****

**ALVIN REINALDO**

**RESTU TRIADI**

**ALIF HILMI AKBAR**

**MUHAMMAD FARID MARZUQ**

**AHMAD MAULVI ALFANSURI**

**MODEL KLASIFIKASI SITUS WEB BERBAHAYA MENGGUNAKAN ALGORITME POHON KEPUTUSAN C5.0 UNTUK MENDUKUNG PROGRAM INTERNET CAKAP**

**DEPARTEMEN ILMU KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**INSTITUT PERTANIAN BOGOR**

**BOGOR**

**2018**

# **ABSTRAK**

ALVIN REINALDO, RESTU TRIADI, ALIF HILMI AKBAR, MUHAMMAD FARID MARZUQ, dan AHMAD MAULVI ALFANSURI. Model Klasifikasi Situs Web Menggunakan Algoritme Pohon Keputusan C5.0 untuk Mendukung Program Internet Cakap. Di bawah bimbingan YENI HERDIYENI.

Indonesia adalah salah satu negara dengan pengguna internet terbesar di dunia. Pada tahun 2016, Indonesia menempati peringkat ke-10 dalam urutan negara-negara pengguna internet terbanyak di dunia. Sayangnya, masih banyak pengguna internet yang tidak sadar tentang keamanan di internet. Hal ini tentu dapat membahayakan pengguna internet mengingat kejadian kejahatan siber di Indonesia tertinggi kedua di dunia setelah Jepang. Pada penelitian ini akan dibuat model klasifikasi situs web menggunakan algoritme pohon keputusan C5.0. Algoritme ini dapat mengatasi banyak *instance* dan atribut baik dalam bentuk numerik maupun kategorik. Model klasifikasi tersebut mudah dibaca sehingga mudah dalam mengimplementasikan modelnya untuk menyaring situs web berbahaya. Penelitian ini berhasil mengklasifikasikan situs web berbahaya dan aman menggunakan algoritme pohon keputusan C5.0. Pada penelitian ini dilakukan *encoding* data karena beberapa karakter pada data tidak dapat diterima oleh *package* C50. Model terbaik dari pohon keputusan memiliki tingkat akurasi sebesar 98.32% sehingga penelitian tidak perlu diiterasi.

Kata kunci: C5.0, klasifikasi, pohon keputusan, situs aman, situs berbahaya

# **ABSTRACT**

ALVIN REINALDO, RESTU TRIADI, ALIF HILMI AKBAR, MUHAMMAD FARID MARZUQ, dan AHMAD MAULVI ALFANSURI. Websites Classification Model Using C5.0 Decision Tree Algorithm for Supporting Internet Cakap Program. Supervised by YENI HERDIYENI.

Indonesia is one of the countries with the largest internet users in the world. In 2016 Indonesia was ranked 10th of the most internet user countries in the world. Unfortunately, there are still many internet users who are not aware of security on the internet. This case certainly can endanger internet users considering Indonesia is the second highest occurence of cyber crime in the world after Japan. This research will create malicious websites classification model using the C5.0 decision tree algorithm. The algorithm can handle many instance and attribute in both numerical and categorical types. The classification model is easy to read so it is easy to implement the model to filter malicious websites. This research suceeded classify malicious and benign websites using the C5.0 decision tree algorithm. In this research data encoding is done because some characters in the data cannot accepted by the C50 package. The best model of the decision tree has an accuracy rate of 98.32% so the research does not need to be iterated.

Keywords: C5.0, classification, decision tree, benign website, malicious website

# **DAFTAR ISI**

[DAFTAR ISI ii](#_Toc532925232)

[DAFTAR TABEL iv](#_Toc532925233)

[DAFTAR GAMBAR iv](#_Toc532925234)

[DAFTAR LAMPIRAN iv](#_Toc532925235)

[PENDAHULUAN 1](#_Toc532925236)

[Latar Belakang 1](#_Toc532925237)

[Perumusan Masalah 2](#_Toc532925238)

[Tujuan Penelitian 2](#_Toc532925239)

[Manfaat Penelitian 2](#_Toc532925240)

[Ruang Lingkup Penelitian 2](#_Toc532925241)

[TINJAUAN PUSTAKA 3](#_Toc532925242)

[Situs Web 3](#_Toc532925243)

[Algoritme Pohon Keputusan C5.0 3](#_Toc532925244)

[*K-Fold Cross Validation* 5](#_Toc532925245)

[METODE PENELITIAN 5](#_Toc532925246)

[Data Penelitian 5](#_Toc532925247)

[Tahapan Penelitian 7](#_Toc532925248)

[Pengumpulan Data 7](#_Toc532925249)

[Eksplorasi dan Praproses Data 7](#_Toc532925250)

[Pembagian Data 8](#_Toc532925251)

[Pemodelan Klasifikasi Pohon Keputusan 8](#_Toc532925252)

[Pengujian Model Klasifikasi 8](#_Toc532925253)

[Lingkungan Pengembangan 8](#_Toc532925254)

[HASIL DAN PEMBAHASAN 9](#_Toc532925255)

[Praproses Data 9](#_Toc532925256)

[Pembagian Data 10](#_Toc532925257)

[Pemodelan Klasifikasi Pohon Keputusan 11](#_Toc532925258)

[Pengujian Model Klasifikasi 11](#_Toc532925259)

[SIMPULAN DAN SARAN 12](#_Toc532925260)

[Simpulan 12](#_Toc532925261)

[Saran 12](#_Toc532925262)

[DAFTAR PUSTAKA 13](#_Toc532925263)

LAMPIRAN 15

# **DAFTAR TABEL**

|  |  |  |
| --- | --- | --- |
| 1 | Fitur pada data situs web yang berbahaya dan aman | 6 |
| 2 | Jumlah data untuk masing-masing kelas | 10 |
| 3 | Distribusi kelas data pada data latih dan data uji | 10 |

# **DAFTAR GAMBAR**

|  |  |  |
| --- | --- | --- |
| 1 | Diagram alur tahapan penelitian | 7 |
| 2 | Potongan dataset yang belum dipraproses | 9 |
| 3 | Contoh hasil proses *ordinal encoding* | 10 |
| 4 | Potongan dataset yang telah dipraproses | 10 |
| 5 | Potongan kode program untuk memodelkan pohon keputusan | 11 |
| 6 | Grafik akurasi model klasifikasi | 12 |

# **DAFTAR LAMPIRAN**

|  |  |  |
| --- | --- | --- |
| 1 | Contoh penggunaan algoritme C5.0 pada dataset kecil situs web berbahaya dan aman | 15 |
| 2 | Kode program *ordinal encoding* dalam bahasa pemrograman Python | 16 |
| 3 | Kode program proses pemodelan pohon keputusan secara keseluruhan dalam bahasa pemrograman R | 18 |
| 4 | Model klasifikasi pohon keputusan dengan algoritme C5.0 | 20 |
| 5 | Visualisasi model klasifikasi pohon keputusan | 22 |

# **PENDAHULUAN**

## **Latar Belakang**

Indonesia adalah salah satu negara dengan pengguna internet terbesar di dunia. Menurut data ITU (2018), pada tahun 2016 Indonesia menempati peringkat ke-10 dalam urutan negara-negara pengguna internet terbanyak di dunia. Sedangkan survei yang dilakukan oleh APJII (2017) menyebutkan bahwa pada tahun 2017 pengguna internet di Indonesia sebanyak 143,26 juta orang dengan tingkat penetrasi internet sebesar 54,86%. Sayangnya, masih banyak pengguna internet yang tidak sadar tentang keamanan di internet. Menurut data APJII (2017), masih ada 34,02% pengguna internet yang tidak sadar bahwa data pengguna dapat diambil melalui internet. Selain itu masih ada 16,02% pengguna internet yang tidak sadar bahwa penipuan dapat terjadi melalui internet. Hal ini tentu dapat membahayakan pengguna internet mengingat kejadian kejahatan siber di Indonesia tertinggi kedua di dunia setelah Jepang (Rizki 2018).

Saat ini pemerintah melalui Kementerian Komunikasi dan Informatika (Kemkominfo) telah meluncurkan program Internet Cakap (Cerdas, Kreatif, dan Produktif). Program ini diluncurkan untuk meningkatkan potensi generasi muda untuk lebih cerdas memilih konten, kreatif menciptakan karya baru, serta produktif untuk mendapatkan manfaat (Kemkominfo Ditjen Aptika DPI 2015). Selain Internet Cakap, Kemkominfo juga meluncurkan program Agen Perubahan Informatika (API). API adalah penggerak revolusi mental di bidang informatika yang dimotori oleh Relawan Teknologi Informasi dan Komunikasi (TIK) (Kemkominfo Ditjen Aptika DPI 2018). Relawan TIK dapat menggunakan dan memanfaatkan TIK dan internet secara cerdas, kreatif, dan produktif serta dapat mempromosikan, menularkan, serta memberikan edukasi kepada masyarakat di bidang informatika. Sayangnya, baik program Internet Cakap maupun API, tidak menekankan literasi keamanan internet. Padahal pemahaman mengenai keamanan internet sangat penting terutama bagi pengguna internet awam.

Berdasarkan paparan di atas terlihat bahwa pemahaman masyarakat mengenai keamanan internet masih kurang, sehingga perlu dilakukan upaya agar masyarakat dapat terhindar dari situs-situs yang berbahaya. Beberapa penelitian telah dilakukan dalam mengklasifikasikan situs-situs berbahaya. Xu et al. (2013) melakukan klasifikasi dengan mengambil fitur dari berbagai lapisan situs web, sehingga dapat memperkecil hambatan dalam mendeteksi situs berbahaya. Pada penelitian tersebut didapatkan 105 fitur *application-layer* dengan 4 sub kelas dan 19 fitur *network-layer* dengan 3 sub kelas. Pada penelitian tersebut juga didapat fitur-fitur yang signifikan menggunakan algoritme *principal component analysis* (PCA) serta CfsSubsetEval dan InfoGainAttributeEval pada perangkat lunak Weka. Fitur yang paling sering muncul pada proses seleksi fitur tersebut adalah URL\_Length, HTTPHead\_server, dan Duration. Penelitian lanjutan dilakukan oleh Urcuqui et al. (2017) untuk mengklasifikasikan situs-situs berbahaya. Pada penelitian tersebut juga diseleksi fitur-fitur yang signifikan. Fitur-fitur yang digunakan yaitu URL\_Length, Number\_Special\_Characters, Charset, Server, Content\_Length, Whois\_Country, Whois\_Statepro, Whois\_Regdate, Whois\_Updated\_Date, TCP\_Conversation\_Exchange, Dist\_Remote\_TCP\_Port, Remote\_IPS, App\_Bytes, Source\_App\_Packets, Remote\_App\_Packets, App\_Packets, dan DNS\_Query\_Times. Pada penelitian tersebut dilakukan perbandingan akurasi antara algoritme *support vector machine* (SVM), regresi logistik, naïve Bayes, dan J48 (*package* algoritme C4.5 pada perangkat lunak Weka). Pada penelitian tersebut akurasi tertinggi didapatkan dengan menggunakan algoritme J48.

Algoritme lain yang dapat digunakan dalam proses klasifikasi adalah algoritme pohon keputusan C5.0. Algoritme ini dapat mengatasi banyak *instance* dan variabel baik dalam bentuk numerik maupun kategorik. Model klasifikasi algoritme pohon keputusan C5.0 dapat disajikan dalam bentuk pohon keputusan atau kumpulan aturan *if*-*then* (Munawaroh et al. 2013). Model tersebut mudah dibaca sehingga mudah dalam mengimplementasikan modelnya untuk menyaring situs web berbahaya.

Oleh karena itu, pada penelitian ini akan dibuat model klasifikasi situs web berbahaya. Penelitian ini menggunakan algoritme pohon keputusan C5.0. Hasil penelitian ini diharapkan dapat digunakan untuk mendukung program Internet Cakap yang diluncurkan oleh Kementerian Komunikasi dan Informatika.

## **Perumusan Masalah**

Berdasarkan latar belakang, perumusan masalah dalam penelitian ini adalah: Bagaimana mengklasifikasikan situs web berbahaya dengan menggunakan algoritme pohon keputusan C5.0?

## **Tujuan Penelitian**

Tujuan penelitian ini adalah membuat model klasifikasi situs web berbahaya dengan menerapkan algoritme pohon keputusan C5.0

## **Manfaat Penelitian**

Hasil dari penelitian ini diharapkan dapat digunakan sebagai acuan dalam mengklasifikasikan situs web berbahaya, sehingga dapat mendukung program Internet Cakap dengan membuat internet lebih aman bagi penggunanya.

## **Ruang Lingkup Penelitian**

Ruang lingkup penelitian ini

1. Data diperoleh dari situs Kaggle yang diunggah oleh akun Christian Urcuqui. Data tersebut berjudul “Malicious and benign websites: classify by application and network features”. Data yang digunakan adalah versi ketiga yang diunggah pada tanggal 9 April 2018
2. Implementasi algoritme C5.0 menggunakan *package* C50 yang terdapat pada perangkat lunak RStudio.

# **TINJAUAN PUSTAKA**

## **Situs Web**

Mengacu pada kamus Merriam-Webster (2018), situs web (*website*) adalah sekumpulan halaman *world wide web* (www) yang biasanya mengandung *hyperlink* satu sama lain dan disediakan oleh individu, perusahaan, lembaga pendidikan, pemerintah, atau organisasi. Situs web dapat diakses melalui jaringan *internet protocol* (IP) publik atau *local area network* (LAN) pribadi dengan merujuk ke *uniform resource locator* (URL) yang mengidentifikasikan situs web tersebut. URL biasanya disebut dengan alamat web.

Tidak semua situs web aman untuk diakses. Ada situs web yang aman (*benign website*) dan ada yang berbahaya (*malicious website*). Situs web berbahaya adalah situs yang mencoba memasang *malware* ke dalam perangkat (Symantec Corporation 2018). *Malware* adalah istilah umum untuk operasi apa pun yang dapat mengganggu operasi komputer, mengumpulkan informasi pribadi, atau bahkan mendapatkan akses penuh dari komputer. Situs web berbahaya diperkirakan akan tetap ada di masa mendatang karena sulitnya mencegah sebuah situs web disusupi atau diretas.

## **Algoritme Pohon Keputusan C5.0**

Pohon keputusan adalah salah satu metode untuk mengklasifikasikan data berdasarkan data latih yang kelasnya sudah diketahui, sehingga termasuk ke dalam *supervised learning*. Konsep utama dari algoritme pohon keputusan adalah mengelompokkan data dengan mengetahui heterogenitas data. Heterogenitas data diukur dengan menggunakan konsep entropi sebagai teknik untuk mengukur ketidakpastian (*uncertainty*) atau keteracakan (*randomness*) data. Semakin kecil nilai entropi menunjukkan bahwa keragaman data semakin homogen. Semakin kecil nilai entropi maka akan semakin besar nilai *information gain­*-nya. *Root* dari pohon keputusan ditentukan menggunakan perhitungan nilai *information* *gain* yang paling tinggi, begitu juga dengan *node*-*node* selanjutnya.

Salah satu algoritme dalam pohon keputusan adalah algoritme C5.0. Algoritme C5.0 merupakan algoritme perbaikan dari algoritme C4.5 dan *iterative dichotomizer* 3 (ID3). Algoritme C5.0 memiliki keunggulan dibanding algoritme C4.5 yaitu dapat menangani klasifikasi dengan data yang berukuran besar. Algoritme C5.0 juga memiliki keunggulan dalam hal kecepatan, efisiensi penggunaan memori, ukuran pohon keputusan, dan kesalahan klasifikasi (Pandya dan Pandya 2015). Algoritme C5.0 dapat menangani atribut baik data numerik maupun data kategorik. Algoritme C5.0 dapat menangani atribut dengan data numerik dengan membuat ambang batas kemudian membagi data menjadi data yang lebih besar dari ambang batas dan kurang dari atau sama dengan ambang batas. Algoritme C5.0 juga dapat menangani *missing value* dengan menandainya sebagai “?” (Pandya dan Pandya 2015).

Parameter yang digunakan dalam membuat pohon keputusan yaitu D (data latih yang telah ditentukan kelasnya), *attribute\_list* (himpunan yang terdiri dari kandidat atribut), dan *attribute\_selection\_method* (prosedur untuk menentukan kriteria pemilihan atribut). Algoritme pohon keputusan (*generate\_decision\_tree*) adalah sebagai berikut:

1. Membuat *node* N,
2. Jika semua *instance* pada D memiliki kelas yang sama (misal kelas C), maka

*node* N sebagai *leaf* dan diberi label kelas C.

1. Jika *attribute\_list* kosong, maka

*node* N sebagai *leaf* dan diberi label nilai kelas terbanyak pada sampel.

1. Menerapkan *attribute\_selection\_method* (D, *attribute\_list*) untuk memperoleh atribut terbaik,
2. Memberi label *node* N dengan atribut terbaik,
3. Jika atribut bernilai diskrit dan pohon yang akan dibuat bukan *binary tree*, maka

*attribute\_list* = *attribute\_list* – atribut terbaik.

1. Untuk setiap nilai j pada hasil atribut terbaik,
   1. Dj menjadi himpunan data D yang memenuhi hasil j
   2. Jika Dj kosong, maka

tambahkan *leaf* yang diberi label nilai kelas terbanyak pada D ke *node* N

* 1. Selainnya tambah cabang baru dengan memanggil fungsi *generate\_decision\_tree* (Dj, *attribute\_list*) ke *node* N.

(Han et al. 2012)

Pada pohon keputusan, *root* merupakan *node* dengan atribut yang memiliki nilai *information gain* paling tinggi, begitu juga dengan *node* selanjutnya. Perhitungan pemilihan atribut didefinisikan pada Persamaan 1. Pada persamaan tersebut digunakan fungsi logaritma berbasis 2 karena informasi yang diolah dalam bit.

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

(Han et al. 2012)

dengan

: informasi yang dibutuhkan untuk mengklasifikasikan label kelas sebuah *instance* pada data latih (entropi),

: jumlah kelas pada data latih ,

: peluang munculnya kelas ke-i pada data latih D.

Atribut pada data latih memiliki atribut yang berbeda {, , ..., }. Atribut dapat digunakan untuk membagi data latih menjadi subhimpunan yang berbeda {, , ..., }, dengan berisi *instance* dari atribut A. Perhitungan untuk mendapatkan nilai entropi yang dihasilkan untuk mengklasifikasi *instance* dari subhimpunan berdasarkan partisi oleh atribut dapat dilihat pada Persamaan 2.

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

(Han et al. 2012)

dengan

: nilai entropi yang dihasilkan untuk mengklasifikasi *instance* dari subhimpunan berdasarkan partisi oleh atribut ,

: jumlah atribut pada data latih ,

: bobot subhimpunan ke-j dari data latih ,

: nilai entropi subhimpunan ke-j dari data latih .

Pemilihan atribut yang akan dijadikan *root* atau *node* menggunakan nilai *information gain*. Atribut dengan nilai *information gain* tertinggi dipilih sebagai *root*. *Node*-*node* selanjutnya juga dipilih berdasarkan *information gain* tertinggi. Untuk mendapatkan nilai *information gain* digunakan Persamaan 3.

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

(Han et al. 2012)

## ***K-Fold Cross Validation***

*Cross-validation* adalah metode yang digunakan untuk mengevaluasi dan membandingkan algoritme *machine* *learning* dengan membagi data menjadi dua bagian. Satu bagian digunakan sebagai data latih dan lainnya digunakan sebagai data uji (Refaeilzadeh et al. 2009). Salah satu algoritme *cross-validation* yang sering digunakan adalah *K-fold cross-validation*. *K-fold cross-validation* membagi data menjadi bagian dengan ukuran yang sama. Dari data tersebut kemudian dilakukan iterasi. Pada setiap iterasi satu *fold* digunakan sebagai data uji dan *fold* lainnya digunakan sebagai data latih. Pada setiap iterasi *fold* yang menjadi data latih dan data uji bergantian sampai setiap *fold* pernah menjadi data uji.

# **METODE PENELITIAN**

## **Data Penelitian**

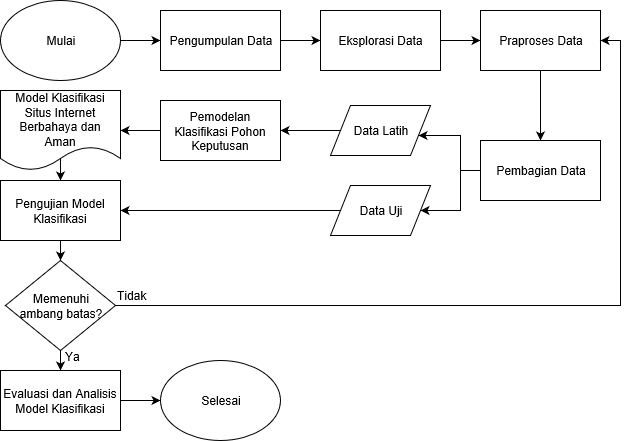
Data yang digunakan dalam penelitian ini adalah data situs web berbahaya dan aman berjudul “Malicious and benign websites: classify by application and network features”. Data yang digunakan adalah versi ketiga yang diunggah oleh akun Christian Urcuqui pada tanggal 9 April 2018. Data tersebut berformat csv. Data tersebut berjumlah 1781 *instance*.Data tersebut terdiri dari 20 fitur ditambah satu fitur kelas (Type). Fitur-fitur pada data situs web yang berbahaya dan aman dapat dilihat pada Tabel 1. Data tersebut merupakan data penelitian yang dilakukan Xu et al. (2013) yang telah melalui proses reduksi fitur pada penelitian yang dilakukan oleh Urcuqui et al. (2017).

Tabel 1 Fitur pada data situs web yang berbahaya dan aman

|  |  |  |
| --- | --- | --- |
| No | Nama Fitur | Keterangan |
| 1 | URL | Identifikasi anonim dari alamat URL yang dianalisis dalam penelitian ini |
| 2 | URL\_LENGTH | Jumlah karakter pada alamat URL |
| 3 | NUMBER\_SPECIAL\_CHARACTERS | Jumlah *special characters* pada alamat URL, misal “?”, “-”, “\_”, “=”, dan “%” |
| 4 | CHARSET | Standar pengkodean karakter (*character set*) yang digunakan |
| 5 | SERVER | Sistem operasi *server* yang diterima dari respons paket |
| 6 | CONTENT\_LENGTH | Ukuran konten dari *header* HTTP |
| 7 | WHOIS\_COUNTRY | Nama negara yang didapat dari respons *server* |
| 8 | WHOIS\_STATEPRO | Nama negara bagian atau provinsi yang didapat dari respons *server* |
| 9 | WHOIS\_REGDATE | Tanggal registrasi *server* yang dianalisis, berformat DD/MM/YYYY HH:MM |
| 10 | WHOIS\_UPDATED\_DATE | Tanggal pembaharuan terakhir dari *server* yang dianalisis, berformat DD/MM/YYYY HH:MM |
| 11 | TCP\_CONVERSATION\_EXCHANGE | Jumlah total paket TCP yang dikirim ke *server* oleh *crawler* |
| 12 | DIST\_REMOTE\_TCP\_PORT | Jumlah total *port* TCP yang berbeda yang digunakan *server* ketika berkomunikasi dengan *crawler* |
| 13 | REMOTE\_IPS | Jumlah total alamat IP yang berbeda yang terhubung oleh *crawler* tidak termasuk alamat IP dari *server* DNS |
| 14 | APP\_BYTES | Jumlah ukuran (dalam Byte) dari data pada *application-layer* yang dikirim oleh *crawler*  ke *server*, tidak termasuk data yang dikirim ke *server* DNS |
| 15 | SOURCE\_APP\_PACKETS | Jumlah paket yang dikirim oleh *crawler* ke *server* |
| 16 | REMOTE\_APP\_PACKETS | Jumlah paket yang dikirim oleh *server* ke *crawler* |
| 17 | SOURCE\_APP\_BYTES | Jumlah ukuran (dalam Byte) dalam komunikasi dari *crawler* ke *server* |
| 18 | REMOTE\_APP\_BYTES | Jumlah ukuran (dalam Byte) dalam komunikasi dari *server* ke *crawler* |
| 19 | APP\_PACKETS | Jumlah total paket IP yang dihasilkan untuk mendapatkan konten yang sesuai dengan *input* URL, termasuk pengalihan dan permintaan DNS |
| 20 | DNS\_QUERY\_TIMES | Jumlah permintaan DNS yang dikirim oleh *crawler* |
| 21 | TYPE | Representasi tipe situs web yang dianalisis, 1 berarti situs web berbahaya dan 0 berarti situs web aman |

## **Tahapan Penelitian**

Pada penelitian ini terdapat beberapa tahapan yaitu pengumpulan data, eksplorasi data, praproses data, pembagian data, pemodelan klasifikasi dengan pohon keputusan, pengujian model klasifikasi, dan evaluasi serta analisis model klasifikasi. Keseluruhan tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1 Diagram alur tahapan penelitian

### **Pengumpulan Data**

Tahap awal penelitian yaitu mengambil data situs web berbahaya dan aman dari situs Kaggle. Data tersebut diunduh pada tanggal 22 April 2018. Data tersebut kemudian disimpan dalam format csv.

### **Eksplorasi dan Praproses Data**

Pada tahap ini data dieksplorasi agar dapat ditentukan praproses data yang perlu dilakukan sebelum data diolah. Setelah dieksplorasi kemudian data dipraproses. Praproses data dilakukan agar data dapat digunakan dengan baik, sehingga dapat meningkatkan akurasi data. Pada penelitian ini praproses yang dilakukan adalah *encoding*. Fungsi *encoder* yang digunakan pada praproses ini adalah *ordinal encoder*. *Ordinal encoder* mengonversi setiap label data kategorik menjadi nilai integer dari 1 sampai sesuai dengan banyaknya label data pada atribut tersebut (Hale J 2018).

### **Pembagian Data**

Sebelum melakukan klasifikasi, data terlebih dahulu dibagi menjadi data latih dan data uji. Data latih digunakan untuk membangun model pohon keputusan, sedangkan data uji digunakan untuk menguji model pohon keputusan. Metode yang digunakan untuk membagi data adalah *K-fold cross-validation*. *K-fold cross-validation* membagi data menjadi bagian. Pada penelitian ini digunakan nilai (Refaeilzadeh et al. 2009).

### **Pemodelan Klasifikasi Pohon Keputusan**

Pemodelan pohon keputusan menggunakan algoritme C5.0. Algoritme C5.0 menyeleksi atribut yang digunakan dengan mencari nilai entropi dan nilai *informatin gain*. Kemudian algoritme C5.0 menggunakan nilai *information gain* untuk memilih *parent* ataupun *node* selanjutnya.

### **Pengujian Model Klasifikasi**

Pengujian model klasifikasi dilakukan dengan menghitung nilai akurasi. Mengacu pada Han et al. (2012), akurasi adalah tingkat kebenaran hasil klasifikasi dibandingkan dengan data kelas sebenarnya. Nilai akurasi dapat diperoleh menggunakan Persamaan 4.

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

(Han et al. 2012)

Ambang batas untuk menguji hasil akurasi dari penelitian sebelumnya yang dilakukan oleh Urcuqui et al. (2017). Pada penelitian tersebut digunakan algoritme *support vector machine* (SVM), regresi logistik, naïve Bayes, dan J48 dengan akurasi berturut-turut adalah 85.46%, 84.51%, 85.46%, dan 96.05%. Pengujian akurasi pada penelitian ini menggunakan nilai rata-rata dari akurasi pada penelitian sebelumnya yaitu sebesar 87.87%.

## **Lingkungan Pengembangan**

Perangkat keras yang digunakan dalam penelitian ini adalah komputer personal dengan spesifikasi sebagai berikut:

1. Prosesor : Intel® Core™ i5-7200U CPU @2.50GHz
2. Memori : 8192 MB RAM
3. VGA : NVIDIA GeForce 940MX

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

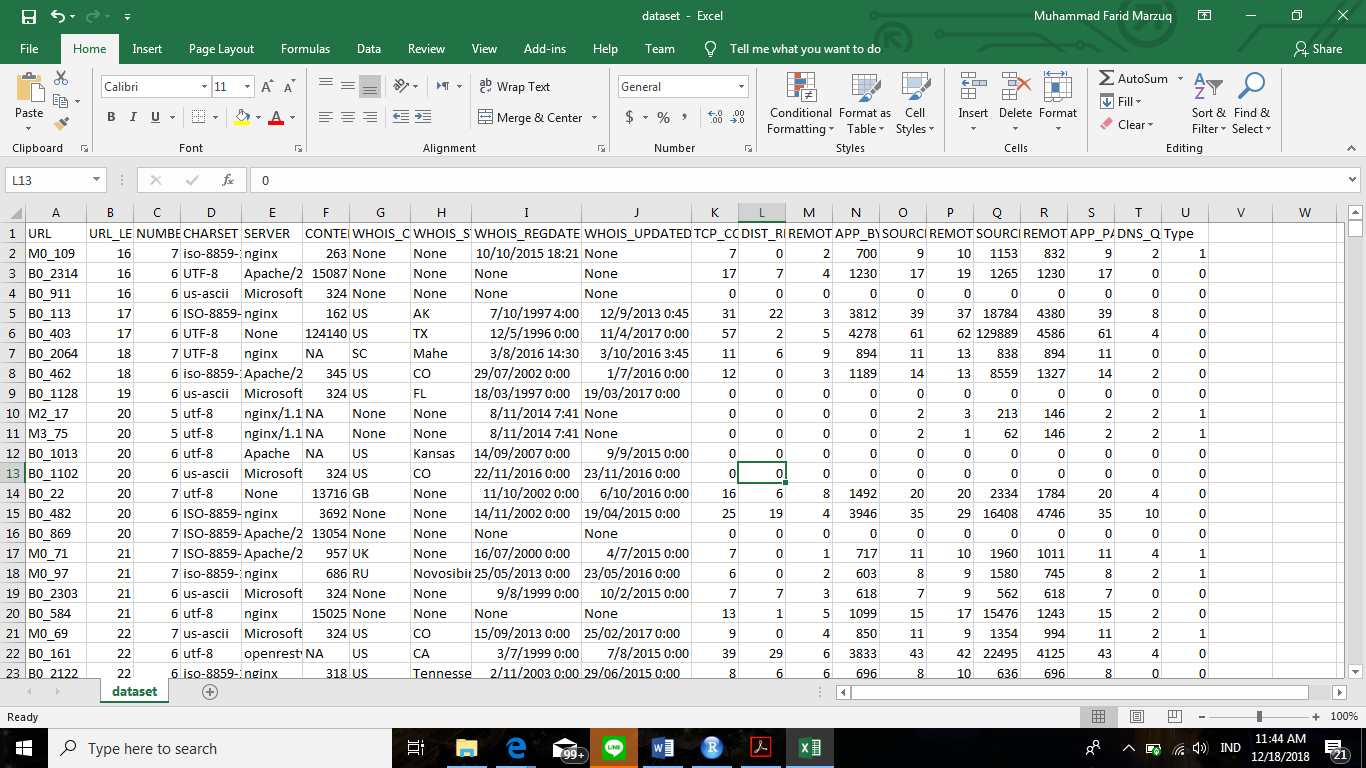
1. Sistem operasi Microsoft Windows 10 (64-bit)
2. Bahasa pemrograman Python versi 3.4.3
3. Bahasa pemrograman R versi 3.5.1
4. Microsoft Excel 2016 untuk eksplorasi data
5. PyCharm Community Edition versi 2018.2.2 untuk proses *encoding* data
6. RStudio 1.1.456 untuk mengolah data keseluruhan

# **HASIL DAN PEMBAHASAN**

Berdasarkan alur tahapan penelitian pada Gambar 1, model klasifikasi dapat dievaluasi jika akurasi model terbaik telah memenuhi ambang batas 87.87%. Percobaan pertama pada penelitian ini telah memenuhi ambang batas tersebut sehingga tidak diperlukan iterasi selanjutnya. Penjelasan lebih lanjut dari tiap tahap dapat dilihat di bawah ini.

## **Praproses Data**

Pada dataset terlihat bahwa banyak *instance* yang memiliki jumlah karakter yang terlalu panjang. Pada dataset juga terlihat banyak mengandung karakter yang tidak dapat diolah oleh *package* C50, seperti tanda dua titik (:) dan tanda titik koma (;). Oleh karena itu atribut-atribut dengan data kategorik perlu dipraproses dengan *encoding*. Pada eksplorasi dataset terlihat juga bahwa atribut URL merupakan atribut identifikasi unik dari data sehingga atribut ini dihilangkan. Gambar 2 menunjukkan potongan data yang belum dipraproses.



Gambar 2 Potongan dataset yang belum dipraproses

Atribut-atribut yang di-*encoding* adalah CHARSET, SERVER, WHOIS\_COUNTRY, WHOIS\_STATEPRO, WHOIS\_REGDATE, dan WHOIS\_UPDATED\_DATE. Implementasi proses *encoding* menggunakan *library* pandas pada bahasa pemrograman Python. Kode program lengkap untuk proses *encoding* dapat dilihat pada Lampiran 2. Contoh hasil proses *encoding* dapat dilihat pada Gambar 2. Setelah data di-*encoding*, hasilnya dikembalikan lagi ke dataset. Tipe data yang terbaca setelah data di-*encoding* berubah menjadi numerik sehingga perlu diubah kembali menjadi tipe data kategorik. Potongan dataset yang telah dipraproses dapat dilihat pada Gambar 3.

'charset'

{0: 'ISO-8859',

1: 'ISO-8859-1',

2: 'None',

3: 'UTF-8',

4: 'iso-8859-1',

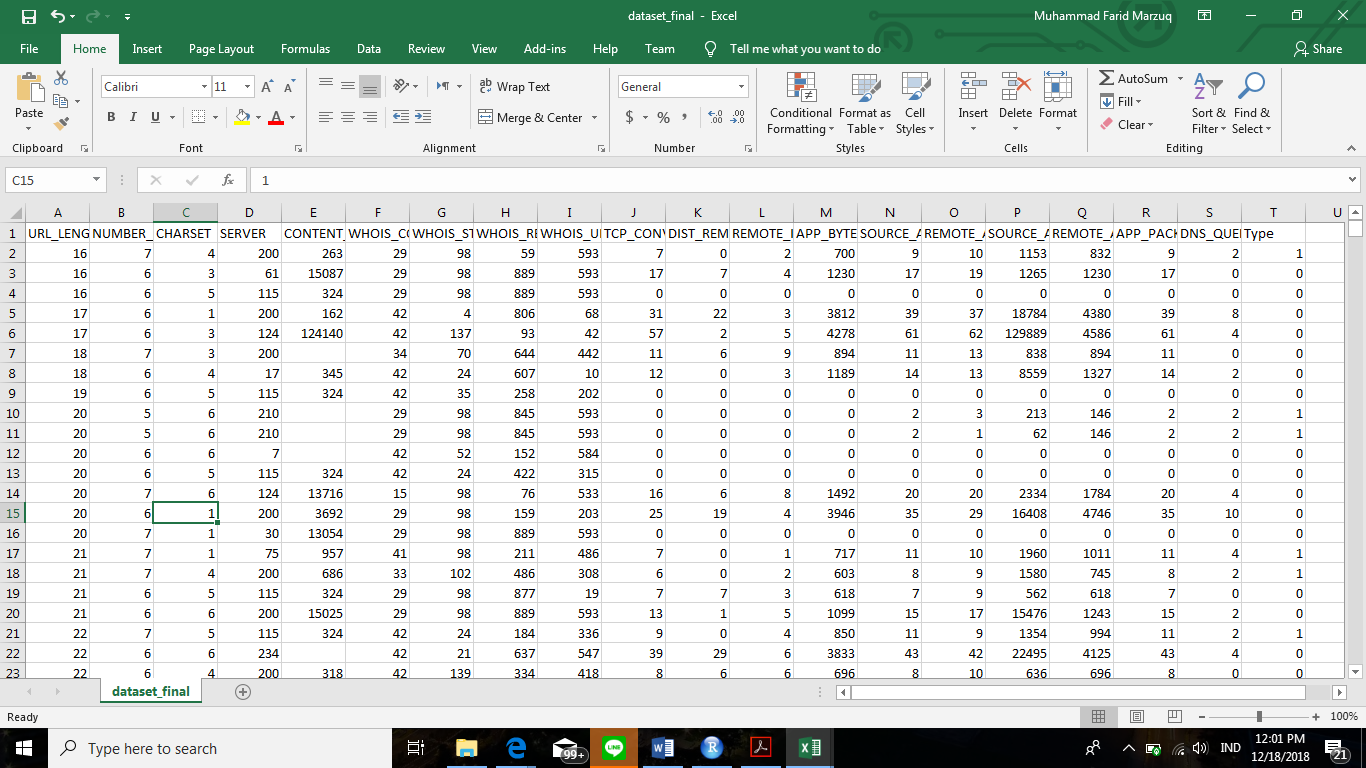
5: 'us-ascii',

6: 'utf-8',

7: 'windows-1251',

8: 'windows-1252'}

Gambar 3 Contoh hasil proses *ordinal encoding*



Gambar 4 Potongan dataset yang telah dipraproses

## **Pembagian Data**

Pada penelitian ini data dibagi menjadi data latih dan data uji. Data latih digunakan untuk membuat model klasifikasi, sedangkan data uji digunakan untuk menguji model klasifikasi. Pembagian data dilakukan menggunakan 10-*folds* *cross*-*validation*. Jumlah data untuk masing-masing kelas dapat dilihat pada Tabel 2.

Tabel 2 Jumlah data untuk masing-masing kelas

|  |  |
| --- | --- |
| Kelas | Jumlah Data |
| 0 | 1565 |
| 1 | 216 |

Pembagian data 10-*folds* *cross*-*validation* sebesar 90% data sebagai data latih dan 10% data sebagai data uji. Pembagian data diulang sebanyak sepuluh kali sampai semua *fold* pernah menjadi data uji. Distribusi kelas data pada data latih dan data uji dapat dilihat pada Tabel 3.

Tabel 3 Distribusi kelas data pada data latih dan data uji

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Fold* | Distribusi Kelas Data Latih | | | Distribusi Kelas Data Uji | | |
| 0 | 1 | Jumlah | 0 | 1 | Jumlah |
| 1 | 1408 | 194 | 1602 | 157 | 22 | 179 |
| 2 | 1409 | 194 | 1603 | 156 | 22 | 178 |
| 3 | 1408 | 195 | 1603 | 157 | 21 | 178 |
| 4 | 1409 | 194 | 1603 | 156 | 22 | 178 |
| 5 | 1408 | 195 | 1603 | 157 | 21 | 178 |
| 6 | 1409 | 194 | 1603 | 156 | 22 | 178 |
| 7 | 1409 | 195 | 1604 | 156 | 21 | 177 |
| 8 | 1408 | 194 | 1602 | 157 | 22 | 179 |
| 9 | 1409 | 195 | 1604 | 156 | 21 | 177 |
| 10 | 1408 | 194 | 1602 | 157 | 22 | 179 |

## **Pemodelan Klasifikasi Pohon Keputusan**

Implementasi pemodelan pohon keputusan menggunakan *package* C50 dalam bahasa pemrograman R. Potongan kode program untuk memodelkan pohon keputusan dapat dilihat pada Gambar 4. Kode program lengkap untuk proses pemodelan pohon keputusan secara keseluruhan dapat dilihat pada Lampiran 3. Pohon keputusan yang ditampilkan merupakan pohon keputusan dengan akurasi terbaik dari model-model yang terbentuk.

treeModel <- C5.0.default(x = trainData[,vars], y = trainData$Type)

assign(paste0("treeModel",i), treeModel)

Gambar 5 Potongan kode program untuk memodelkan pohon keputusan

Pohon keputusan yang terbaik dihasilkan oleh model dari *fold* ke-5. Model tersebut menunjukkan atribut WHOIS\_COUNTRY dengan nilai *information gain* tertinggi. Atribut lain yang digunakan adalah REMOTE\_APP\_PACKETS (95.26%), APP\_BYTES (62.17%), DIST\_REMOTE\_TCP\_PORT (60.61%), URL\_LENGTH (17.98%), NUMBER\_SPECIAL\_CHARACTERS (8.11%) dan WHOIS\_STATEPRO (3.18%). Hasil pemodelan pohon keputusan dapat dilihat pada Lampiran 4 dan Lampiran 5.

## **Pengujian Model Klasifikasi**

Tahap ini menguji tingkat akurasi model klasifikasi. Semakin mendekati 100%, maka tingkat akurasi model klasifikasi semakin baik. Semakin tinggi tingkat akurasi menunjukkan semakin sedikit data yang diklasifikasikan salah. Pada tahap ini tingkat akurasi model terbaik dibandingkan dengan ambang batas yang didapat dari penelitian sebelumnya (87.87%). Model terbaik memiliki akurasi sebesar 98.32% sehingga penelitian ini tidak perlu diiterasi. Hasil akurasi dapat dilihat pada Gambar 5.

Gambar 6 Grafik akurasi model klasifikasi

# **SIMPULAN** **DAN SARAN**

## **Simpulan**

Penelitian ini berhasil mengklasifikasikan situs web berbahaya dan aman menggunakan algoritme pohon keputusan C5.0. Pada penelitian ini dilakukan *encoding* data karena beberapa karakter pada data tidak dapat diterima oleh *package* C50, seperti tanda dua titik (:) dan titik koma (;). Atribut-atribut yang di-*encoding* adalah CHARSET, SERVER, WHOIS\_COUNTRY, WHOIS\_STATEPRO, WHOIS\_REGDATE, dan WHOIS\_UPDATED\_DATE. Atribut URL dihilangkan dari dataset karena merupakan data identifikasi unik dari data. Pada model klasifikasi atribut yang digunakan adalah WHOIS\_COUNTRY (100%), REMOTE\_APP\_PACKETS (95.26%), APP\_BYTES (62.17%), DIST\_REMOTE\_TCP\_PORT (60.61%), URL\_LENGTH (17.98%), NUMBER\_SPECIAL\_CHARACTERS (8.11%) dan WHOIS\_STATEPRO (3.18%). Model terbaik dari pohon keputusan memiliki tingkat akurasi sebesar 98.32% sehingga penelitian tidak perlu diiterasi.

## **Saran**

Saran yang dapat dilakukan bagi pemerintah adalah menggunakan model klasifikasi pada penelitian ini untuk menyaring situs-situs web yang dapat membahayakan pengguna internet. Pemerintah dapat mewajibkan pemasangan model klasifikasi pada penelitian ini pada penyedia jasa internet. Ketika pengguna terdeteksi membuka situs-situs web berbahaya dapat dimunculkan halaman peringatan sehingga pengguna internet dapat lebih berhati-hati.

Saran bagi penelitian selanjutnya adalah memperbaharui data dengan data yang lebih relevan dengan kondisi di Indonesia, seperti memperbanyak data yang berasal dari situs-situs di Indonesia. Selain itu disarankan untuk membuat aturan dari pohon keputusan yang dibuat agar mempermudah dalam implementasi model klasifikasi.

# **DAFTAR PUSTAKA**

[APJII] Asosiasi Penyelenggara Jasa Internet Indonesia (ID). 2017. *Infografis Penetrasi dan Perilaku Pengguna Internet Indonesia.* Jakarta (ID): Asosiasi Penyelenggara Jasa Internet Indonesia.

Hale J. 2018 Sep 11. Smarter ways to encode categorical data for machine learning (Part 1 of 3): exploring category encoders. [diakses 2018 Des 18]. Tersedia pada: https://towardsdatascience.com/smarter-ways-to-encode-categorical-data-for-machine-learning-part-1-of-3-6dca2f71b159

Han J, Kamber M, dan Pei J. 2012. *Data Mining: Concept and Techniques*. Edisi ke-3. Waltham, MA (US): Morgan Kaufmann Publishers.

[ITU] International Telecommunication Union (CH). 2018. Internet users by region and country, 2010–2016. International Telecommunication Union. [diakses 2018 Nov 25]. Tersedia pada: [https://www.itu.int/en/ITUD/Statistics/Pages/ stat/tr](https://www.itu.int/en/ITUD/Statistics/Pages/%20stat/tr)eemap.aspx

[Kemkominfo Ditjen Aptika DPI] Kementerian Komunikasi dan Informatika, Direktorat Jenderal Aplikasi Informatika, Direktorat Pemberdayaan Informatika (ID). 2015. *Internet Cerdas, Kreatif dan Produktif (Internet Cakap)*. Jakarta (ID): Direktorat Pemberdayaan Informatika, Direktorat Jenderal Aplikasi Informatika, Kementerian Komunikasi dan Informatika.

[Kemkominfo Ditjen Aptika DPI] Kementerian Komunikasi dan Informatika, Direktorat Jenderal Aplikasi Informatika, Direktorat Pemberdayaan Informatika (ID). 2018. Tentang API. Agen Perubahan Informatika. [diakses 2018 Nov 25]. Tersedia pada: http://www.api.id/tentang

Merriam-Webster Inc (US). 2018. Website. Merriam-Webster. [diakses 2018 Nov 26]. Tersedia pada: https://www.merriam-webster.com/dictionary/website

Munawaroh H, Khusnul B, dan Kustiyaningsih Y. 2013. Perbandingan algoritme ID3 dan C5.0 dalam identifikasi penjurusan siswa SMA. *Jurnal Sarjana Teknik Informatika* [internet]. [diunduh 2018 Des 18]; 1(1): 1–12. Tersedia pada: <https://anzdoc.com/perbandingan-algoritma-id3-dan-c50-dalam-indent>

ifikasi-penju.html

Pandya R dan Pandya J. 2015. C5.0 Algorithm to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Application* [internet]. [diunduh 2018 Nov 29]; 117(16): 18–21. Tersedia pada: <https://research.ijcaonline.org/volume117/number16/pxc3903>

318.pdf

Refaeilzadeh P, Tang L, dan Liu H. 2009. Cross-validation. Di dalam: Liu L dan Özsu MT, editor. *Encyclopedia of Database Systems*. Boston, MA (US): Springer.

Rizki R. 2018 Jul 17. Polri: Indonesia tertinggi kedua kejahatan siber di dunia. CNN Indonesia. [diakses 2018 Nov 25]. Tersedia pada: <https://www.cnnindonesia>.

com/nasional/20180717140856-12-314780/polri-indonesia-tertinggi-kedua-kejahatan-siber-di-dunia

Symantec Corporation (US). 2018. What are malicious websites?. Norton. [diakses 2018 Nov 27]. Tersedia pada: <https://us.norton.com/internetsecurity-malware>

-what-are-malicious-websites.html

Urcuqui C, Navarro A, Osorio J, dan García M. 2017. Machine learning classifiers to detect malicious websites. *CEUR Workshop Proceedings*. *Spring School of Networks* [internet]; 2017 Okt; Pućon, Chili. Aachen (DE): CEUR-WS.org. hlm 14–17; [diunduh 2018 Nov 26]. Tersedia pada: http://ceur-ws.org/Vol-1950/paper4.pdf

Xu L, Zhan Z, Xu S, dan Ye K. 2013. Cross-layer detection of malicious websites. *Proceedings of the Third ACM Conference on Data and Application Security and Privacy* [internet]; 2013 Feb 18–20; San Antonio, Texas, Amerika Serikat. New York, NY (US): ACM. hlm 141–152; [diunduh 2018 Nov 25]. Tersedia pada: <https://dl.acm.org/citation.cfm?id=2435366>

Lampiran 1 Contoh penggunaan algoritme C5.0 pada dataset kecil situs web berbahaya dan aman

Potongan data berikut diambil dari dataset keseluruhan dengan *stratified random sampling* berdasarkan kelas 0 (situs web aman) dan kelas 1 (situs web berbahaya).

Tabel 1.1 Potongan data dari dataset keseluruhan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | CHARSET | SERVER | WHOIS\_COUNTRY | WHOIS\_STATEPRO |
| 1 | 1 | 75.0 | 41 | 98 |
| 1 | 3 | 7.0 | 13 | 17 |
| 0 | 6 | 7.0 | 42 | 24 |
| 0 | 1 | 148.0 | 27 | 113 |
| 0 | 3 | 214.0 | 29 | 98 |
| 0 | 3 | 184.0 | 42 | 21 |
| 0 | 3 | 7.0 | 42 | 13 |
| 0 | 6 | 124.0 | 42 | 90 |
| 0 | 6 | 224.0 | 29 | 98 |
| 0 | 4 | 73.0 | 15 | 98 |

Berdasarkan tabel tersebut akan dilakukan seleksi atribut menggunakan rumus entropi dan *information gain*. Nilai *information gain* tertinggi akan menjadi *node* yang dipilih. Entropi dari kelas Type adalah sebagai berikut.

Nilai 8 didapat dari jumlah data dengan kelas 0 dan nilai 2 didapat dari jumlah data dengan kelas 1. Setelah itu dilakukan perhitungan untuk mendapatkan nilai *information gain* dari keseluruhan data. Rumus untuk menentukan nilai entropi bersyarat dari atribut CHARSET terhadap kelas Type adalah sebagai berikut.

Perhitungan di atas dilakukan juga untuk tiga atribut lainnya. Setelah itu dilakukan perhitungan *information gain*.

Berdasarkan perhitungan di atas didapatkan nilai *information gain* terbesar adalah atribut WHOIS\_COUNTRY. *Root* diambil dari nilai *information gain* tertinggi sehingga *root* dari pohon keputusan yang akan dibuat adalah atribut WHOIS\_COUNTRY. Perhitungan di atas diulang sampai data telah terklasifikasikan.

Lampiran 2 Kode program *ordinal encoding* dalam bahasa pemrograman Python

import numpy as np

import pandas as pd

import datetime

import pprint

dataset = pd.read\_csv("dataset.csv")

attributes = ['URL','URL\_LENGTH','NUMBER\_SPECIAL\_CHARACTERS','CHARSET','SERVER','CONTENT\_LENGTH','WHOIS\_COUNTRY','WHOIS\_STATEPRO','WHOIS\_REGDATE','WHOIS\_UPDATED\_DATE','TCP\_CONVERSATION\_EXCHANGE','DIST\_REMOTE\_TCP\_PORT','REMOTE\_IPS','APP\_BYTES','SOURCE\_APP\_PACKETS','REMOTE\_APP\_PACKETS','SOURCE\_APP\_BYTES','REMOTE\_APP\_BYTES','APP\_PACKETS','DNS\_QUERY\_TIMES','Type']

encoding = ['CHARSET','SERVER','WHOIS\_COUNTRY','WHOIS\_STATEPRO','WHOIS\_REGDATE','WHOIS\_UPDATED\_DATE']

dict\_charset = dict()

dict\_server = dict()

dict\_whois\_country = dict()

dict\_whois\_statepro = dict()

dict\_whois\_regdate = dict()

dict\_whois\_updated\_date = dict()

dataset["CHARSET"], uniq\_charset = dataset['CHARSET'].factorize(sort = True)

dataset["SERVER"], uniq\_server = dataset['SERVER'].factorize(sort = True)

dataset["WHOIS\_COUNTRY"], uniq\_whois\_country = dataset['WHOIS\_COUNTRY'].factorize(sort = True)

dataset["WHOIS\_STATEPRO"], uniq\_whois\_statepro = dataset['WHOIS\_STATEPRO'].factorize(sort = True)

dataset["WHOIS\_REGDATE"], uniq\_whois\_regdate = dataset['WHOIS\_REGDATE'].factorize(sort = True)

dataset["WHOIS\_UPDATED\_DATE"], uniq\_whois\_updated\_date = dataset['WHOIS\_UPDATED\_DATE'].factorize(sort = True)

index = 0

for i in list(uniq\_charset):

dict\_charset[index] = i

index += 1

index = 0

for i in list(uniq\_server):

dict\_server[index] = i

index += 1

index = 0

for i in list(uniq\_whois\_country):

Lampiran 2 Lanjutan

dict\_whois\_country[index] = i

index += 1

index = 0

for i in list(uniq\_whois\_statepro):

dict\_whois\_statepro[index] = i

index += 1

index = 0

for i in list(uniq\_whois\_regdate):

dict\_whois\_regdate[index] = i

index += 1

index = 0

for i in list(uniq\_whois\_updated\_date):

dict\_whois\_updated\_date[index] = i

index += 1

pprint.pprint("charset")

pprint.pprint(dict\_charset)

pprint.pprint("server")

pprint.pprint(dict\_server)

pprint.pprint("whois country")

pprint.pprint(dict\_whois\_country)

pprint.pprint("whois statepro")

pprint.pprint(dict\_whois\_statepro)

pprint.pprint("whois regdate")

pprint.pprint(dict\_whois\_regdate)

pprint.pprint("whois updated date")

pprint.pprint(dict\_whois\_updated\_date)

dataset.to\_csv("dataset\_encoded.csv")

Lampiran 3 Kode program proses pemodelan pohon keputusan secara keseluruhan dalam bahasa pemrograman R

#Menambahkan library

library(C50)

#Mempersiapkan data

data <- read.csv("dataset\_final.csv", header=TRUE)

summary(data)

set.seed(5674)

#Mengubah tipe data

data$CHARSET <- as.factor(data$CHARSET)

data$SERVER <- as.factor(data$SERVER)

data$WHOIS\_COUNTRY <- as.factor(data$WHOIS\_COUNTRY)

data$WHOIS\_STATEPRO <- as.factor(data$WHOIS\_STATEPRO)

data$WHOIS\_REGDATE <- as.factor(data$WHOIS\_REGDATE)

data$WHOIS\_UPDATED\_DATE <- as.factor(data$WHOIS\_UPDATED\_DATE)

data$Type <- as.factor(data$Type)

summary(data)

#Membagi data per kelas

vars <- c("URL\_LENGTH","NUMBER\_SPECIAL\_CHARACTERS","CHARSET","SERVER","CONTENT\_LENGTH","WHOIS\_COUNTRY","WHOIS\_STATEPRO","TCP\_CONVERSATION\_EXCHANGE","DIST\_REMOTE\_TCP\_PORT","REMOTE\_IPS","APP\_BYTES","SOURCE\_APP\_PACKETS","REMOTE\_APP\_PACKETS","SOURCE\_APP\_BYTES","REMOTE\_APP\_BYTES","APP\_PACKETS","DNS\_QUERY\_TIMES")

str(data[, c(vars, "Type")])

data <- data[sample(nrow(data)),]

dataClass0 <- data[which(data$Type == '0'),]

dataClass0 <- data[sample(nrow(dataClass0)),]

dataClass1 <- data[which(data$Type == '1'),]

dataClass1 <- data[sample(nrow(dataClass1)),]

#10 folds

##Membuat 10 folds

foldsClass0 <- cut(seq(1,nrow(dataClass0)), breaks = 10, labels = FALSE)

foldsClass1 <- cut(seq(1,nrow(dataClass1)), breaks = 10, labels = FALSE)

sumAccuracy = 0

for(i in 1:10){

##Membagi ke dalam folds ke-i

indexClass0 <- which(foldsClass0 == i, arr.ind = TRUE)

indexClass1 <- which(foldsClass1 == i, arr.ind = TRUE)

testDataClass0 <- dataClass0[indexClass0,]

trainDataClass0 <- dataClass0[-indexClass0,]

testDataClass1 <- dataClass1[indexClass1,]

trainDataClass1 <- dataClass1[-indexClass1,]

Lampiran 3 Lanjutan

testData <- rbind(testDataClass1,testDataClass0)

trainData <- rbind(trainDataClass1,trainDataClass0)

assign(paste0("dataTest",i), testData)

assign(paste0("dataTrain",i), trainData)

##Membuat model pohon keputusan

treeModel <- C5.0.default(x = trainData[,vars], y = trainData$Type)

assign(paste0("treeModel",i), treeModel)

##Menghitung akurasi

predict <- (predict(treeModel, testData))

sum=0

for (j in 1:nrow(testData)){

if(predict[j] == testData$Type[j])

sum = sum + 1

}

accuracy <- sum/nrow(testData)\*100

assign(paste0("accuracyFold",i), accuracy)

sumAccuracy = sumAccuracy + accuracy

}

#Menghitung rata-rata akurasi

avgAccuracy = sumAccuracy/10

Lampiran 4 Model klasifikasi pohon keputusan dengan algoritme C5.0

WHOIS\_COUNTRY = 25:0(0)

WHOIS\_COUNTRY in {4,10,13,32,33,39,40,43,46}: 1(76/2)

WHOIS\_COUNTRY in {0,1,2,3,5,6,7,8,9,11,12,14,15,16,17,18,19,

: 20,21,22,23,24,26,27,28,29,30,31,34,35,36,37,38,41,42

: ,44,45,47,48}:

:...REMOTE\_APP\_PACKETS <= 0:0(530)

REMOTE\_APP\_PACKETS > 0:

:...APP\_BYTES <= 132:

:...WHOIS\_STATEPRO = 21:0(3)

: WHOIS\_STATEPRO in {0,1,2,3,4,5,6,7,8,9,10,11,12,

: 13,14,15,16,17,18,19,20,22,23,24,25,26,27

: ,28,29,30,31,32,33,34,35,36,37,38,39,40,

: 41,42,43,44,45,46,47,48,49,50,51,52,53,54

: ,55,56,57,58,59,60,61,62,63,64,65,66,67,

: 68,69,70,71,72,73,74,75,76,77,78,79,80,81

: ,82,83,84,85,86,87,88,89,90,91,92,93,94,

: 95,96,97,98,99,100,101,102,103,104,105,

: 106,107,108,109,110,111,112,113,114,115,

: 116,117,118,119,120,121,122,123,124,125,

: 126,127,128,129,130,131,132,133,134,135,

: 136,137,138,139,140,141,142,143,144,145,

: 146,147,148,149,150,151,152,153,154,155,

: 156,157,158,159,160,161,162,163,164,165,

: 166,167,168,169,170,171,172,173,174,175,

: 176,177,178,179,180,181}:1(22/2)

APP\_BYTES > 132:

:...DIST\_REMOTE\_TCP\_PORT > 1:0(683/9)

DIST\_REMOTE\_TCP\_PORT <= 1:

:...URL\_LENGTH > 42:0(158/31)

URL\_LENGTH <= 42:

:...NUMBER\_SPECIAL\_CHARACTERS > 9:1(41/2)

NUMBER\_SPECIAL\_CHARACTERS <= 9:

:...NUMBER\_SPECIAL\_CHARACTERS <= 6:0(14)

NUMBER\_SPECIAL\_CHARACTERS > 6:

:...URL\_LENGTH <= 27:1(11)

URL\_LENGTH > 27:

:...NUMBER\_SPECIAL\_CHARACTERS <=

8:0(38/4)

NUMBER\_SPECIAL\_CHARACTERS >

8:

:...WHOIS\_STATEPRO in {21,27

: ,39,67,101}:0(11)

WHOIS\_STATEPRO in {0,1,3

,3,4,5,6,7,8,9,10,

11,12,13,14,15,16,

17,18,19,20,22,23,

24,25,26,28,29,30,

31,32,33,34,35,36

37,38,40,41,42,43,

44,45,46,47,48,49,

50,51,52,53,54,55,

Lampiran 4 Lanjutan

56,57,58,59,60,61,

62,63,64,65,66,68,

69,70,71,72,73,74,

75,76,77,78,79,80,

81,82,83,84,85,86,

87,88,89,90,91,92,

93,94,95,96,97,98,

99,100,102,103,104

,105,106,107,108,

109,110,111,112,

113,114,115,116,

117,118,119,120,

121,122,123,124,

125,126,127,128,

129,130,131,132,

133,134,135,136,

137,138,139,140,

141,142,143,144,

145,146,147,148,

149,150,151,152,

153,154,155,156,

157,158,159,160,

161,162,163,164,

165,166,167,168,

169,170,171,172,

173,174,175,176,

177,178,179,180,

181}:1(15/1)

Lampiran 5 Visualisasi model klasifikasi pohon keputusan

