algoritma Boyer Moore adalah sebagai berikut :



Gambar 1

Pada gambar diatas menunjukkan bahwa pencarian pada text “Mencari makan” dengan pattern “diri” dimulai dengan membandingkan huruf yang paling kanan pada pattern tersebut yaitu huruf “c” dengan huruf “i”. Pada pencarian tersebut huruf “c” pada teks tidak ada pada pattern. Sehingga pattern harus bergeser sebanyak jumlah pattern tersebut. Sehingga posisi dari pencarian pattern tersebut menjadi seperti ini :



Gambar 1b

Pada contoh diatas menunjukkan bahwa boyer moore melakukan loncatan yang besar pada pencarian teks tersebut. Sehingga pada pencarian boyer moore akan menjadi lebih cepat dibandingkan dengan algoritma pencarian biasa yang dilakukan dengan menggunakan algoritma brute force.

Langkah-langkah dalam algoritma boyer moore ini adalah sebagai berikut:

- Membuat table indeks terakhir pada pattern yang akan digunakan untuk menentukan jumlah pergeseran pencarian kalimat. Pergeseran tersebut akan ditentukan dengan melihat apakah indeks yang dicari pada teks tersebut nilai nya ada pada table indeks terakhir atau tidak.
- Apabila teks yang dicari tersebut tidak ada dalam pattern maka akan dilakukan pergeseran sebanyak jumlah pattern tersebut.
- Apabila indeks pencarian yang tidak sama menunjukkan ada dalam table indeks terakhir pattern dan nilai indeks yang sedang dicari lebih besar dengan index terakhir maka akan dilaakukan pergeseran pada pencarian tersebut sebanyak indeks terakhir yang terdapat dalam pattern.
- Apabila indeks pencarian yang tidak sama menunjukkan ada dalam table indeks terakhir pattern dan nilai indeks yang sedang dicari lebih kecil dengan index terakhir maka akan dilaakukan pergeseran pada pencarian tersebut sebanyak indeks terakhir yang terdapat dalam pattern.
- Apabila pencarian dalam indeks tersebut sama dengan pattern yang paling terakhir maka akan dilakukan pergeseran sebanyak 1 pergeseran ke kanan.

Pembentukan table indeks terakhir dalam pencarian booyer moore dapat dilakukan dengan cara apapun. Dalam pembentukan table ini dapat dilakukan dengan cara melakukan iterasi terus menerus sampai pada akhir pattern dapat dilakukan untk menentukan table indeks terakhir ini. Selain dengan cara iterasi sampai ke belakang dapat juga dilakukan dengan cara menggunakan kode ASII sebagai pembentukan table ini. Namun apabila dengan pembentukan kode ASCII ini hanya dapat digunakan pada kalimat yang tidak mempunyai symbol-simbol yang tidak memenuhi kode ASCII.

| | | | | |
|--------|---|---|---|---|
| HURUF | d | i | r | i |
| Indeks | 1 | 4 | 2 | 4 |

Gambar 1c

Pada gambar di atas menunjukkan bahwa indeks yang

dicatat apabila terdapat huruf yang sama dalam pattern tersebut nilai indeks yang diambil adalah nilai indeks yang terakhir.dalam gambar di atas huruf "i" terdapat 2 huruf dalam pattern tersebut yaitu terdapat pada indeks 2 dan 4, sehingga huruf "i" tersebut indeks yang diambil adalah 4.

Algoritma dari Boyer moore adalah sebagai berikut :

Algoritm BMMatch(T, P):

Input: String T (text with n characters and P (pattern) with m characters

Output: Starting index offirst substring of T matching p, or an indication that P is not a substring of T

i <-- m - 1

j <-- m - 1

repeat

if P[j] = T[i] **then**

if j = 0 **then**

return i {a match!}

else {check next character}

i <-- i - 1

j <-- j - 1

else { P[j] <> T[i] move the pattern}

i <-- i + m - j - 1 {reset i to where it began in most-recent test}

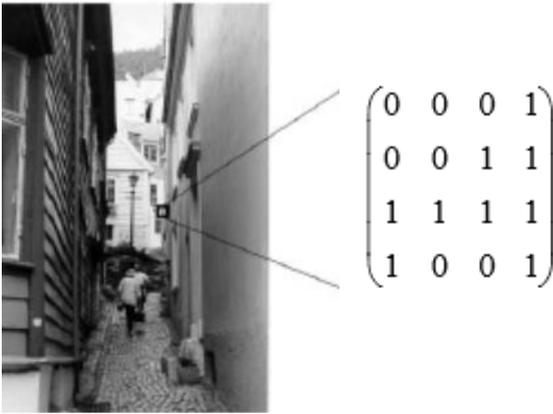
i <-- i + max(j - last(T[i]), match(j)) {shift P relative to T}
 {note that even if j-last(T[i]) is negative, we will still perform apositive shift, because match (j) is always at least 1.}

j <-- m-1

until i > n - 1

2. Color Image

Binary image adalah sebuah digital image pertama yang hanya mempunyai dua nilai dalam sebuah piksel. Dalam binary image biasanya warna yang digunakan adalah hitam dan putih dalam sebuah piksel. Warna hitam dan putih tersebut direpresentasikan dalam bilangan biner yaitu 0 dan 1. Dimana 0 menunjukkan warna putih dan 1 menunjukkan warna hitam. Sehingga pada sebuah piksel akan terdapat matric yang hanya berisi nilai 1 dan 0.Warna hitam dan putih tersebut digunakan untk menunjukkan warna dari background colour yang jika dikumpulkan dalam banyak matrik akan dapat membentuk sebuah gambar.

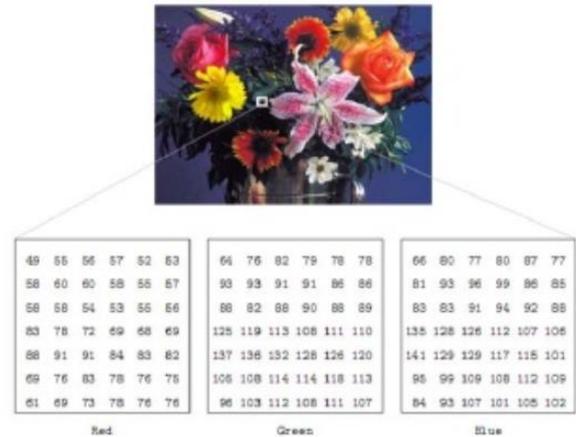


Gambar 1d

Gambar diatas adalah contoh dari matrik yang berukuran 4x4 yang terdapat dalam sebuah piksel dalam sebuah gambar monochrome. Sehingga gambar-gambar tersebut sebenarnya adalah terdiri dari matrik yang berisi matrik binary image dan membentuk sebuah gambar.

Binary image ini dapat disimpan sebagai gambar yang bertipe bitmap. File bitmap ini dapat disimpan dalam resolusi 640x480. Hal ini menunjukkan bahwa gambar bitmap ini mempunyai matrik of binary image sebanyak 640 x 480.

Namun dengan seiring perkembangan kualitas gambar saat ini matrik image ini dapat disimpan tidak hanya dalam bilangan biner saja. namun sudah dapat diisi dengan bilangan 0-255 yang hal ini terdapat pada warna RGB. Pada gambar RGB ini terdapat 3 matrik yang berbeda dalam setiap pikselnya. Hal ini digunakan untuk menentukan warna apa yang terdapat pada piksel tersebut dengan semua kombinasi dari warna merah, hijau dan biru ini. Sehingga untuk membuat warna ungu akan membutuhkan perbandingan warna merah dan biru untuk dapat menghasilkan warna ungu tersebut. Hal ini menyebabkan warna yang ditimbulkan tiak hanya berwarna hitam dan putih namun dapat juga berwarna merah, hijau, kuning dan sebagainya. Dengan menggunakan teknologi ini maka gambar yang dihasilkan dapat berwarna warni seperti gambar yang sudah mendekati nyata. Untuk format penyimpanan gambar biasanya disimpan dalam bentuk .JPG,PNG dll. Contoh matrik piksel ada gambar dapat ditunjukkan pada gambar di bawah ini.



Gambar 1e

Pada gambar diatas menunjukkan untuk setiap piksel sudah tidak hanya mempunyai nilai 1 bit yaitu 0 dan 1. Namun pada gambar diatas memiliki matrik dengan untuk setiap isi nya 0-255 yang bertipe RGB.

III. PENERAPAN ALGORITMA BOYER MOORE

A. Penerapan algoritma boyer moore

Pada matrik-matrik dari setiap piksel gambar tersebut akan dilakukan perubahan yaitu akan dilakukan dalam bentuk list linear yang akan memudahkan operasi boyer moore ini.

| | | | | |
|-----|-----|----|----|----|
| 22 | 33 | 44 | 55 | 11 |
| 44 | 22 | 11 | 23 | 55 |
| 123 | 215 | 29 | 1 | 12 |
| 16 | 28 | 32 | 12 | 15 |

Gambar 2a

Pada gambar diatas yang mempunyai matrik 4 x 4 ini akan diubah menjadi list linear yaitu 22,33,44,55,11,44,22,11,23,55,123,215,29,1,12,16,28,32, 12,15. Dengan mengubah menjadi list linear tersebut akan memudahkan dalam melakukan pembentukan indeks terbesar dalam sebuah pattern. Karena indeks yang dimiliki hanya mempunyai 1 indeks linear. Apabila tidak diubah dalam indeks linear dan masih menggunakan indeks matrik hal ini akan membuat pembentukan indeks table terbesar akan menjadi sulit.

Untuk pencarian potongan gambar ini akan dilakukan pencocokan pada sebuah matrik piksel. Yang akan dilakukan pengecekan secara traversal. Sehingga apabila terdapat satu angka saja yang tidak sama di awal maka hanya akan langsung di anggap tidak sama pada piksel tersebut. Dengan menggunakan piksel ini dapat dilakukan pencarian potongan gambar pada gambar yang lebih besar dengan melakukan pencarian gambar terhadap matrik piksel yang akan dicari.

Algoritma untuk melakukan perbandingan piksel ini adalah sebagai berikut :

Algoritma isPikselSama(P1, P2):

P1 adalah sebuah list angka yang merepresentasikan warna piksel 1

P2 adalah sebuah list angka yang merepresentasikan warna piksel 2

```
i=0
for (i < Panjang P2)
    if P1.at(i) != P2.at(i)
        then return false
return true
```

Pada proses pertama akan dilakukan pembuatan struktur list linear dari setiap matrik yang ada dalam piksel pada gambar. Karena gambar yang akan digunakan dalam RGB maka akan terdapat 3 matrik yang berbeda. Untuk dapat menggabungkan ketiga matrik ini akan dilakukan perkalian matrik dari ketiga matrik tersebut sehingga akan menghasilkan sebuah matrik yang merupakan hasil dari perkalian matrik-matrik tersebut. Perbandingan akan dilakukan untuk setiap piksel yang ada dalam gambar pencarian dengan piksel yang ada dalam gambar yang sedang diperbandingkan.



Pada proses pencocokan pertama akan dilakukan pembuatan table indeks terakhir dari setiap piksel yang terdapat dalam gambar tersebut. Hal ini digunakan untuk memudahkan untuk menentukan seberapa banyak akan dilakukan pergeseran piksel akan dilakukan. Sehingga pada proses perbandingan gambar tersebut hanya akan melakukan pergeseran piksel-piksel tersebut. Berikut adalah gambar dari table indeks piksel:

| 1 | 2 |
|---|---|
| 22,33,44,55,11,44,22,11,23,55,44,66,99,77 | 77,99,55,55,11,33,22,66,23,57,44,66,99,77 |

Gambar 2b

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 2 | 5 | 7 |

Gambar 2c

Dalam pembuatan table indeks tertinggi ini sudah tidak dapat menggunakan kode ASCII untuk membangun table ini. Untuk membuat table indeks tertinggi ini diperlukan proses iterasi untuk melihat piksel mempunyai indeks tertinggi pada posisi berapa. Proses pencocokan piksel ini dilakukan menggunakan fungsi isPiksel sama yang sudah dibuat sebelumnya. Untuk isi dari setiap matrik piksel akan disimpan dalam bentuk indeks 1,2,3 dan seterusnya. Dengan setiap indeks ini menunjukkan isi dari indeks tersebut dan isi dari matrik tersebut tentunya sudah tidak dalam bentuk matrik lagi namun sudah dalam bentuk list linier. Hal ini bertujuan untuk mempercepat proses pencarian dalam table indeks tertinggi dalam table tersebut. Sehingga hanya perlu melihat indeks piksel ke berapa pada table indeks tertinggi seperti pada contoh gambar 2c.

Dalam penerapan algoritma boyer moore ini tidak lagi menggunakan string sebagai pembandingnya. Namun langsung menggunakan piksel yang terdapat dalam fungsi yang sudah dibuat sebelumnya. Namun untuk proses pencarian potongan gambar ini tetap saja dilakukan dari piksel yang paling terakhir pada gambar pencarian. Sehingga apabila dalam piksel terakhir tersebut sudah tidak sama maka matrik piksel tersebut juga akan bergeser sebanyak piksel yang sudah dibuat dalam algoritma boyer moore tersebut dengan melihat table indeks tertinggi.

Algoritma boyer moore yang digunakan dalam pencocokan gambar ini sebagai berikut:

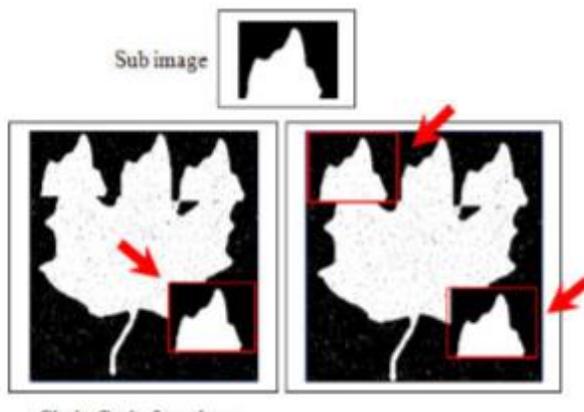
```
int cekPiksel(Piksel testCase, Piksel pattern)
{
    listIndexPattern[] = indexOfPiksel(pattern);
    skip;
    for(i=0; i <= testCase.length() - pattern.length(); i++)
    {
        skip=0;
        for(j=pattern.length()-1; j >= 0; j--)
        {
            if(pattern.pikselAt(j) != testCase.pikselAt(i+j))
            {
                skip = max(1, j - listIndexPattern[pattern.pikselAt(i+j)]);
                break;
            }
        }
        if(skip == 0)
        {
            return i; //mengembalikan posisi array
        }
    }
    return -1; //mengembalikan nilai -1 apabila
```

tidak ditemukan pattern yang sama
}

B. Hasil Penerapan Algoritma Boyer Moore

Hasil yang diberikan pada penerapan algoritma boyer moore ini akan menunjukkan pada potongan mana mana saja gambar yang ditemukan dengan menggunakan piksel sebagai pembanding nya. Hasil potongan yang diberikan adalah hasil yang memang sama persis dengan potongan gambar yang sedang dicari. Apabila dalam matriks tersebut terdapat angka yang tidak sama hanya 1 saja akan tetap saja dianggap tidak sama.

Pada proses pencarian dengan menggunakan algoritma ini untuk setiap hasil yang ditemukan benar-benar sama maka proses tidak akan diberhentikan. Namun proses perbandingan akan dilanjutkan untuk mencari potongan gambar yang lain dalam sebuah gambar tersebut. Untuk dapat menyimpan potongan gambar yang sudah ditemukan tadi maka posisi piksel tersebut akan disimpan dalam sebuah queue atau list yang dapat menampung seluruh hasil pencarian tersebut. Pada hasil akhir pencarian dan sudah ditemukan semua pada penerapan ini akan dapat menampilkan semua posisi potongan gambar dari list atau queue yang sudah dibuat sebelumnya



Gambar 2d

Pada gambar diatas adalah hasil dari pencarian potongan daun yang terdapat pada gambar daun yang lebih besar. Pada hasil pencarian diatas ditemukan 2 potongan gambar yang sama persis. Sehingga gambar tersebut dianggap mempunyai potongan gambar yang sama dengan gambar potongan yang sedang dicari. Hal ini dikarenakan gambar yang sedang dicari nilai semua piksel benar-benar sama semua.

Hasil pencarian dari penerapan algoritma ini juga hanya dapat mendapatkan potongan gambar yang ukuran matrik pikselnya sama. Contohnya ukuran potongan gambar 200x200, maka potongan gambar yang akan didapatkan juga akan berukuran 200x200. Hal ini di sebabkan oleh pencarian dilakukan berdasarkan jumlah piksel yang terdapat pada potongan gambar. Apabila terdapat potongan gambar yang sama namun ukuran piksel yang

terdapat pada gambar tersebut berbeda, maka pada penerapan algoritma ini juga tidak akan menganggap bahwa itu adalah sudatu gambar yang sama.

IV. ANALISIS

Pencarian potongan gambar dengan menggunakan algoritma boyer moore yng berdasarkan pada perbandingan piksel ini tidak melakukan perbandingan piksel secara keseluruhan. Namun apabila piksel yang terakhir sudah salah maka akan dilanjutkan piksel yang selanjutnya dengan melakukan pergeseran piksel sebanyak yang diperlukan. Berbeda dengan algoritma brute force yang dilakukan dari awal piksel. Apabila terdapat piksel yang berbeda maka akan dilakukan penggeseran sebanyak satu kali. Hal ini tentunya akan membuat jumlah perbandingan yang cukup banyak yaitu dengan jumlah terbanyak n^2 dengan n adalah jumlah piksel yang ada dalam potongan gambar tersebut.

Berbeda dengan algoritma boyer moore yang melakukan pecarian dari belakang. Apabila terdapat piksel yang tidak sama maka akan melakukan pergeseran yang berdasar dari table indeks tertinggi. Hal ini tentunya akan banyak mengurangi jumlah perbandingan pada piksel di gambar. Namun pada algoritma ini tidak dapat dilakukan untuk pencarian yang memiliki tingkat kesamaan yang masih berbeda. Hasil yang diberikan pada penerapan algoritma ini harus sama. Apabila tidak sama walaupun sedikit akan dianggap tidak sama.

V. KESIMPULAN

Pencarian potongan gambar dengan menggunakan algoritma boyer moore hanya dapat digunakan pada exact searching yaitu gambar hanya dapat dikatakan sama apabila memang tidak ada yang sama sedikitpun. Namun dengan menggunakan algoritma Boyer moore ini akan dapat mengurangi jumlah perbandingan piksel yang dilakukan. Hal itu akan berdampak pada waktu proses pencocokan gambar. Karena jumlah perbandingan gambar lebih sedikit maka waktu yang akan digunakan untuk membandingkan 2 buah gambar akan juga lebih cepat.

Untuk dapat melakukan pencarian potongan gambar yang tidak exact searching maka menggunakan algoritma bruteforce dan menggunakan tingkat kesamaan. Dengan menggunakan algoritma brute force maka akan dapat mencari angka yang terdapat dalam piksel yang nilainya sama. Setiap nilai yang sama tersebut dilakukan penambahan tingkat kemiripan.

REFERENCES

- [1] http://www.fastgraph.com/help/monochrome_bitmaps.html diakses 11.39 19-12-2013
- [2] Rafael C. Gonzalez and Richard E. Woods, 2008, Digital Image Processing, prentice Hall, 3rd edition.
- [3] Sleit, A., AlMobaideen, W., Qatawneh, M., Saadeh, H., 2008, Efficient Processing for Binary Submatrix Matching, American Journal of Applied Sciences, Vol. 6, No.1, pp. 78-88.

- [4] Bronson, R., Schaum, S., 1989, Outline of Theory and Problems of Matrix Operations, McGraw Hill.
- [5] Munir, Rinaldi, "Strategi Algoritmik", Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2007
- [6] http://www.academia.edu/5026976/Klasifikasi_Warna_Menggunakan_Pengolahan_Model_Warna_HSV di akses 11.39 19-12-2013

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, 19 Desember 2013

ttd



Andrian Octavianus (13512602)