Sistem Pencarian Berdasarkan Audio dan Visual

Irvan Aditya (13510016)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510016@std.stei.itb.ac.id

***Abstrak*—Teknologi sudah semakin maju. Mesin pencari tidak lagi hanya mampu mencari kecocokan dalam kata atau kalimat. Mesin pencari paling dikenal, Google, sudah dapat mencari gambar berdasarkan gambar atau cuplikan gambar yang kita berikan. Pencarian berdasarkan suara juga sudah tersedia. Pada dasarnya semuanya menggunakan prinsip *pattern matching*.**

***Kata Kunci—pattern matching*, pencarian, audio, visual**

I. PENDAHULUAN

Mesin pencari adalah suatu program yang dapat mencari objek berdasarkan kecocokannya dengan kunci yang dimasukkan. Pada awalnya mesin pencari dibuat untuk mencari kecocokan kata atau kalimat dalam sebuah teks. Saat ini, mesin pencari masih sangat mengandalkan teks untuk pencarian.

Salah satu contoh mesin pencari berbasis teks yang terkenal adalah aplikasi grep yang terpasang dalam setiap sistem operasi berbasis seperti-UNIX. Grep terkenal sangat cepat dalam mencari kecocokan dalam teks[2]. Namun seiring berkembangnya pengetahuan, mesin pencari tidak hanya dapat mencari kecocokan pola dalam teks. *Pattern matching* juga dapat dilakukan untuk mengenali kecocokan yang terdapat dalam berkas audio dan visual.

Saat ini, telah dikembangkan mesin pencari yang berbasis audio dan visual. Google sudah menerapkan pencarian berbasis visual, yaitu pencarian gambar. Pencarian gambar dengan Google sekarang dapat dilakukan dengan menggunakan gambar atau cuplikan gambar. Google akan menampilkan hasil pencarian berupa gambar yang serupa dan yang sama.

Gambar 1 Pencarian gambar menggunakan Google

Selain itu, di internet juga sudah beredar fasilitas pencarian berdasarkan audio. Fasilitas tersebut berupa pencarian lagu dengan menggunakan cuplikan lagu. Salah satu mesin pencari yang dikenal adalah sebuah aplikasi berbasis mobile yang bernama Shazam. Aplikasi ini merekam cuplikan suara, kemudian akan mencari kecocokannya dengan lagu yang ada. Jika berhasil Shazam akan memberitahukan judul lagu tersebut beserta tautan untuk menikmati lagu tersebut.

Gambar 2 Pencarian lagu menggunakan Shazam

Dalam mesin pencari, salah satu algoritma yang tidak dapat diabaikan adalah algoritma *pattern matching*. Ada cukup banyak algoritma untuk melakukan *pattern matching*. Untuk kasus teks, beberapa algoritma yang dikenal adalah algoritma Boyer-Moore dan Knuth-Morris-Pratt. Bahkan *pattern matching* dapat juga dilakukan secara brute force.

Dalam makalah ini akan dibahas bagaimana implementasi *pattern matching* dalam mesin pencari, terutama mesin pencari berbasis audio dan visual.

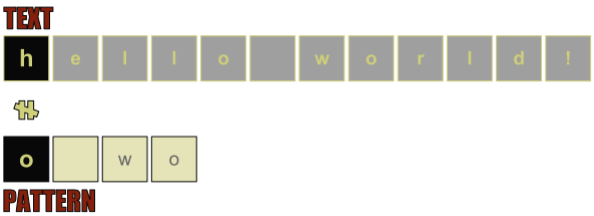
II. Dasar Teori *Pattern Matching*

Dasar implementasi *pattern matching* adalah mencocokkan pola yang terdapat dalam suatu data dengan data lainnya. Setidaknya ada tiga jenis algoritma *pattern matching*, khususnya pencocokan *string* yang telah dipelajari di kuliah Strategi Algoritma, yaitu brute force dan Boyer-Moore.

A. Algoritma Brute Force

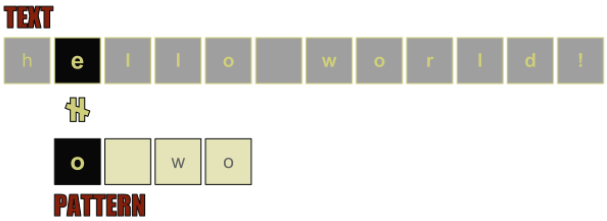
Proses pencarian *string* dengan menggunakan algoritma brute force sangat sederhana. Namun, di balik kesederhanaan algoritma brute force, ternyata algoritma ini tidak mangkus. Pencarian *string* dengan algoritma brute force akan relatif lebih lama dibandingkan kedua algoritma lainnya.

Pertama, kita akan mencocokkan karakter pertama dari pola dengan karakter pertama dari teks.



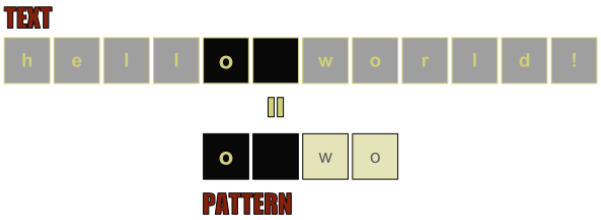
Gambar 3 Pencocokkan karakter pertama

Jika karakter pertama tidak cocok, kita maju ke karakter berikutnya dalam teks. Lalu kita cocokkan karakter tersebut dengan karakter pertama dalam pola.



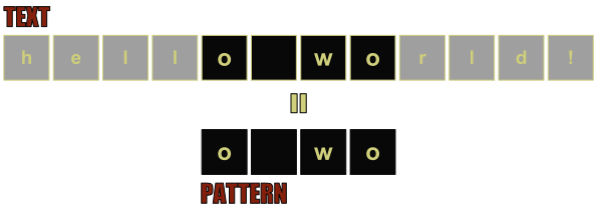
Gambar 4 Pencocokkan karakter kedua

Jika ternyata cocok, kita maju ke karakter kedua dalam pola dan karakter berikutnya dalam teks.



Gambar 5 Pencocokkan karakter berikutnya

Walaupun karakter telah cocok, bukan berarti pola tersebut ada di dalam teks. Kita harus terus menguji seluruh karakter sehingga terbukti bahwa pola tersebut ada di dalam teks.



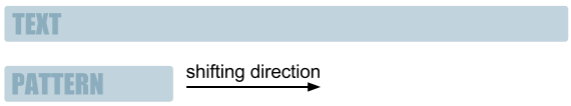
Gambar 6 Pola ditemukan dalam teks

Algoritma brute force dapat dikatakan lambat. Kompleksitas algoritma brute force adalah *O(nm)* di mana *n* adalah panjang dari teks dan *m* adalah panjang dari *string* pola.[4]

B. Algoritma Boyer-Moore

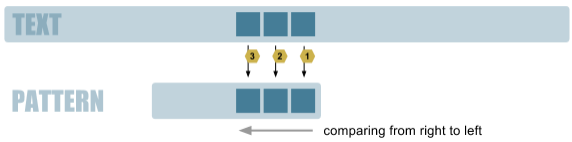
Algoritma Boyer-Moore disusun oleh Robert S. Boyer dan J. Strother Moore pada tahun 1977. Algoritma ini dapat memperbaiki kinerja pencarian pola dalam teks setelah melalui beberapa observasi.

Pertama, algoritma ini membandingkan pola dengan bagian paling kiri dari teks dan maju ke arah kanan.



Gambar 7 Pola bergerak dari kiri ke kanan

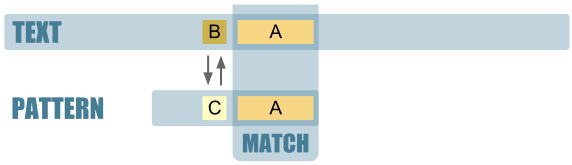
Tidak seperti algoritma pencarian *string* lainnya, algoritma Boyer-Moore membandingkan pola dari kanan menuju ke kiri.



Gambar 8 Perbandingan dimulai dari kanan

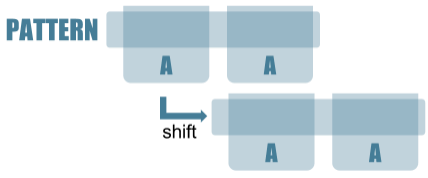
Gagasan utama algoritma Boyer-Moore untuk meningkatkan kinerja adalah observasi pola. Istilah-istilah dalam algoritma Boyer-Moore adalah *good-suffix shift* dan *bad-character shift*.

Kita mulai mencocokkan pola dari karakter paling kanan. Setelah beberapa karakter kita menemukan ketidakcocokan.



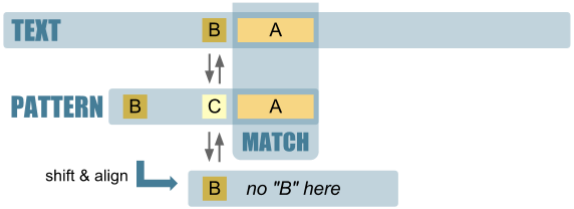
Gambar 9 Terjadi ketidakcocokan

Kita harus mengamati pola tersebut untuk menghindari perbandingan yang tidak perlu. Misalnya terdapat sebagian pola yang berulang. Kita harus menggeser pola sehingga posisinya sama dengan bagian yang terulang dalam teks.



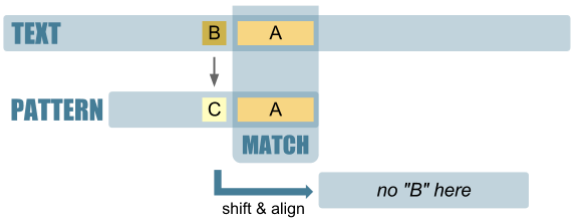
Gambar 10 Terdapat sebagian pola yang terulang

Teknik tersebut adalah dinamakan *good-suffix shift*. Selain itu ada juga teknik *bad-character shift*. Jika terjadi ketidakcocokan kita dapat menghindari perbandingan yang tidak perlu jika sepertinya karakter di teks tidak ada di dalam pola.



Gambar 11 Karakter ada di awal pola

Dalam contoh tersebut karakter ‘B’ berada di awal pola. Jadi kita bisa menggeser pola sehingga karakter ‘B’ sejajar dengan karakter yang sama dalam teks.



Gambar 12 Karakter tidak ada dalam pola

Jika karakter tidak ada dalam pola, kita dapat menggeser pola hingga melewati karakter yang tidak ada dalam pola tersebut.

Kompleksitas algoritma Boyer-Moore untuk kasus terburuk adalah *O(n+m)*. Algoritma ini bekerja dengan baik untuk bahasa alami.[5]

III. Prinsip *Pattern Matching* dalam Berkas Audio dan Visual

Untuk melakukan *pattern matching* dalam berkas audio dan visual, kita memerlukan pola yang tepat untuk dapat digunakan. Untuk itu, kita harus tahu apa saja yang dapat digunakan sebagai pola dalam *pattern matching*.

A. Penemuan pola dalam berkas gambar

Dalam memproses berkas gambar, kita dapat membuat pola berdasarkan fitur-fitur gambar. Ekstraksi fitur-fitur gambar adalah proses yang mendasar dalam memproses gambar. Beberapa fitur gambar yang sering digunakan adalah fitur visual, fitur statistik pixel, dan fitur koefisien transformasi.[6]

Fitur-fitur visual dalam gambar telah mengalami standardisasi MPEG-7. Dalam standard MPEG-7 ada lima buah *descriptor*, yaitu *Dominant Colour Descriptor, Scalable Colour Descriptor, Colour Structure Descriptor, Color Layout Descriptor,* dan *Edge Histogram Descriptor.*

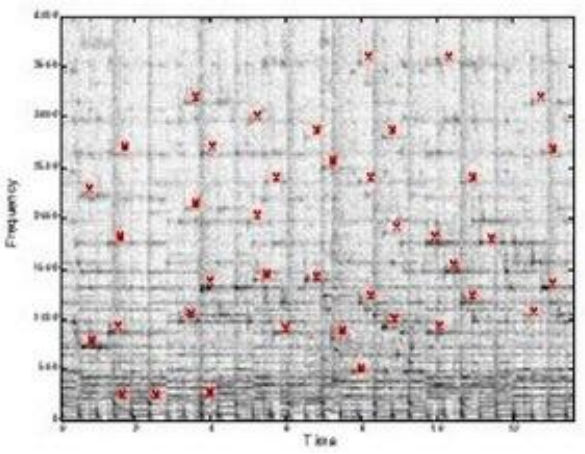
Secara umum, fitur terbaik untuk tidak diketahui dan berbeda-beda untuk setiap konsep. Konsep biasanya tidak dikarakterisasi oleh hanya satu fitur.[7]

B. Penemuan pola dalam berkas suara

Dalam menentukan pola dalam berkas suara, teknik yang digunakan biasanya disebut *audio fingerprinting*. Semua *fingerprint* disimpan dalam basis data. Salah satu tekniknya adalah menggunakan grafik waktu dan frekuensi yang disebut *spectrogram.*

Dalam *spectrogram* terdapat intensitar dari frekuensi setiap waktu. Jika waktu adalah sumbu-X dan frekuensi adalah sumbu-Y, garis horizontal merepresentasikan nada yang berkelanjutan dan garis vertikal merepresentasikan suara gangguan yang instan dan mendadak.

Dalam contoh spectrogram di Gambar 13, setiap titik merah adalah intensitas puncak.[8]



Gambar 13 Contoh *spectogram*

B. Pencocokan pola

Setelah mendapatkan pola-pola yang siap diproses, barulah berkas audio dan visual dapat diproses dengan algoritma yang sudah ada. Biasanya pemrosesan dilakukan dengan mencocokkan secuplik berkas gambar atau suara dan membandingkannya dengan berkas-berkas yang ada dalam basis data.

V. KESIMPULAN

Pattern matching dapat diterapkan dalam pencarian selain teks. Hanya saja kita perlu mencari pola yang tepat untuk dicocokkan menggunakan algoritma *pattern matching* yang sudah dikembangkan.

REFERENSI

[1] http://google.com/imghp diakses tanggal 19 Desember 2012.

[2] http://lists.freebsd.org/pipermail/frebsd-current/2010-August/019310.html diakses tanggal 19 Desember 2012.

[3] http://appleinsider.com/articles/09/05/14/apple_att_sued_over_ties_to_shazam_music_id_service.html diakses tanggal 19 Desember 2012.

[4] http://www.stoimen.com/blog/2010/03/07/computer-algorithms-brute-force-*string*-matching/ diakses tanggal 20 Desember 2012.

[5] http://www.stoimen.com/blog/2010/04/17/computer-algorithms-boyer-moore-*string*-search-and-matching/ diakses tanggal 20 Desember 2012.

[6] http://www.sciencedirect.com/science/article/pii/003132039190063B diakses tanggal 20 Desember 2012.

[7] http://ieeexplore/ieee.org/stamp/stamp.jsp?arnumber=04061152 diakses tanggal 20 Desember 2012.

[8] http://gizmodo.com/5647458/how-shazam-works-to-identify-nearly-every-song-you-throw-at-it diakses tanggal 20 Desember 2012.

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, 21 Desember 2012

ttd

Irvan Aditya

13510016