Penerapan Algoritma Pencocokan String dalam Perangkat Lunak Pemblokir Akses Situs Negatif

Ahmad Aidin - 13513020
Program Studi Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung (ITB)
Bandung, Indonesia
13513020@std.stei.itb.ac.id

***Abstrak* - Kemudahan akses informasi memberikan peluang besar bagi manusia melakukan berbagai positif yang bisa membantu masalah di sekitarnya. Di sisi lain, peluang ini bisa digunakan untuk hal negatif yang menimbulkan kerugian terhadap dirinya sendiri dan bahkan orang lain. Dengan adanya berbagai dampak kehidupan yang ditimbulkan oleh penggunaan internet yang tidak baik, diperlukan kakas yang bisa membatasi akses terhadap situs-situs negatif. Algortima yang bisa digunakan dalam aplikasi ini adalah algortima *brute force, Knuth-Morris-Pratt,* dan Boyer-Moore, yaitu algoritma pencocokan *string* pola dengan suatu teks.**

Kata kunci - Penggunaan internet, blokir situs negatif, pencocokan string

I. PENDAHULUAN

Pesatnya pertumbuhan teknologi informasi membuat pertumbuhan yang pesat terhadap pengaksesan internet. Tercatat, di Indonesia saja ada sebanyak 132,7 juta penduduk yang telah terhubung di Internet dari total populasi penduduk Indonesia sebanyak 256,2 juta. Data ini didasarkan pada survey yang dilakukan oleh Asosiasi Penyelenggara Internet Indonesia pada tahun 2016. Hal ini menunjukan kenaikan pengguna internet sebanyak 51,8 persen dibandingkan pada tahun 2014 yaitu hanya 88 juta pengguna.

Pengguaan internet yang meningkat tentunya akan berdampak positif jika digunakan untuk keperluan yang bermanfaat dan sebaliknya jika digunakan untuk hal yang sia-sia apalagi hal yang buruk. Kenyataannya, penggunaan internet untuk hal yang negatif sangat besar. Pada surat kabar terdapat dapat pengaksesan situs porno yang didapatkan dari salah satu situs porno terbesar dunia. Pada data ini tercatat bahwa pada tahun 2016, total durasi pemutaran video porno pada situs tersebut sebanyak 4.599.000.000 jam atau sebanyak 12.600.000 jam setiap hari. Jumlah ini terhitung sangat besar mengingat hanya satu situs yang diperhitungkan padahal di Internet sangat banyak situs lainnya.

Persoalan penggunaan internet yang tidak semestinya dapat menimbulkan persoalan-persoalan lain yang serius seperti seks bebas, pelecehan seksual, hingga perkosaan. Berdasarkan data komnas perempuan, sepanjang tahun 2016 terdapat pelaporan kasus perkosaan sebanyak 1.389 kasus yang diikuti kasus pencabulan sebanyak 1.266 kasus. Salah satu faktor penyebab terjadinya kasus ini adalah aktivitas menonton video porno.

Melihat buruknya dampak yang diakibatkan pengaksesan situs negatif di internet, perlu adanya tindakan preventif untuk mencegah hal itu terjadi. Salah satu tindakan preventif yang bisa dilakukan adalah pencegahan akses terhadap situs-situs negatif oleh pengguna. Metode yang bisa dilakukan adalah menggunakan aplikasi tambahan pada *browser* atau penambahan fungsi pengecekan pada mesin pencarian untuk melakukan pemblokiran akses terhadap suatu situs. Pemblokiran didasarkan pada pustaka yang berisi daftar alamat situs negatif dan daftar kata yang dikategorikan sebagai kata yang mengantarkan pada situs-situs negatif. Jika ada pengguna yang memasukkan alamat ataupun kata-kata yang ada pada pustaka maka akses akan dihentikan.

II. DASAR TEORI

A. Situs Negatif

Berdasarkan Peraturan Menteri Komunikasi dan Informatika No 19 Tahun 2014 tentang Penanganan Situs Internet Bermuatan Negatif, situs internet bermuatan negatif adalah pornografi, dan kegiatan ilegal lainnya berdasarkan ketentuan peraturan perundang-undangan. Kegiatan ilegal yang dimaksud adalah yang pelaporannya berasal dari Kementrian atau Lembaga Pemerintah yang berwenang sesuai dengan peraturan perundang-undangan. Adapun pelaporan yang berasal dari masyarakat bisa disampaikan kepada Direktur Jenderal melalui fasilitas penerimaan laporan berupa e-mail aduan dan/atau pelaporan berbasis situs yang disediakan. Pelaporan dari masyarakat bisa dikategorikan mendesak bila menyangkut privasi, pornografi anak, kekerasan, suku, agama, ras, antargolongan (SARA), dan/atau muatan lain yang berdampak negatif dan menimbulkan keresahan masyarakat luas.

B. Konsep String

String adalah susunan dari karakter yang terdapat dalam suatu abjad. Prefiks dari suatu *string* adalah *substring* dari *string* tersebut yang berisi karakter pertama string hingga satu karakter sebelum karakter terakhir. Sufiks dari suatu *string*

adalah *substring* dari *string* tersebut yang berisi karakter terakhir hingga satu karakter sebelum kedua dari *string*.

- Misalkan *S* adalah *string* dengan ukuran *m*
  $S = x_0 x_1 \ldots x_{m-1}$
- Prefiks dari *S* adalah *substring S[0…k]*
- Sufiks dari *S* adalah *substring S[k…m-1]*
  k adalah indeks antara *0* dan *m-1*

Contoh:
Terdapat *string* "STRING".

- Semua prefiks yang mungkin: "S", "ST", "STR", "STRI", dan "STRIN"
- Semua sufiks yang mungkin: "G", "NG", "ING", "RING", dan "TRING"

C. *Pencocokan String*

Suatu string dikatakan cocok dengan suatu teks dikatakan cocok apabila pada teks terdapat susunan karakter yang sama dengan string tersebut. Algorirma pencocokan string mengeluarkan lokasi dimana string ditemukan pada teks.

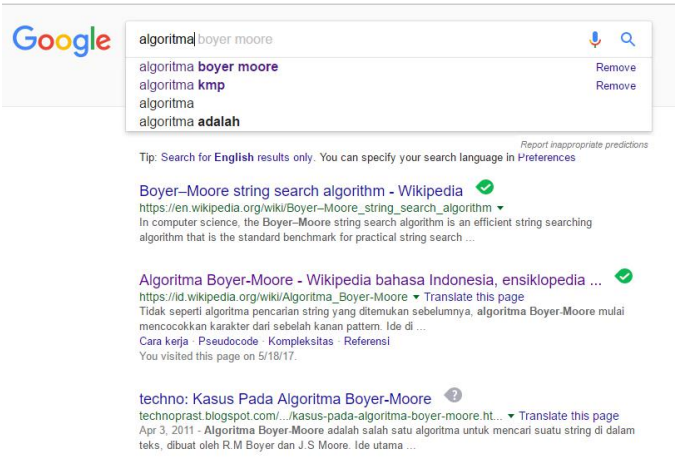
Definisi:

Diberikan :

1. T: Teks, yaitu string yang panjangnya n karakter
2. P: Pola, yaitu string yang panjangnya m karakter dengan asumsi m<<<n. Pola akan dicari di dalam teks
3. Akan dicari lokasi pertama di dalam teks yang sesuai dengan pola.

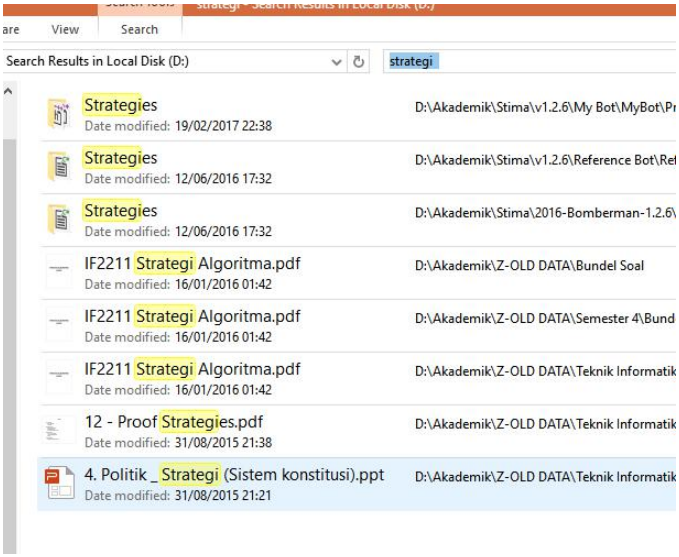
Contoh:
T: "ini adalah contoh teks yang berisi pola yang dicari"
P: "cari"
Beberapa algortima yang bisa digunakan dalam pencocokan string ini adalah *Brute Force*, *Knuth-Morris-Pratt*, dan *Boyer-Moore*

Contoh aplikasi pencocokan *string*/pola:
1. mesin pencarian situs internet



Gambar 1 contoh penggunaan pencocokan string pada mesin pencarian situs internet
(sumber: dokumentasi pribadi)

2. Pencarian berkas



Gambar 2 contoh penggunaan pencocokan string pada pencarian berkas di *explorer*
(sumber: dokumentasi pribadi)

D. *Algortima Brute Force*

Algortima *Brute Force* adalah algortima yang bisa dipakai untuk persoalan apapun. Pada pencocokan string ini, algoritma *Brute Force* melakukan pengecekan untuk setiap karakter hingga ditemukan pola yang ingin dicari pada teks atua karakter pada teks sudah dicocokkan semuanya. Pengecekan dilakukan dari kiri ke kanan. Jika ditemukan suatu karakter pada pola tidak cocok dengan karakter pada teks, pola digeser satu karakter ke kanan dan memulai pencocokan dari awal karakter pada pola.

Langkah *Brute Force* yang dilakukan sebagai berikut.

1) Pola dicocokkan dari awal teks
2) Lakukan pencocokan dari karakter pertama pola
3) Jika ditemukan karakter sesuai, lanjutkan untuk karakter berikutnya pada pola dan teks hingga ditemukan ketidaksesuaian atau seluruh karakter pada pola sesuai dengan teks.
4) Jika ditemukan ketidaksesuaian, geser pola satu karakter ke kanan dan ulangi dari langkah 2

Pseudocode untuk bruteforce

```
repeat
   if (char of text == char of pattern) then
        compare next char of pattern to next
        char of text
   else
        shift pattern by one char
until (all char match or end of text)
```

Contoh:

```
Pola: SUKASARI
Teks: KU SUKA SUKASARI

 KU SUKA SUKASARI,KAK
1 S
2  S
3   S
4    SUKAS
5     S
6      S
7       S
8        S
9         SUKASARI
```

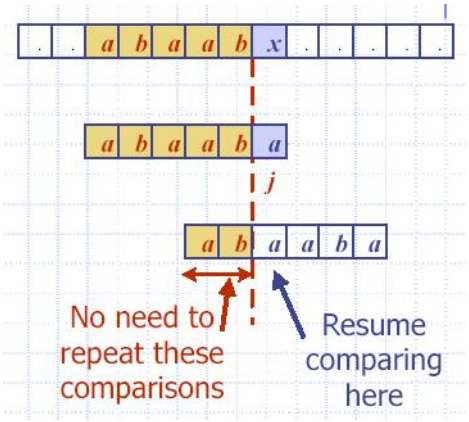
Pada contoh tersebut, pola ditemukan pada indeks ke 9 dari teks.

Untuk kasus terbaik, yaitu jika karakter pertama pada pola tidak pernah sama dengan karakter pada teks, perbandingan yang dilakukan adalah sebanyak maksimal *n* kali. Pada kasus ini, kompleksitas algoritma *brute force* adalah *O(n)*.

Untuk kasus terburuk, yaitu seluruh karakter pada pola kecuali karakter terakhir selalu sesuai pada teks, jumlah perbandingan yang dilakukan adalah sebanyak *m(n-m+1) = O(mn)*. Untuk kasus pada umumnya, kompleksitas algortima ini bisa mencapai *O(m+n)*. Algortima ini cepat jika alfabet yang digunakan memiliki jumlah karakter yang banyak, dan sebaliknya lambat jika jumlah karakter pada alfabet sedikit.

E. Algoritma Knuth-Morris-Pratt (KMP)

Tahun 1977, Donald E. Knuth dan James H. Morris bersama Vaughan R. Pratt mempublikasikan Algoritma *Knuth-Morris-Pratt (KMP)*. Algoritma ini adalah perbaikan dari algortima *bruto force*. Jika algortima brute force hanya bergeser satu karakter ketika menemukan ketidakcocokan karakter, maka KMP memungkinan pergeseran yang lebih banyak. Idenya adalah jika terjadi ketidakcocokan pola dan teks pada indeks *j* pada pola, maka pergeseran terjauh yang bisa dilakukan adalah bergeser hingga prefiks terbesar dari *P[0..j]* yang juga sufiks dari *P[1..j]* berada pada posisi sufiks yang bersangkutan. Contoh:



Untuk melakukan implementasi algortitma KMP, digunakan *border function* (fungsi pinggiran), yaitu suatu preproses terhadap pola untuk menemukan kecocokan prefiks dan sufiks dari pola tersebut. Misalkan, ketika proses pencocokan ditemukan ketidakcocokan karakter pada indeks *j* pada pola *P,* dan *k=j-1,* maka fungsi pinggiran *b(j)* mengembalikan nilai ukuran terbesar dari prefiks *P[0..j]* yang juga sufiks dari *P[1..j]*. Fungsi pinggiran ini memiliki nama lain fungsi kegagalan *(Failure function).* Selanjutnya jumlah pergeseran yang dilakukan adalah sebanyak *j-b(j).*

Pseudocode algoritma failure function KMP sebagai berikut.

```
failure function b()
while (pattern not found and not end of text)do
   if (char of pattern == char of text) then
      compare next char of patter to next char
      of text
   else if (not first check) then
      next pattern index = b(current pattern
      index - 1)
   else check next char of text
```

Algortima KMP memiliki kompleksitas waktu *O(m+n)* dengan kompleksitas waktu perhitungan fungsi pinggiran *O(m)* dan kompleksitas waktu pencaria string *O(n)*. Keuntungan penggunaan algoritma ini adalah karakter pada teks yang sudah pernah dicek kecocokannya dengan pola tidak pernah dicek ulang. Tentu saja ini sangat bangus apabila karakter pada teks sangat besar.

F. Algoritma Boyer-Moore

Algortima pencocokan string lainnya adalah algortima Boyer Moore. Algortima ini dipublikasikan oleh Robert S. Boyer dan J. Storther Moore pada tahun 1977. Algortima ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Cara kerja algortima ini adalah dengan mencocokan pola pada teks dari kanan ke kiri. Teknik ini disebut dengan teknik *looking-glass.* Selanjutnya jika tidak cocok, pola akan digeser ke kanan dan dicocokkan lagi, yang disebut dengan teknik *character-jump.* Terdapat 3 kemungkinan ketika proses pencocokan dilakukan. Misalkan saat ini sedang mencocokkan huruf c yang ada pada teks.

Kemungkinan pertama, pada pola terdapat huruf c maka geser pola hingga c pada teks bersesuaian dengan c dengan posisi paling kanan pada pola. Kemungkinan kedua, pada pola terdapat c tetapi c ini sudah berada di sebelah kanan huruf yang saat ini dicocokkan maka geser pola satu huruf ke kanan. Kemungkinan ketiga, yaitu ketidakcocokan selain kemungkinan pertama dan kedua, (tidak ada c pada pola) maka geser pola hingga huruf pertamanya bersesuaian dengan huruf setelah c pada teks.

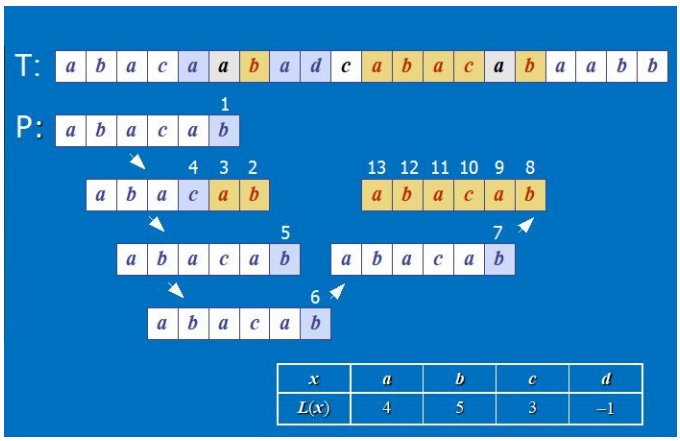
Algoritma Boyer Moore melakukan preposes terhadap pola dan alfabet yang dipakai sebelum dicocokkan dengan teks. Preproses ini membentuk fungsi *Last Occurence,* yaitu fungsi yang memetakan seluruh huruf di alfabet dan lokasi kemunculan terakhirnya pada pola.

Definisi fungsi *Last Occurence, L(x):*

- *L(x)* adalah i, yaitu indeks terakhir pada pola, sedemikian sehingga *P(i) == x,* atau
- -1, jika tidak ada *x* pada *P*

```
last occurrence function L()
   repeat
      if (char of pattern == char of text)then
         if (current pattern index == 0) then
             return current text index
         else check previous letter
      else jump pattern according to L()
         function
   until (pattern found or end of text)
```

Contoh pemakaian algortima *Boyer-Moore*



Kompleksitas terburuk algortima Boyer Moore adalah *O(mn+ jumlah abjad)*, Algortima ini cepat jika jumlah abjadnya banyak dan lambat jika jumlah abjad sedikit. Misalnya bagus untuk teks-teks umum dan kurang bagus untuk biner atau rantai DNA.

III. Analisis dan Penyelesaian Persoalan

Pemblokiran akses situs negatif adalah metode preventif guna mengurangi penggunaan negatif internet. Pemblokiran ini bisa dilakukan terhadap akses pada suatu alamat atau juga bisa menolak penggunaan kata-kata tertentu pada mesin pencarian. Untuk melakukan pemblokiran ini dibutuhkan kakas ataupun fungsi guna mengecek suatu masukan pengguna termasuk ke dalam daftar alamat situs negatif ataupun kata-kata yang berkaitan dengan situs negatif.

Langkah yang perlu dilakukan melakukan pemblokiran adalah membuat suatu pustaka yang berisi alamat situs negatif dan kata-kata tertentu yang terkait dengan situs negatif. Masukan kakas atau fungsi berasal dari pengguna. Selanjutnya masukan pengguna dicocokkan dengan setiap isi pustaka hingga ditemukan kecocokan atau tidak ada satupun isi pustaka yang sesuai. Jika panjang *string* masukan pengguna lebih panjang dari isi pustaka yang sedang dicocokkan, isi pustaka dianggap sebagai pola dan sebaliknya jika isi pustaka lebih panjang dari *string* masukan pengguna, maka masukan pengguna yang dianggap sebagai pola.

Pada tahap pencocokan string, bisa diunakan algortima *brute force*, KMP, ataupun Boyer More. Output yang diharapkan adalah lokasi indeks ditemukannya pola pada teks. Jika keluaran -1, berarti tidak ditemukan pola pada teks sehingga masukan pengguna dianggap tidak mengarah ke situs negatif. Jika keluaran selain itu, maka pola ditemukan pada teks sehingga masukan pengguna dianggap mengarah ke situs negatif. Jika masukan pengguna mengarah ke situs negatif, akses terhadap situs tersebut akan diblokir.

IV. Analisis Algoritma

Analisis terhadap algortima dilakukan berdasarkan eksperimen. Akan dikumpulkan beberapa alamat yang dianggap sebagai situs negatif dan kata-kata yang dianggap bisa mengarah ke situs negatif sehingga dari ketiga algoritma yaitu *brute force, Knoth-Morris-Pratt,* dan *Boyer-Moore* bisa dilihat mana yang paling cepat eksekusinya.

Percobaan dilakukan dengan isi pustaka kata terlarang diambil dari lima puluh nama situs porno terkenal di dunia, dua puluh kategori porno dan dua puluh aktris porno yang menjadi *keyword* pencarian terpopuler untuk mengakses porno. Selain untuk memperbanyak data, ditambah seribu teks *dummy* di awal pencarian. Percobaan akan dilakukan sebanyak empat kali untuk setiap algortima, percobaan pertama dan kedua akan memakai kata kunci yang lebih panjang dari seluruh kata-kata di pustaka. Bedanya pada percobaan pertama kata kunci akan memuat kata yang ada di pustaka sedangkan percobaan kedua tidak. Percobaan ketiga dan keempat akan memakai kata kunci bisa lebih pendek atau lebih panjang tergantung kata di pustaka. Pada percobaan ketiga ini kata kunci dibuat tidak cocok dengan isi pustaka sedangkan percobaan keeempat akan sesuai dengan isi pustaka, Untuk *keyword* yang lebih pendek dari isi pustaka, maka *keyword* berperan sebagai pola, dan sebaliknya apabila *keyword* lebih panjang dari isi pustaka, maka *keyword* berperan sebagai teks.

Berikut hasil percobaan yang dilakukan.

- Algortima *Brute Force, p*ercobaan 1

```
Waktu eksekusi : 2112 microseconds

Brute Force: 'free streaming video mia
khalifa' dilarang
```

- Algortima *Brute Force, p*ercobaan 2

```
Waktu eksekusi : 3369 microseconds

Brute Force: 'free downloads love akira in
texas 2017 film blue ray HD quality' aman
```

- Algortima *Brute Force, p*ercobaan 3

```
Waktu eksekusi : 1037 microseconds

Brute Force: 'suka hen' aman
```

- Algortima *Brute Force, p*ercobaan 4

```
Waktu eksekusi : 1137 microseconds
Brute Force: 'porn' dilarang
```

- Algortima *Knuth-Morris-Pratt,* percobaan 1

```
Waktu eksekusi : 5058 microseconds
Knuth-Morris-Pratt: 'free streaming video mia khalifa' dilarang
```

- Algortima *Knuth-Morris-Pratt,* percobaan 2

```
Waktu eksekusi : 4862 microseconds
Knuth-Morris-Pratt: 'free downloads love akira in texas 2017 film blue ray HD quality' aman
```

- Algortima *Knuth-Morris-Pratt,* percobaan 3

```
Waktu eksekusi : 1656 microseconds
Knuth-Morris-Pratt: 'suka hen' aman
```

- Algortima *Knuth-Morris-Pratt,* percobaan 4

```
Waktu eksekusi : 1557 microseconds
Knuth-Morris-Pratt: 'porn' dilarang
```

- Algortima *Boyer-Moore*, percobaan 1

```
Waktu eksekusi : 4695 microseconds
Boyer-Moore: 'free streaming video mia khalifa' dilarang
```

- Algortima *Boyer-Moore*, percobaan 2

```
Waktu eksekusi : 6304 microseconds
Boyer-Moore: 'free downloads love akira in texas 2017 film blue ray HD quality' aman
```

- Algortima *Boyer-Moore*, percobaan 3

```
Waktu eksekusi : 2821 microseconds
Boyer-Moore: 'suka hen' aman
```

- Algortima *Boyer-Moore*, percobaan 4

```
Waktu eksekusi : 7485 microseconds
Boyer-Moore: 'porn' dilarang
```

Berikut hasil waktu eksekusi untuk tiap algortima dalam mikrodetik.

	Brute Force	*KMP*	*Boyer-Moore*
Percobaan 1	2112	5058	4695
Percobaan 2	3369	4862	6304
Percobaan 3	1037	1656	2821
Percobaan 4	1137	1557	7485
Rata-rata	1913,75	3283,25	5326,5

Hasil percobaan menunjukkan bahwa algortima *Brute Force* adalah algoritma yang paling cepat untuk persoalan ini dibandingkan *Knuth-Morris-Pratt* dan *Boyer-Moore.* Meskipun kompleksitas terbutuk dari brute force adalah *O(mn),* algortima brute force ini bagus untuk teks yang pendek, dan pada persoalan ini text yang digunakan adalah teks yang pendek.

V. KESIMPULAN

Algortima pencocokan string yang paling mangkus untuk digunakan dalah persoalan ppemblokiran akses terhadap situs negatif adalah algoritma *brute force.* Meskipun kadang algoritma ini dihindari untuk hal yang membutuhkan efisiensi dalam pemrosesan, dalam pencocokan string untuk teks yang pendek, algortima ini sangat bagus kinerjanya.

UCAPAN TERIMAKASIH

Penulis pertama-tama ingin mengucapkan syukur kepada Tuhan Yang Maha Esa karena rahmat dan berkat-Nya yang selalu menyertai penulis hingga pembuatan makalah ini selesai. Penulis juga ingin berterima kasih kepada kedua orang tua penulis yang selalu memberi support dan semangat kepada penulis. Tak lupa penulis ucapkan terima kasih kepada Bapak Rinaldi Munir dan Ibu Nur Ulfa Maulidevi dan Ibu Masayu Laylia Kodra karena melalui pengajarannya, penulis dapat memahami konsep algoritma termasuk didalamnya algoritma pencarian string yang menjadi dasar makalah Strategi Algoritma IF2211.

REFERENSI

[1] Dr.Ir.Rinaldi Munir,M.T. , Diktat Kuliah IF2111 : Strategi Algoritma, Bandung : Program Studi Teknik Informatika, Insitut Teknologi Bandung , 2009

[2] Menteri Komunikasi dan Informatika Republik Indonesia, "Peraturan Menteri Komunikasi dan Informatika RI no 19 tahun 2014 tentang Penanagan Situs Internet Bermuatan Negatif", Jakarta, 2014

[3] The top 500 sites on the web. Diakses pada May 18 Mei, 2017. Dari: http://www.alexa.com/topsites/category/Top/Adult.

[4] Erin Korber. String Matching Algorithms,Lecture 28: 14 Aug.CS61B Summer 2006. Diakses pada 18 Mei 2017. Dari http://inst.eecs.berkeley.edu/~cs61b/su06/lecnotes/lec28.pdf

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 April 2017

Ahmad Aidin
13513020