

Penerapan Algoritma Pencocokan String untuk Menentukan Keamanan URL

Hera Shafira 13519131 (*Author*)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail : 13519131@std.stei.itb.ac.id

Abstract—URL (*Uniform Resource Locator*) merupakan suatu alamat yang akan membawa pengguna internet kepada website tertentu. Di era kemajuan teknologi saat ini, selain berperan sebagai sarana berkomunikasi, media sosial juga dapat berperan sebagai sarana penyebaran ancaman bagi penggunaannya. Salah satu media perantara ancaman yang umum digunakan pada media sosial adalah berupa URL berbahaya yang disebarkan oleh oknum tertentu. URL berbahaya dapat mengakibatkan berbagai macam ancaman pada pengunjungnya seperti instalasi malware pada perangkat pengunjung. Namun sayangnya, masih banyak pengguna internet yang belum mengetahui perbedaan URL berbahaya dengan URL yang aman untuk dikunjungi. Pada makalah ini, penulis akan menerapkan algoritma pencocokan string pada suatu URL dengan pattern tertentu untuk menguji keamanan dari URL tersebut.

Keywords—URL, keamanan, pencocokan string, media sosial

I. PENDAHULUAN

Internet merupakan sistem jaringan komputer yang saling terhubung secara global dengan menggunakan paket protokol internet (TCP/IP) untuk menghubungkan perangkat-perangkat di seluruh dunia. Jaringan ini terdiri dari jaringan privat, publik, akademik, bisnis, dan pemerintah lokal ke lingkup global yang dihubungkan oleh beragam teknologi elektronik, nirkabel, dan jaringan optik. Fungsi utama internet adalah untuk mempermudah akses sumber informasi, seperti dokumen hiperteks yang saling terkait dan aplikasi World Wide Web (WWW), email, telepon, dll.

Salah satu sistem informasi yang paling umum digunakan dalam Internet adalah World Wide Web (WWW). World Wide Web (WWW) atau yang cukup dikenal sebagai web merupakan sistem informasi yang menyimpan dokumen-dokumen atau sumber informasi lainnya dengan bantuan Uniform Resource Relocator (URL) sebagai alat pengidentifikasi suatu web. Dokumen-dokumen yang tergabung dalam WWW mungkin saja saling bertautan dengan adanya hyperlink dan seluruh dokumen yang terdapat pada WWW dapat diakses melalui Internet. Sumber informasi pada web ditransfer melalui Hypertext Transfer Protocol (HTTP) yang dapat diakses oleh pengguna dengan menggunakan aplikasi perangkat lunak yang disebut browser web, sebelumnya, sumber informasi ini diterbitkan terlebih dahulu oleh aplikasi perangkat lunak yang disebut server web.

Uniform Resource Locator (URL) yang digunakan untuk mengakses dokumen pada World Wide Web (WWW) sendiri adalah rangkaian karakter yang mengikuti suatu format standar tertentu dan digunakan untuk menunjukkan alamat suatu sumber informasi seperti dokumen dan gambar di Internet.

Seiring dengan perkembangan teknologi, kini penggunaan internet pun menjadi semakin pesat. Salah satu faktor yang mendorong melesatnya penggunaan internet adalah munculnya fenomena media sosial. Media sosial merupakan sebuah media daring yang digunakan oleh para penggunanya agar bisa dengan mudah berpartisipasi, berinteraksi, berbagi, dan menciptakan isi blog, jejaring sosial, wiki, forum dan dunia virtual tanpa dibatasi oleh ruang dan waktu. Tentu saja, walaupun media sosial hadir dengan segudang manfaatnya, media sosial juga memiliki ancaman tersendiri bagi penggunanya.

Salah satu ancaman yang timbul dari maraknya penggunaan media sosial adalah bermunculannya akun-akun palsu yang akan menyebar URL kepada suatu golongan pengguna media sosial. Selain itu, perkembangan WWW secara umum pun menimbulkan fenomena baru yang lain yaitu munculnya iklan pada laman web. Walaupun iklan pada laman web dibuat dengan tujuan yang baik yaitu sebagai sarana untuk mempromosikan hal apapun yang bisa disimak oleh banyak orang, namun nyatanya terdapat beberapa oknum yang memanfaatkan iklan sebagai sarana untuk menyebar URL berbahaya pada pengunjung laman web yang mengklik iklan tersebut.

URL berbahaya sendiri merupakan sebuah URL yang diciptakan dengan tujuan khusus untuk melakukan penipuan dan penyerangan cyber. Mengklik URL berbahaya dapat mengakibatkan terunduhnya ransomware, virus, trojan, atau malware lain yang mungkin dapat merusak komputer pengguna dan jaringannya. URL berbahaya juga dapat digunakan untuk memperoleh informasi sensitif dari pengunjungnya dengan cara berkamuflase sebagai laman web yang terpercaya. Tentu saja hal ini sangat berbahaya bagi pengguna internet yang awam akan keberadaan URL seperti ini.

Sesungguhnya suatu URL memiliki penanda umum yang bisa membedakan apakah URL tersebut aman untuk dikunjungi atau tidak. Kemudian, terdapat juga tendensi suatu grup oknum pemilik URL berbahaya untuk memiliki beberapa URL

berbahaya dengan URL yang mirip antara satu sama lain. Data-data URL yang berbahaya sendiri bisa diperoleh dari suatu laman khusus yang menyediakan data URL berbahaya pada internet. Dengan adanya fakta-fakta tersebut ditambah dengan fakta bahwa seluruh URL merupakan *string*, maka tentu saja ciri-ciri URL berbahaya ini bisa diidentifikasi dengan menggunakan algoritma pencocokan string.

Dengan mengidentifikasi pattern yang menandakan keberbahayaan dari suatu URL menggunakan algoritma pencocokan string, maka diharapkan URL-URL berbahaya ini dapat diidentifikasi secara lebih baik sehingga pengguna internet pun dapat menghindari untuk mengunjungi URL tersebut.

II. LANDASAN TEORI

A. Algoritma Pencocokan String

Algoritma pencocokan string merupakan sebuah algoritma yang digunakan untuk mencari kemunculan sebuah string pendek yang disebut sebagai $pattern[0..n-1]$ pada sebuah string yang lebih panjang yang disebut sebagai $teks[0..m-1]$ dengan $m \geq n$. String sendiri merupakan tipe data yang digunakan untuk menyimpan barisan karakter. Contoh dari algoritma pencocokan string yang umum digunakan adalah algoritma brute force, algoritma Knuth-Morris-Pratt (KMP), dan algoritma Boyer-Moore (BM). Dalam dunia nyata, algoritma pencocokan string dapat digunakan untuk melakukan pencarian dalam editor text, web search engine, analisis citra, bioinformatics, dll.

Sebuah string sendiri dapat memiliki prefix dan suffix. Prefix merupakan awalan dari sebuah string, sedangkan suffix merupakan akhiran dari sebuah string. Apabila misal, terdapat sebuah string $S = x_0x_1...x_{m-1}$. Maka prefix dari string tersebut adalah $S_{prefix} = x_0x_1...x_{p-1}$ dengan $p \leq m$ dengan karakter yang berurutan diambil dari awal string ke akhir string, sedangkan suffix dari string tersebut adalah $S_{suffix} = x_0x_1...x_{p-1}$ dengan $p \leq m$ dengan karakter yang berurutan diambil dari akhir string ke awal string. Apabila misal terdapat string $S = \text{"Hera"}$, maka prefix yang memungkinkan dari string ini secara berurutan adalah "H", "He", "Her", dan "Hera". Kemudian, suffix yang memungkinkan dari string ini secara berurutan adalah "a", "ra", "era", dan "Hera".

B. Algoritma Brute Force

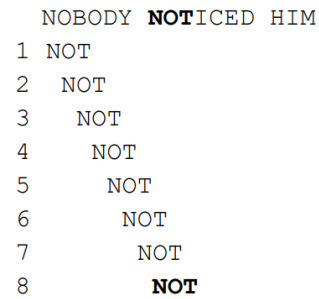
Algoritma Brute Force merupakan algoritma pencocokan string yang paling sederhana karena algoritma ini hanya perlu untuk melakukan pencocokan pada tiap posisi pada suatu teks T untuk melihat apakah pattern P dimulai pada posisi tersebut. Algoritma brute force dilakukan dengan tahapan sebagai berikut :

1. Mulai cocokan pattern P dengan string pada posisi awal teks T, pencocokan dilakukan dari kiri ke kanan.
2. Bandingkan setiap karakter pada pattern P dengan karakter yang bersesuaian pada teks T.
3. Apabila ditemukan mismatch (karakter yang tidak sesuai dengan pattern) maka pattern digeser 1 indeks ke kanan pada teks T.

4. Lakukan terus hal ini hingga berhasil ditemukan match untuk seluruh karakter pada pattern.

Kompleksitas algoritma brute force pada kasus terburuk adalah $O(mn)$, sedangkan kompleksitas kasus terbaiknya adalah $O(n)$. Namun, pada rata-rata kasus, kompleksitas algoritma ini adalah $O(m+n)$. Secara umum, algoritma ini bagus digunakan ketika karakter yang digunakan dalam suatu teks bervariasi. Namun, algoritma ini akan kurang baik untuk digunakan apabila karakter yang digunakan dalam suatu teks hanya sedikit, seperti pada string binary.

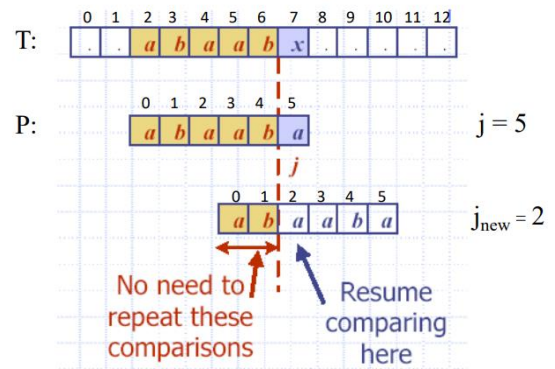
Teks: NOBODY NOTICED HIM
 Pattern: NOT



Gambar 2.1 Ilustrasi Pencocokan String dengan Algoritma Brute Force

C. Algoritma Knuth-Morris-Pratt (KMP)

Algoritma Knuth-Morris-Pratt (KMP) merupakan salah satu algoritma pencocokan string yang melakukan pencarian pola pada suatu teks dari kiri ke kanan, mirip seperti pada algoritma brute force. Hanya saja, metode pencarian pola yang digunakan oleh algoritma KMP lebih cerdas jika dibanding dengan brute force. Algoritma ini dilakukan dengan cara menggeser pola pencarian ketika ditemukan karakter yang tidak cocok untuk menghindari operasi perbandingan yang sia-sia. Misalkan ada suatu teks T dan suatu pattern P, algoritma KMP menyatakan bahwa banyaknya karakter maksimum dari suatu pattern yang dapat digeser ketika ditemukan karakter yang tidak cocok pada indeks j adalah sebanyak panjang prefix terbesar dari $P[0..j-1]$ yang juga merupakan suffix dari $P[1..j-1]$.



Gambar 2.2 Ilustrasi Pencocokan String dengan Algoritma Knuth-Morris-Pratt (KMP)

Dari ilustrasi di atas, dapat dilihat bahwa dengan menggunakan algoritma KMP, ketika ditemui karakter yang tidak cocok dengan pattern, maka pattern akan digeser sejauh prefix terbesar dari $P[0..j-1]$ yang juga merupakan suffix dari $P[1..j-1]$ sehingga tidak ada operasi perbandingan yang sia-sia. Algoritma KMP memiliki kompleksitas waktu yang jauh lebih baik jika dibandingkan dengan brute force yaitu $O(m+n)$. Keuntungan dari penggunaan algoritma KMP adalah algoritma KMP cocok digunakan untuk memproses file berukuran besar yang diakses dari penyimpanan eksternal karena algoritma KMP tidak memerlukan pergerakan mundur dalam implementasinya. Namun di sisi lain, kekurangan dari algoritma KMP adalah algoritma ini semakin mungkin untuk melakukan kesalahan seiring dengan meningkatnya jumlah alfabet.

Untuk mempermudah menentukan prefix terbesar dari $P[0..j-1]$ yang juga merupakan suffix dari $P[1..j-1]$, maka digunakanlah fungsi pinggiran (*border function*). Jika misal terdapat sebuah pattern $P = \text{"abaaba"}$, maka tabel fungsi pinggiran yang akan terbentuk adalah sebagai berikut.

j	0	1	2	3	4	5
P[j]	a	b	a	a	b	a
k	-	0	1	2	3	4
b(k)	-	0	0	1	1	2

Tabel 2.1 Fungsi Pinggiran Pattern P= "abaaba"

D. Algoritma Boyer-Moore (BM)

Algoritma Boyer-Moore merupakan salah satu algoritma pencocokan string yang melakukan pencarian pola pada suatu teks dari kanan ke kiri. Pencocokan string dari kanan ke kiri dilakukan dengan menggunakan teknik looking-glass, sedangkan pergeseran pencocokan pattern P dengan teks T dilakukan dengan menggunakan teknik character-jump.

Teknik looking-glass adalah teknik mencocokkan suatu pattern P dengan teks T dari belakang. Maka apabila ada suatu teks $T[0..m-1]$, pencocokan akan dimulai dari $T[m-1]$ ke arah $T[0]$. Di sisi lain, teknik character-jump adalah teknik yang digunakan untuk melewati karakter yang tidak sesuai pada text hingga ditemukan karakter yang sesuai pada pattern. Untuk mengetahui seberapa jauh harus dilewati, dipakai sebuah fungsi yaitu last occurrence function pada pattern. Dengan menggunakan fungsi tersebut, kemunculan terakhir dari karakter disimpan untuk kemudian digunakan untuk melewati pencocokan. Terdapat tiga kemungkinan kasus character-jump yaitu :

1. Terjadi ketika last occurrence karakter yang mismatch berada di sebelah kiri. Misal indeks penunjuk teks T adalah i , sedangkan indeks penunjuk untuk teks P adalah j . Maka $i_{new} = i + (m-1) - Lo$, sedangkan j disejajarkan dengan i_{new} .
2. Terjadi ketika last occurrence karakter yang mismatch berada di sebelah kanan. Misal indeks penunjuk teks T adalah i , sedangkan indeks penunjuk untuk teks P

adalah j . Maka $i_{new} = i + (m-j)$, sedangkan j disejajarkan dengan i_{new} .

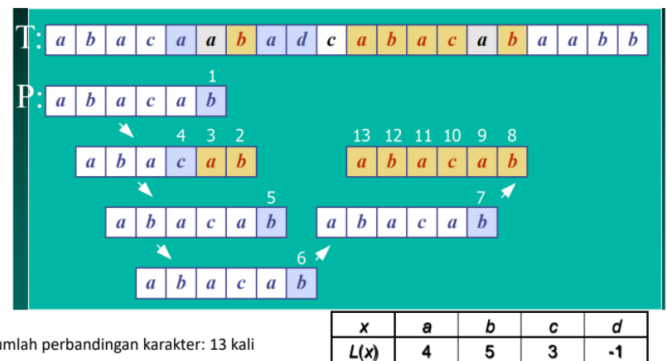
3. Terjadi ketika karakter yang mismatch tidak ada dalam pattern. Misal indeks penunjuk teks T adalah i , sedangkan indeks penunjuk untuk teks P adalah j . Maka $i_{new} = i + m$, sedangkan j disejajarkan dengan i_{new} .

Untuk mempermudah menentukan indeks last occurrence dari suatu karakter, maka digunakanlah fungsi last occurrence. Tabel fungsi last occurrence dibuat dengan cara mencatat seluruh indeks last occurrence pada pattern P dari semua karakter yang ada pada teks T, apabila suatu karakter pada teks T tidak ada di pattern P maka, indeks last occurrence karakter tersebut dinyatakan menjadi -1. Misal terdapat pattern $P = \text{"abacab"}$ dengan himpunan karakter-karakter yang ada pada string S yaitu $A = \{ 'a', 'b', 'c', 'd' \}$. Maka tabel fungsi last occurrence yang terbentuk adalah sebagai berikut :

x	a	b	c	d
L(x)	4	5	3	-1

Tabel 2.2 Fungsi Pinggiran Pattern P= "abaaba"

Kompleksitas waktu terburuk dari algoritma Boyer-Moore adalah $O(nm+A)$. Algoritma Boyer-Moore baik digunakan ketika karakter dalam teks bervariasi dan buruk ketika karakter yang digunakan hanya sedikit seperti kalimat binary. Algoritma Boyer-Moore jauh lebih cepat jika dibandingkan dengan algoritma brute force.



Gambar 2.3 Ilustrasi Pencocokan String dengan Algoritma Boyer-Moore (BM)

E. Malicious Uniform Resource Locator (URL)

Uniform Resource Locator (URL) merupakan rangkaian karakter menurut suatu format standar tertentu, yang digunakan untuk menunjukkan alamat suatu sumber seperti dokumen dan gambar di Internet. Malicious URL atau URL berbahaya merupakan URL yang dibuat dengan tujuan untuk melakukan penipuan dan serangan cyber. Dengan mengklik URL yang terinfeksi, pengguna dapat mengunduh ransomware, virus, trojan, atau jenis malware lainnya yang akan membahayakan perangkat hingga jaringan perangkat penggunanya. URL berbahaya juga dapat digunakan untuk mendapatkan informasi sensitif pengunjungnya di situs palsu. Metode penipuan dengan URL berbahaya yang paling umum adalah dengan menggunakan spam dan phishing. Phishing merupakan jenis

penipuan yang digunakan oleh penjahat yang mencoba menipu korban dengan menyamar sebagai organisasi atau orang terkenal dan tepercaya. Hal ini berarti, seseorang mungkin menerima URL berbahaya melalui email dari temannya jika akun email milik temannya telah disusupi. Tautan berbahaya juga dapat disembunyikan di tautan unduhan yang seharusnya aman dan dapat menyebar dengan cepat melalui berbagi file dan pesan di jaringan berbagi.

Kemudian, malware sendiri adalah seluruh perangkat lunak yang sengaja dirancang untuk menyebabkan kerusakan pada komputer, peladen, klien, atau jaringan computer. Berikut ini adalah beberapa contoh kategori malware yang umum disebarakan melalui URL berbahaya.

Jenis Malware	Penjelasan
Ransomware	Ransomware merupakan sebuah nama dari kelas malware yang terdiri dari dua kata, ransom (tebusan) dan malware, yang bertujuan untuk menuntut pembayaran untuk data / informasi pribadi yang telah dicuri, atau data yang aksesnya dibatasi (enkripsi).
Trojan	Trojan adalah perangkat lunak yang terlihat sah tetapi menjalankan fungsi yang berbahaya. Malware ini sengaja didesain agar terlihat tidak berbahaya untuk menipu orang-orang sehingga mereka mau untuk mengunduhnya. Biasanya Trojan akan menyamar dalam bentuk perangkat lunak gratis, video atau musik, antivirus palsu, atau iklan yang terlihat sah. Perangkat berbahaya ini dirancang untuk merusak, mengganggu, mencuri, atau secara umum menimbulkan beberapa tindakan berbahaya lainnya pada data atau jaringan pengguna. Trojan bertindak seperti aplikasi atau file yang kredibel untuk menipu pengguna.
Virus	Virus komputer merupakan program komputer yang dapat menggandakan atau menyalin dirinya sendiri dan menyebar dengan cara menyisipkan salinan dirinya ke dalam program atau dokumen lain. Virus komputer dapat merusak (misalnya dengan merusak data pada dokumen), membuat pengguna komputer merasa terganggu, maupun tidak menimbulkan efek sama sekali. Virus komputer umumnya dapat merusak perangkat lunak komputer dan tidak dapat secara langsung merusak perangkat keras komputer tetapi dapat mengakibatkan kerusakan dengan cara memuat program yang memaksa over process ke perangkat tertentu.

Tabel 2.3 Contoh Jenis Malware

III. PEMBAHASAN

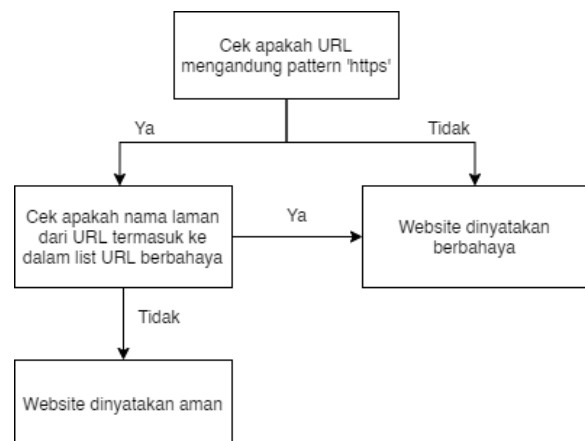
URL berbahaya memang banyak bertebaran di Internet, namun hal ini bukan berarti tidak ada cara untuk mencegah pengunjungan URL berbahaya. Salah satu cara untuk mengecek keamanan dari suatu URL adalah dengan melihat protocol yang digunakan oleh URL tersebut. Protokol yang umumnya digunakan oleh suatu URL adalah Hypertext Transfer Protocol (HTTP) dan Hypertext Transfer Protocol Secure (HTTPS). Perbedaan utama dari kedua protokol ini adalah, HTTPS lebih aman dibanding dengan HTTP karena protokol HTTPS mengenkripsi data yang dikirimkan dari browser menuju server sehingga data pengunjung tidak mudah untuk disalahgunakan. Oleh karena itu, ciri pertama dari URL yang tidak aman adalah penggunaan protokol HTTP pada URL tersebut.

Namun, penggunaan protokol HTTPS juga belum tentu menjamin keamanan dari suatu URL karena nyatanya banyak URL yang sudah berprotokol HTTPS dan tetap dikategorikan sebagai URL yang berbahaya. Oleh karena itu dibutuhkan informasi tambahan mengenai URL mana saja yang sudah berprotokol HTTPS namun berkategori berbahaya. Informasi tambahan ini penulis peroleh dari situs <https://db.aa419.org/fakebankslis.php>. Berikut ini adalah beberapa contoh URL yang sudah berprotokol HTTPS namun masih dinyatakan sebagai URL yang berbahaya. Penulis hanya menggunakan data ini dalam implementasi kode.

nama laman URL	
tkmalinoshop	parcel-supply
gulfassuredfinance	consulting-trans
trustedunity	horizonexpresscargo
freelottoukpromo	metro-alliance
skypetco	globalroyalplace
kittenagency	shipworldexpress

Tabel 3.1 Tabel Nama Laman URL Berbahaya

Berdasarkan kedua informasi tersebut, maka algoritma yang akan digunakan untuk pengecekan keamanan URL pada tulisan ini adalah sebagai berikut :



Gambar 3.1 Flowchart Algoritma Pengecekan

Berikut ini adalah realisasi kode program dalam bahasa Python dari algoritma yang sudah dijabarkan sebelumnya :

1. Fungsi last occurrence dari suatu string

```
def LO(string):  
    list_lo = [-1 for i in range(256)]  
  
    for i in range(len(string)):  
        list_lo[ord(string[i])] = i  
  
    return list_lo
```

2. Fungsi algoritma Boyer-Moore (Sumber : Website Pak Rinaldi Munir dengan modifikasi penulis)

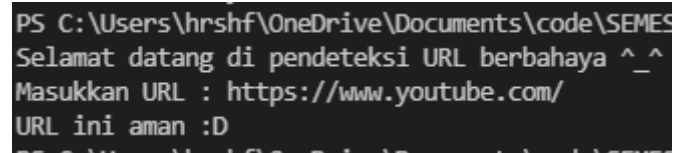
```
def BM(teks, pattern):  
  
    m = len(pattern)  
    n = len(teks)  
    i = m - 1  
    list_lo = LO(teks)  
  
    if i > n-1:  
        return False #pattern tidak ditemukan  
  
    j = m-1  
  
    isValid = True  
    while isValid:  
        if pattern[j]==teks[i]:  
            if j==0:  
                return True #pattern ditemukan  
            else:  
                i = i - 1  
                j = j - 1  
        else:  
            lo = list_lo[ord(teks[i])] + 1  
            i = i + m - min(j, lo+1)  
            j = m-1  
            isValid = (i <= n-1)  
    return False #pattern tidak ditemukan
```

3. Fungsi untuk mengecek apakah suatu URL berbahaya atau tidak

```
def detectMaliciousURL():  
  
    url = url.lower()  
    if BM(url, "https")==False:  
        print("URL ini berbahaya")  
    else:  
        isMalicious = False  
        for mal in malicious:  
            if BM(url, mal)==True:  
                print("URL ini berbahaya")
```

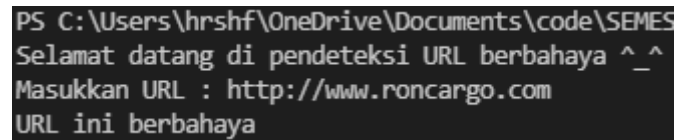
```
isMalicious = True  
break  
if isMalicious==False:  
    print("URL ini aman :D")
```

Berikut ini adalah hasil pengujian program pengecek URL berbahaya :



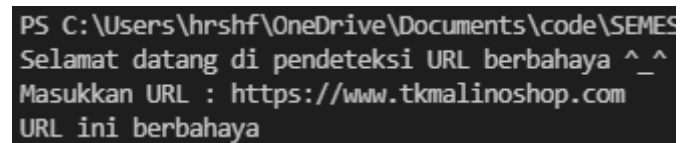
```
PS C:\Users\hrshf\OneDrive\Documents\code\SEMES  
Selamat datang di pendeteksi URL berbahaya ^_^  
Masukkan URL : https://www.youtube.com/  
URL ini aman :D
```

Gambar 3.2 Hasil Kasus Uji URL berprotokol HTTPS dan tidak masuk ke dalam list URL berbahaya



```
PS C:\Users\hrshf\OneDrive\Documents\code\SEMES  
Selamat datang di pendeteksi URL berbahaya ^_^  
Masukkan URL : http://www.roncargo.com  
URL ini berbahaya
```

Gambar 3.3 Hasil Kasus Uji URL berprotokol HTTP



```
PS C:\Users\hrshf\OneDrive\Documents\code\SEMES  
Selamat datang di pendeteksi URL berbahaya ^_^  
Masukkan URL : https://www.tkmalinoshop.com  
URL ini berbahaya
```

Gambar 3.4 Hasil Kasus Uji URL berprotokol HTTPS dan masuk ke dalam list URL berbahaya

IV. KESIMPULAN

Kemajuan teknologi bukanlah sesuatu yang bisa manusia hindari. Walaupun kemajuan teknologi membawa banyak manfaat bagi manusia, namun di sisi lain kemajuan teknologi juga membawa banyak ancaman bagi manusia. Salah satu ancaman yang timbul akibat berkembang pesatnya teknologi Internet adalah maraknya penyebaran URL berbahaya melalui platform seperti media sosial. Pengklikan URL berbahaya ini dapat mengakibatkan terunduhnya malware pada perangkat pengguna yang tentu sangat berbahaya. URL yang aman memiliki ciri berprotokol HTTPS, sedangkan URL yang tidak aman akan memiliki protokol HTTP. Kemudian nama-nama laman URL yang berbahaya pun sudah banyak dibuat blacklist agar pengguna internet dapat membedakan URL mana yang aman dan yang tidak. Namun sayangnya, masih ada orang-orang yang awam terhadap hal ini. Oleh karena itu, otomatisasi pengecekan keamanan URL dapat menjadi solusi dari permasalahan ini.

V. PENUTUP

Segala puji bagi Allah SWT yang telah senantiasa memberikan penulis kesehatan sehingga penulis pun dapat menyelesaikan tulisan ini dengan tepat waktu. Penulis juga mengucapkan terima kasih sebesar-besarnya pada Bapak Prof. Ir. Dwi Hendratmo Widyantoro, M.Sc., Ph.D. selaku dosen pembimbing mata kuliah IF2211 Strategi Algoritma Kelas K3 yang telah memberikan ilmu yang sangat bermanfaat kepada penulis sehingga penulis pun mampu menyusun tulisan ini. Tak

lupa, penulis juga mengucapkan terima kasih kepada teman-teman penulis yang senantiasa memberikan dukungan secara moral kepada penulis sehingga penulis selalu berada dalam kondisi penuh semangat ketika menulis tulisan ini. Perlu penulis akui, tulisan ini masih belum sempurna, penulis tulisan ini dapat dikembangkan lebih jauh lagi kelak.

VIDEO LINK AT YOUTUBE

<https://youtu.be/B2IscXIOdKg>

REFERENCES

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>
- [2] https://www.niagahoster.co.id/blog/perbedaan-http-dan-https/#Perbedaan_HTTP_dan_HTTPS
- [3] <https://gatefy.com/blog/what-malicious-url/>
- [4] Darma, Jarot S., Shenia A, Buku Pintar Menguasai Internet, halaman 416. mediakita.

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, 11 Mei 2021



Hera Shafira
13519131