# UNIFORM RESOURCE LOCATOR (URL) DETECTION SECURITY SYSTEM BASED ON ANDROID

**Haruno Sajati[1], Harliyus Agustian[2], Eko Murdiansyah[3]**
Program Studi Teknik Informatika
Sekolah Tinggi Teknologi Adisutjipto
Jl. Janti Blok-R Lanud Adisutjipto Yogyakarta
Email : [1]masjati.stta@gmail.com, [2]h4rliyus@gmail.com, [3]ekomurdiansyah89@gmail.com

**Abstract**

*The use of the internet, which is currently increasing dramatically, certainly brings the convenience of finding information. The increase in internet usage also eventually gave rise to cybercrime crimes, one of which was by spreading a URL or fake site to steal someone's personal data. The research done is how to build an Android-based application that can detect the security of a URL. The goal is that internet users, especially social media, can avoid cybercrime crime that wants to steal personal data. Making an application uses the Regular Expression method to analyze each line of the Webpage Source Code in the URL based on 8 criteria taken from the World Wide Web Consortium (W3C). The application was then tested with 10 phishing-charged URLs and compared with Kaspersky, McAfee, and AdBlock applications. Based on the results of trials and comparisons, applications that have been made are able to detect 6 or 60% of the 10 URLs. Kaspersky and McAfee applications can detect 70%, while AdBlock only detects 3 or 30% of 10 URLs that contain phishing.*
***Keywords**: Internet, Cybercrime, URL*

## 1. Latar Belakang

Di era yang modern sekarang ini, orang-orang tidak dapat terlepas dari *internet* dan *gadget.* Seakan sudah menjadi kebutuhan pokok, kebanyakan dari orang-orang berlomba untuk memperbanyak akun sosial media mereka, dari yang hanya untuk melihat informasi-informasi terbaru yang ada, hingga yang ingin mencari kepopuleran melalui aplikasi-aplikasi seperti Facebook, Twitter, Instagram, Snapchat, dan masih banyak lagi. Peningkatan dalam penggunaan *internet* juga yang akhirnya melahirkan kejahatan *cybercrime*, salah satuya dengan cara menyebarkan URL atau situs palsu untuk mencuri data-data pribadi seseorang. Penelitian yang dikerjakan adalah membangun sebuah aplikasi berbasis Android yang mampu mendeteksi keamanan dari sebuah URL. Tujuannya agar para pengguna *internet* khususnya sosial media terhindar dari kejahatan *cybercrime* yang ingin mencuri data-data pribadi.

Sebuah penelitian [1] mengatakan bahwa dalam upaya untuk mendeteksi sebuah situs *phishing* bisa dilakukan dengan mengambil karakteristik sebuah *web* yang sah dari standar *World Wide Web Consortium* (W3C), dengan demikian kita dapat menemukan beberapa kriteria dari sebuah situs *phishing*, dan diterapkan kedalam sebuah aplikasi untuk mendeteksi sebuah situs *phishing. Phishing* adalah sebuah tindakan kriminal dengan tujuan mencari informasi pribadi orang lain menggunakan entitas elektronik, salah satunya adalah *website*, dikategorikan sebagai *website phishing* apabila memenuhi karakteristik *phishing.* Phishing dikelompokan menjadi empat kelompok yaitu, *Address Bar Based Feature*, *Abnormal Based Feature, HTML and Javascript Based Feature* dan *Domain Based Feature* [4]. Penelitian selanjutnya [2] juga mengatakan bahwa dalam mendeteksi sebuah situs *phishing* bisa dilakukan dengan metode lain yaitu *Corelation Based Feature Selection* dan diterapkan menggunakan *Binary Logistic Regression* untuk meminimalisir waktu komputasi yang dibutuhkan dalam mendeteksi situs *phishing.*

## 2. Metodologi Penelitian

### 2.1. Analisa Kebutuhan Sistem

Pada pembuatan aplikasi digunakan teknologi dan metode yang digunakan untuk membangun system ini. *Android* adalah sistem operasi berbasis Linux untuk telepon seluler, misal smartphone atau tablet. Para pengembang dapat membuat aplikasi mereka sendiri karena *Android* merupakan *platform* yang terbuka untuk digunakan diberbagai macam *smartphone* [3]. Pemanfaatan *Android* untuk menguji coba aplikasi yang dibuat dengan memanfaatkan *Regular Expression* atau yang sering disebut *regex* adalah sebuah formula untuk pencarian pola suatu kalimat atau *string*. Pada level-level tertentu misal pada level rendah *regex* mampu mencari penggalan kata dalam *string*, dan bila pada level tinggi *regex* juga mampu mengontrol data seperti mencari, menghapus, dan merubah data *string* [5]. Teknik *regex* diterapkan pada *Webpage Source Code* untuk mengetahui suatu halaman website memiliki *phishing*. *Webpage Source Code* merupakan sekumpulan kode pembentuk sebuah *website* yang didalamnya berupa gambar dan teks yang mana akan dialamatkan menjadi sebuah URL agar dapat diakses dan ditampilkan sebagai halaman *web.* Bila melihat dari standar yang terdapat pada *World Wide Web Consortium* (W3C), maka akan ditemukan struktur kode-kode penyusun website yang sudah dinyatakan sah dan aman, namun apabila didalam *Webpage Source Code* tersebut terdapat beberapa kode yang asing, maka URL tersebut dinyatakan tidak sah atau tidak aman[1].

### 2.2. Analisa Kebutuhan Data

Data merupakan aspek penting yang dibutuhkan dalam proses penelitian ataupun pembuatan suatu aplikasi. Kebutuhan akan suatu data ini dapat dipenuhi dengan cara melakukan studi literatur, dan juga wawancara dengan narasumber yang berkaitan dengan apa yang akan diteliti. Beberapa kebutuhan data didapat dari berbagai jurnal penelitian untuk kebutuhan penelitian ini antara lain:

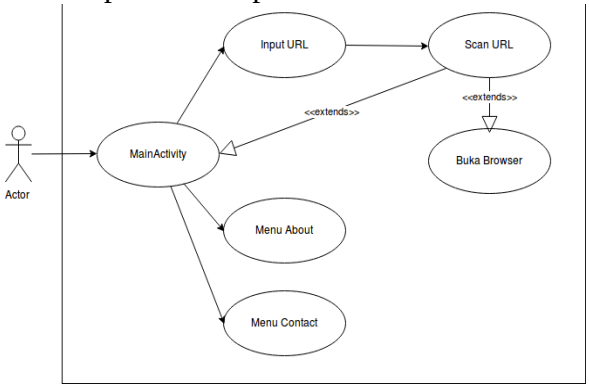
1. Kriteria penilaian URL dari W3C untuk mengecek *Webpage Source Code* adalah *Https, Eksternal Images, Suspicious Url, Domaintag, Iframe, Suspicious Script, Popup Window* [1].
2. Aturan atau *rule* dalam penggunaan *regex* untuk mencocokan *Webpage Source Code* dari sebuah URL dengan *pattern* yang telah dibuat salah satunya terdapat pada pengecekan *Domaintag*.

```
boolean has_ip() {
      String _url = url.getUrl();
      Pattern p = Pattern.compile("(?:(?:25[0-5]|2[0-
4][0-9]|[01]?
      [0-9][0-9]?)\\.){3}(?:25[0-5]|2[0-4][0-
9]|[01]?[0-9][0-9]?)");
      Matcher m = p.mathcer(_url);
      return m.find();
  }
```

Pengujian aplikasi juga memerlukan sebuah data yaitu URL, dan dengan ini penulis mengumpulkan beberapa URL yang didapat dari *website Phistank (https://www.phishtank.com/)* dan juga URL *phishing* yang sengaja dibuat oleh penulis. *Phistank* adalah sebuah *website* yang mana berfungsi untuk mendeteksi *phshing* dalam URL dan *website* tersebut juga memiliki banyak URL yang sudah dinyatakan sebagai *phishing*.

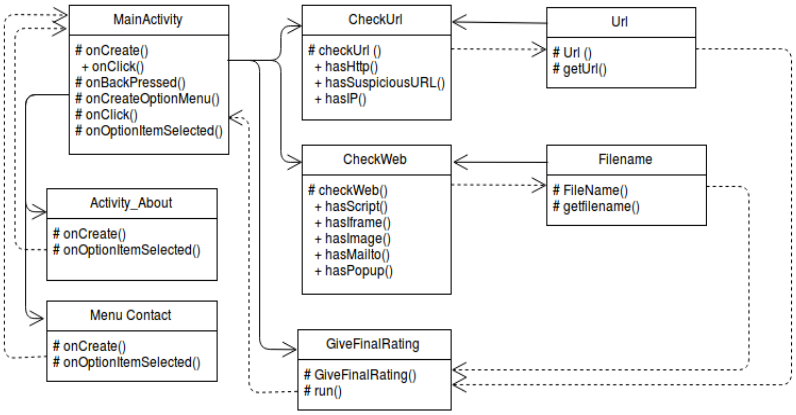
## 2.3 Perancangan Sistem

*Use case* adalah abstraksi dari interaksi antara *system* dan *actor/user*. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *actor* sebuah *system* dengan sistemnya sendiri melalui sebuah proses bagaimana sebuah sistem dipakai. Gambar 1 adalah diagram *use case* yang digunakan dalam pembuatan penelitian ini.



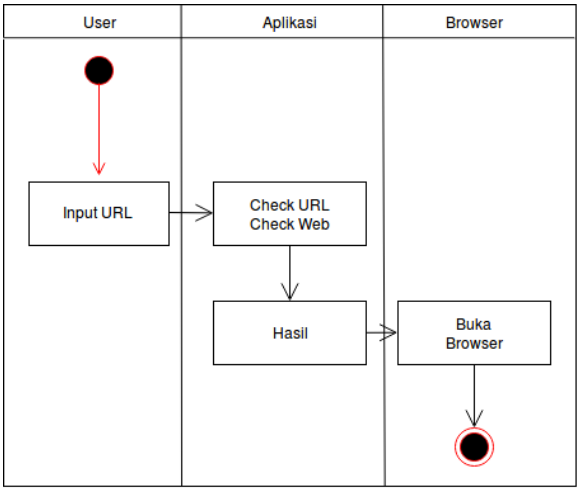

Gambar 1. *Use Case* Url Scanner

*Class diagram*, lihat gambar 2, merupakan sebuah bentuk deskripsi kelompok obyek-obyek dengan *property* dan reaksi yang sama. Berikut adalah class diagram yang digunakan pada pembuatan aplikasi ini



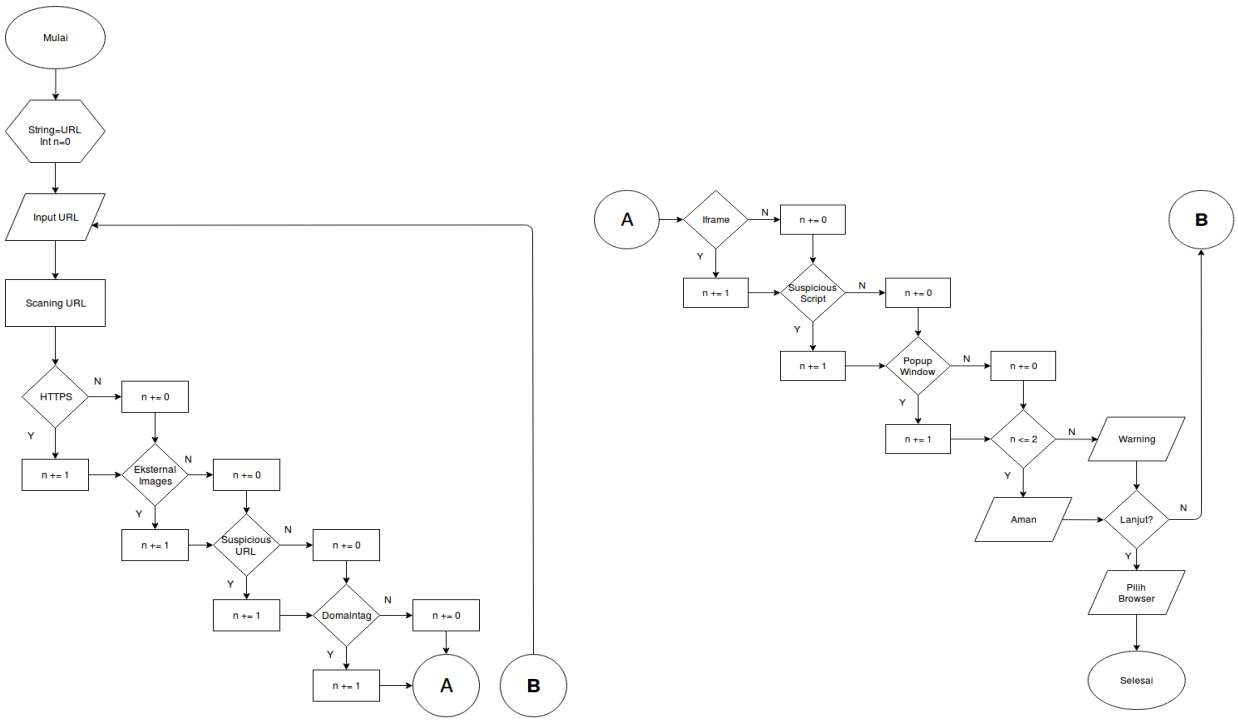

Gambar 2. *Class Diagram* Url Scanner

*Activity diagram* merupakan pendeskripsian logika prosedural dan aliran kerja dalam tahap perancangan. *Activity diagram* menggambarkan alur aktivitas dalam *system* yang sedang dirancang. Berikut adalah *activity diagram* yang digunakan pada pembuatan aplikasi ini dapat dilihat pada gambar 3.




Gambar 3. *Activity Diagram* Url Scanner

Berikut adalah *flowchart system* dari pembuatan aplikasi ini dapat dilihat pada Gambar 4.




Gambar 4. *Flowchart System* Url Scanner

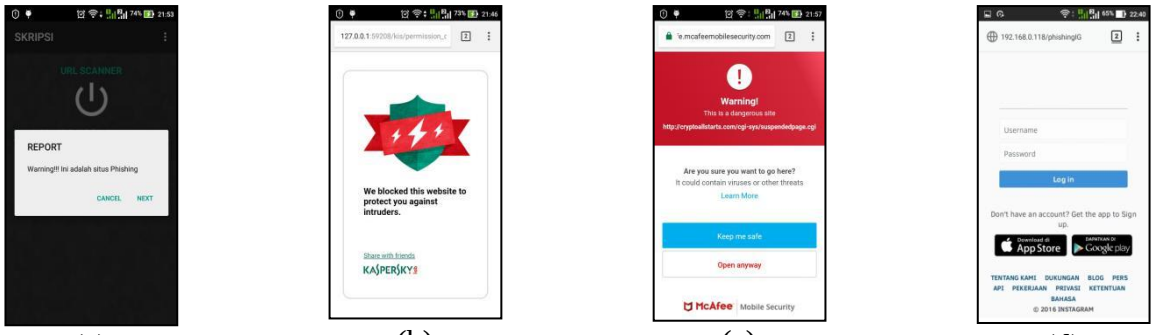
## 3. Hasil dan Pembahasan

### 3.1. Pengujian

Pengujian dilakukan dengan melakukan perbandingan kinerja antara aplikasi Url Scanner dengan Kaspersky, McAfee dan AdBlock. Tahapan pengujian menggunakan 10 URL yang mengandung *phishing*, lalu masing-masing URL akan diakses langsung menggunakan aplikasi Url Scanner dan *mobile browser* yang sudah terkonfigurasi dengan Kaspersky, McAfee, dan AdBlock.

### 3.2. Tampilan *Input*

. Berikut adalah contoh pengujian menggunakan salah satu URL yang terdapat pada Tabel 1.

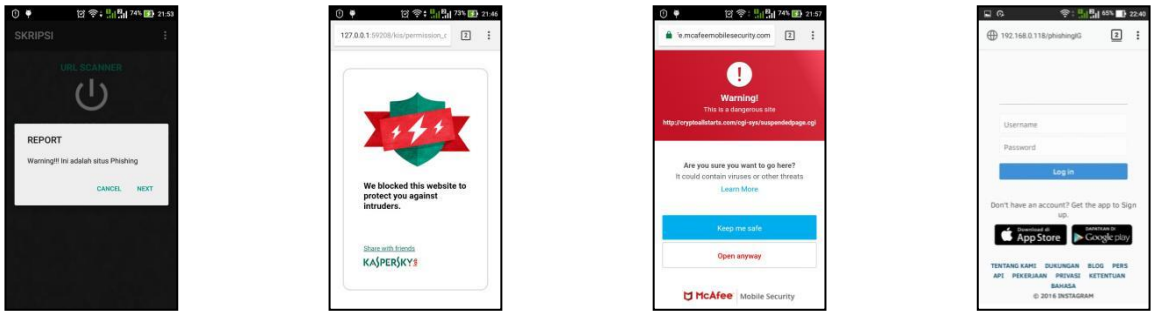



(a) (b) (c) (d)

Gambar 5 Proses *Input* URL.

Gambar 5. (a),(b) merupakan proses *input* Url Scanner dari Gmail dan WhatsApp, Gambar 5.(c) merukapan proses *input* Kaspersky dan McAfee dari Gmail, Gambar 5.(d) merupakan proses *input* AdBlock dari WhatsApp)

### 3.3. Tampilan *Output*




(a) (b) (c) (d)

Gambar 6. *Output* Pengujian URL

Gambar 6. (a) *output* dari Url Scanner, (b) *output* dari Kaspersky, (c) *output* dari McAfee, (d) *output* dari AdBlock)

*Output* yang diperoleh kemudian dikumpulkan pada sebuah tabel dan dianalisa untuk mendapatkan total hasil dari masing-masing aplikasi dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian

| No. | URL | Aplikasi | Hasil |
|---|---|---|---|
| 1. | http://cryptoallstarts.com/wp-includes/pomo/NAVY | Url Scanner | B |
| | | Kaspersky | W |
| | | McAfee | W |
| | | AdBlock | W |

| | | | |
|---|---|---|---|
| 2. | http://cryptoallstarts.com/wp-includes/pomo/NAVY/que.php | Url Scanner | B |
| | | Kaspersky | W |
| | | McAfee | W |
| | | AdBlock | W |
| 3. | http://cryptoallstarts.com/wp-includes/pomo/NAVY/full.php | Url Scanner | B |
| | | Kaspersky | W |
| | | McAfee | W |
| | | AdBlock | W |
| 4. | http://www.seanfeitoakes.com/wp/ www.loginalibaba.com/alibaba/alibaba/ login.alibaba.com.php?email=pro-ricky@outook.com %5Cr%5Cn | Url Scanner | W |
| | | Kaspersky | W |
| | | McAfee | W |
| | | AdBlock | B |
| 5. | http://192.168.0.118/phishingIG/ | Url Scanner | W |
| | | Kaspersky | B |
| | | McAfee | B |
| | | AdBlock | B |
| 6. | http://webdicomviewer.com/ 705646f811ef8210c0fc91afbfd6eb45/ | Url Scanner | W |
| | | Kaspersky | B |
| | | McAfee | W |
| | | AdBlock | B |
| 7. | http://192.168.0.118/facebook/ | Url Scanner | W |
| | | Kaspersky | B |
| | | McAfee | B |
| | | AdBlock | B |
| 8. | http://utm.io/ub8Jo | Url Scanner | B |
| | | Kaspersky | W |
| | | McAfee | B |
| | | AdBlock | B |
| 9. | http://smartpropertiesmy.com/wealt/docusign-lock/ index.php | Url Scanner | W |
| | | Kaspersky | W |
| | | McAfee | W |
| | | AdBlock | B |
| 10. | http://larayasociados.com.mx/atz/l0ginn | Url Scanner | W |
| | | Kaspersky | W |
| | | McAfee | W |
| | | AdBlock | B |

Keterangan: *Warning* (W), Bukan *Phishing* (B).

### 3.4. Analisa Hasil Pengujian

Tabel 2 menunjukkan hasil pengujian dalam bentuk persentase yang telah didapatkan dari hasil perhitungan berdasarkan kriteria penilaian *Warning* (W), Bukan *Phishing* (B). Dari tabel diatas terlihat bahwa aplikasi Kaspersky dan McAfee memiliki persentase yang lebih besar terhadap sepuluh URL yang mengandung *phishing* dan konten ilegal dengan jumlah 70%, kemudian aplikasi Url Scanner memiliki jumlah persentase 60% dari uji coba menggunakan sepuluh URL *phishing.* AdBlock memiliki persentase yang jauh lebih kecil terhadap URL yang mengandung *phishing* dan konten ilegal yaitu 30%.

Tabel 2. Analisa Hasil Pengujian

| No. | Aplikasi | Kriteria | |
|---|---|---|---|
| | | W | B |
| 1. | Url Scanner | 60% | 40% |
| 2. | Kaspersky | 70% | 30% |
| 3. | McAfee | 70% | 30% |
| 4. | AdBlock | 30% | 70% |

## 4. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan pada penelitianini dapat disimpulkan bahwa:

1. Pengujian yang mengacu pada pengecekan *Webpage Source Code* dari sepuluh URL menggunakan metode *Regular Expression*, dapat disimpulkan bahwa metode ini cukup efektif untuk digunakan dalam mendeteksi muatan *phishing* dan konten ilegal didalam URL.
2. Aplikasi Url Scanner mampu mendeteksi 60% dari sepuluh URL yang mengandung *phishing*. Kaspersky dan McAfee yang mendapat total hasil 70%, yang berarti aplikasi Url Scanner cukup bagus digunakan untuk mendeteksi URL yang mengandung *phishing* dan konten ilegal, sedangkan pada AdBlock mendapatkan total hasil 30% dan berarti aplikasi ini kurang cocok digunakan untuk mendeteksi *phishing* didalam URL.

## Daftar Pustaka

[1] Alkhozae, M. G., & Batarfi, O. A. (2011). Phishing websites detection based on phishing characteristics in the webpage source code. International Journal of Information and Communication Technology Research, 1(6).

[2] Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Intelligent rule-based phishing websites classification. IET Information Security, 8(3), 153-160.

[3] Horton, J. (2015). Android Programming for Beginners. Packt Publishing Ltd.

[4] Kuchling, A. M. (2014). Regular Expression HOWTO. Regular Expression HOWTO—Python, 2(10).

[5] Sudaryanto, S. (2018, November). Implementation Port Security for Security Systems Network at the Computing Laboratory of Adisutjipto College of Technology. In Conference SENATIK STT Adisutjipto Yogyakarta (Vol. 4).

[6] Setiawan, Y., Sajati, H., & Sudibya, B. (2013). Perancangan Keamanan Sistem Menggunakan Algoritma Honeypot pada Aplikasi Krs Online (Studi Kasus: Sekolah Tinggi Teknologi Adisutjipto). Compiler, 2(2).

[7] Zonggonau, K., & Sajati, H. (2015). Membangun Sistem Keamanan ARP Spoofing Memanfaatkan Arpwatch dan Addons Firefox. Compiler, 4(1).

[8] Hamka, C. A., Sajati, H., & Indrianingsih, Y. (2014). Sistem Keamanan Jail Bash Untuk Mengamankan Akun Legal Dari Kejahatan Internet MenggunakanThc-hydra. Compiler, 3(1).

[9] Suhendar, A. S. S., Sajati, H., & Astuti, Y. (2013). Perancangan Algoritma Anggi (Aa) dengan Memanfaatkan Diffie-hellman dan Ronald Rivest(Rc4) untuk Membangun Sistem Keamanan Berbasis Port Knocking. Compiler,2(2).

[10] Widodo, A. P., Sarwoko, E. A., Suharto, E., & Siahaan, J. F. O. (2016). Pengamanan Data Foto Pada Perangkat Os Android Menggunakan Teknikkriptografi Hill Cipher. JISKA (Jurnal Informatika Sunan Kalijaga), 1(2).

[11] Rifa'i, A. F. (2016). Sistem Pendeteksi Dan Monitoring Kebocoran Gas(Liquefied Petrolum Gas) Berbasis Internet Of Things.