

Penggunaan Algoritma Brute Force dan KMP dalam Pencocokan Dua Gambar

Yusuf Ardi Nugroho - 13510002¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13510002@std.stei.itb.ac.id

Abstract—Saat ini algoritma *string matching* telah banyak berkembang, Pengembangan algoritma *string matching* ini sendiri sering digunakan untuk berbagai macam hal. Salah satu contohnya adalah *pattern matching*. Penggunaan *pattern matching* terutama pada gambar adalah pada fitur *search* dengan menggunakan gambar pada *google image*. Ide dasar dari aplikasi tersebut adalah mencocokkan pola 2 buah gambar. Cara ini bisa digunakan dengan mencocokkan pixel atau *data point*.

Pada Makalah kali ini hanya akan dijelaskan contoh penerapan algoritma *Brute Force* dan *KMP* dalam pencocokan dua buah gambar dengan menggunakan pixel, serta perbandingan kompleksitas yang terjadi bila menggunakan kedua algoritma tersebut.

Index Terms—*Image, Pixel, String Matching, Pattern, Knuth-Morris-Pratt, Brute Force.*

I. DASAR TEORI

1. *String Matching*

String matching merupakan algoritma pencarian yang sangat penting di dalam dunia pendidikan informatika. *String matching* digunakan pada algoritma pemrosesan teks. Algoritma ini merupakan dasar dari berbagai aplikasi yang banyak digunakan saat ini. Salah satu contoh aplikasi yang menggunakan *string matching* sebagai dasar algoritmanya adalah Google.

Contoh lain yang sering kita gunakan adalah pada fitur Find dan Find and Replace pada notepad atau aplikasi pemrosesan teks lainnya. Secara umum *string matching* memudahkan kita dalam mencari suatu bagian dari teks yang cukup besar hingga pada akhirnya mengefisiensikan waktu kita untuk mencari.

Pada makalah ini, hal yang akan dibahas akan hanya akan menekankan pada jenis penggunaan lain dari algoritma *string matching*, yaitu pencocokan dua buah gambar.

Pada pembahasan tentang pencocokan dua buah gambar jenis objek yang diproses berbeda dari pencocokan *string* teks biasa. Jenis objek yang digunakan adalah gambar. Pada sebuah gambar tidak terdapat teks atau karakter eksplisit yang bisa langsung diproses menggunakan algoritma *string matching* tanpa memproses elemen-elemen penyusun gambar tersebut terlebih dahulu. Pixel adalah elemen penyusun dari sebuah gambar yang terdiri dari kode bit.

Alternatif cara untuk memperoleh “*string*” untuk dicocokkan pada kasus kali ini adalah dengan menggunakan pixel.

2. *Naïve String Search (Brute Force)*

Algoritma Pencarian *string* dengan cara *brute force* merupakan cara yang paling mudah dibuat. Sesuai dengan kata “*Brute Force*”, algoritma ini mencocokkan seluruh *string* yang terdapat pada *pattern* dengan yang ada di dalam teks secara keseluruhan.

Pencocokan dimulai dari awal *pattern* dan awal teks, setiap ada kesalahan maka *pattern* digeser 1 karakter. Sehingga pencarian dengan *brute force* akan membutuhkan waktu yang relatif lama jika dibandingkan dengan algoritma lain, contohnya algoritma Knuth-Morris-Pratt seperti yang dibahas pada bagian berikutnya.

```

ANPANMAN-
PAN-----
-PAN-----
--PAN-----
---PAN-----
----PAN-----
-----PAN-----

```

Gambar 1 Kesalahan pencocokan pada algoritma brute force dan pencocokan setelahnya

3. Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt merupakan algoritma pencarian string yang dikembangkan oleh Donald E. Knuth, James H. Morris, dan Vaughan R. Pratt di tempat dan waktu yang terpisah. Algoritma ini merupakan algoritma hasil pengembangan brute force dengan membuat algoritma yang lebih pintar melalui cara “mengingat” dan “melompat”.

Untuk menunjang hal tersebut, algoritma KMP menggunakan sebuah fungsi yang diberi nama *Border Function* yang digunakan untuk “melompati” pengecekan *sub-string* agar tidak perlu terjadi pengecekan berulang yang tidak perlu. Penggunaan *border function* adalah dengan mengingat banyaknya kesamaan *sub-string* sebelum kesalahan pencocokan ditemukan pada suatu karakter, dan menggunakannya untuk melakukan lompatan pergeseran. Algoritma KMP melakukan perbandingan *pattern* mulai dari kiri ke kanan seperti halnya yang dilakukan oleh algoritma *brute force*.

Cara kerja algoritma ini dimulai dengan membuat *Border Function*($B(x)$) dari pattern yang digunakan. Border function adalah sekumpulan kemiripan pada sub-pattern yang digunakan untuk “mengabaikan” sub-pattern tertentu pada pencarian berikutnya setelah kesalahan pencocokan terjadi.

i	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3
W[i]	P	A	R	T	I	C	I	P	A	T	E	I	N	P	A	R	A	C	H	U	T	E		
T[i]	-1	0	0	0	0	0	0	0	1	2	0	0	0	0	0	1	2	3	0	0	0	0	0	0

Gambar 2 Contoh Border Function

Selanjutnya setelah kita memiliki *border function*, maka kita hanya melakukan pencocokan seperti biasa. Hal yang membedakan algoritma KMP dengan algoritma brute force adalah cara penggeseran pattern yang dilakukan setelah terjadi kesalahan. Bila pada algoritma brute force, penggeseran pattern setelah terjadi kesalahan akan dilakukan dengan menggeser pattern sejauh 1 karakter pada teks dan mencocokkan seluruh pattern tersebut hingga kesalahan lainnya ditemukan. Sedangkan pada algoritma KMP pergeseran pattern akan disesuaikan dengan nilai pada border function di tempat karakter sebelum kesalahan pencocokan terjadi.

```

           1      2
m: 01234567890123456789012
S: ABC ABCDAB ABCDABCDABDE
W:           ABCDABD
i:           0123456

```

Gambar 3 Kesalahan pencocokan string

```

           1      2
m: 01234567890123456789012
S: ABC ABCDAB ABCDABCDABDE
W:           ABCDABD
i:           0123456

```

Gambar 4 Pencocokan setelah terjadi kesalahan pada gambar 2 (lihat posisi awal pattern dicocokkan)

II. PENCOCOKAN DUA BUAH GAMBAR

Pencocokan dua buah gambar dapat dilakukan dengan mencari pixel-pixel penyusunnya kemudian mencocokkannya dengan menggunakan algoritma pencarian yang telah ditentukan.

1. Brute Force

Bandung, 21 Desember 2012

A handwritten signature in black ink, appearing to read 'Yusuf Ardi Nugroho', with a long horizontal stroke extending to the left.

Yusuf Ardi Nugroho
13510002